Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion

JOHN J. DUDLEY, University of Cambridge
KEITH VERTANEN, Michigan Technological University
PER OLA KRISTENSSON, University of Cambridge

We present the VISAR keyboard: An augmented reality (AR) head-mounted display (HMD) system that supports text entry via a virtualised input surface. Users select keys on the virtual keyboard by imitating the process of single-hand typing on a physical touchscreen display. Our system uses a statistical decoder to infer users' intended text and to provide error-tolerant predictions. There is also a high-precision fall-back mechanism to support users in indicating which keys should be unmodified by the auto-correction process. A unique advantage of leveraging the well-established touch input paradigm is that our system enables text entry with minimal visual clutter on the see-through display, thus preserving the user's field-of-view. We iteratively designed and evaluated our system and show that the final iteration of the system supports a mean entry rate of 17.75wpm with a mean character error rate less than 1%. This performance represents a 19.6% improvement relative to the state-of-the-art baseline investigated: A gaze-then-gesture text entry technique derived from the system keyboard on the Microsoft HoloLens. Finally, we validate that the system is effective in supporting text entry in a fully mobile usage scenario likely to be encountered in industrial applications of AR HMDs.

CCS Concepts: • **Human-centered computing → Text input**;

Additional Key Words and Phrases: Augmented reality, text entry

ACM Reference format:
John J. Dudley, Keith Vertanen, and Per Ola Kristensson. 2018. Fast and Precise Touch-Based Text Entry for Head-Mounted Augmented Reality with Variable Occlusion. *ACM Trans. Comput.-Hum. Interact.* 25, 6, Article 30 (December 2018), 40 pages.
https://doi.org/10.1145/3232163

1 INTRODUCTION

Recent progress in head-mounted displays (HMDs) for augmented reality (AR), such as the Microsoft HoloLens, demonstrates the commercial potential of AR to support new forms of interaction and work in a range of industries including construction, education and health. Text entry is an integral activity in such AR environments, allowing users to, for example, send short messages,

P. O. Kristensson was supported in part by a Google Faculty Research Award and EPSRC Grants EP/N010558/1 and EP/N014278/1. K. Vertanen was supported in part by a Google Faculty Research Award. J. Dudley was supported by the Trimble Fund. Supporting data for this publication is available at https://doi.org/10.17863/CAM.25391.

Authors' addresses: J. J. Dudley and P. O. Kristensson, Department of Engineering, University of Cambridge, Trumpington St, Cambridge CB2 1PZ, UK; emails: {jjd50, pok21}@cam.ac.uk; K. Vertanen, Department of Computer Science, Michigan Technological University, 1400 Townsend Dr, Houghton, MI 49931; email: vertanen@mtu.edu.

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 the author(s) 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.
© 2018 Copyright held by the owner/author(s). Publication rights licensed to ACM

1073-0516/2018/12-ART30 $15.00
https://doi.org/10.1145/3232163

annotate physical objects and digital content, compose documents or fill out forms. The placement of the user within a virtually augmented environment introduces new and exciting opportunities for the interface designer. The design space is considerably broadened by the additional dimensionality available and new forms of interaction are made possible.

New challenges also emerge in providing effective text entry for AR HMDs. First, currently available devices are typically limited in terms of their display region size. Compounding the limited size is the fact that the display region is located in the centre of the user's field-of-view. Delivering a text entry method that preserves field-of-view while supporting effective input presents a unique design challenge. Second, delivering immersive and fully mobile AR applications in which the user can freely explore and interact with both the physical and virtual environment suggests avoiding input devices that encumber the user. Avoiding encumbering the user while maintaining freedom of mobility means that external (off-body) sensing to support text entry is also not practical. Third, a key goal of AR applications in general should be to minimise or eliminate the distinction between physical and virtual content from the perspective of the user. A text entry method for AR should thus be consistent with the broader experience and maintain any developed sense of immersion.

In response to the identified challenges, this article presents a novel system that enables users to type on a virtual keyboard using a head-mounted AR device and hand localisation derived from body-fixed sensors. We call this system the virtualised input surface for augmented reality (VISAR) keyboard. The VISAR keyboard is a probabilistic auto-correcting translucent keyboard system with variable occlusion, specifically designed for AR HMDs, such as the Microsoft HoloLens. Our system seeks to leverage learned keyboard interaction behaviour and exploit the additional dimensionality of the design space available in AR. By adapting a state-of-the-art probabilistic decoder, we enable people to type in a fashion that is familiar and akin to typing on their mobile phone keyboard or on a wall-mounted touch-capable display. To the authors' knowledge, this is the first investigation of providing a touch-driven text entry method specifically designed for AR and based upon body-fixed (as opposed to space-fixed) sensor data. The system is thus fully encapsulated by the head-mounted device and enables truly mobile, unencumbered text entry for AR. Furthermore, we seek to specifically address the unique design requirements of optical see-through head-mounted AR by accommodating design objectives that relate to the constrained display size, and therefore explore minimising occlusion of the user's field-of-view. We also guide future text entry design for AR HMDs by presenting the results of four experiments and one validation study that investigate the implications of the design choices in the VISAR keyboard. The design of VISAR is underpinned by six solution principles for AR text entry, which we have distilled from the literature and prior experience in text entry design.

Experiment 1 revealed that novice users with minimal practice reach entry rates that are comparable with the current standard interaction technique used within the Microsoft HoloLens default system keyboard, which requires that users move their head to place a gaze-directed cursor on the desired key, and then perform a hand gesture. However, we find that in terms of discrete selection events, the virtual touch technique used in VISAR is on average 17.4% faster than the baseline method.

In Experiment 2, we investigated how allowing users to seamlessly shift from probabilistic auto-correcting text entry to literal text input without an explicit mode-switch affects performance. The results revealed that users most commonly exploited the inferred-to-literal fall-back method to pre-emptively enter words they did not expect the decoder to recognise. The inferred-to-literal fall-back method introduces a speed penalty due to the requirement to dwell on a key to make a selection. Despite this penalty associated with dwell, for phrases with a high degree of uncertainty under the language model, participants were able to type as quickly as they did without the fall-back method but with reduced character error rates (CER).

Experiment 3 revealed that the interaction techniques applied in VISAR enable users to type effectively even when key outlines and labels are hidden. Out of the 12 participants who completed both Experiment 2 and 3, 10 achieved their highest entry rates in one of the two reduced occlusion configurations examined. This shows that the majority of participants were able to readily transfer their learned typing skills to the novel interface approach. Varying keyboard occlusion in AR typing has to our knowledge never been proposed or explored before.

Experiment 4 returns to the baseline comparison but with the design improvements identified in Experiments 1 to 3 incorporated into VISAR and with the addition of word predictions. User performance was evaluated under extended use with participants typing between 1.5 to 2 hours in each condition over a fixed number of test blocks. We demonstrate that the refined VISAR design achieves a mean entry rate of 16.76 words-per-minute (wpm) compared with 14.26wpm in the baseline condition. Analysing only the results from the final four blocks in each condition (i.e., after the most pronounced learning effect has subsided), the mean entry rates are then 17.75wpm and 14.84wpm for the VISAR and the baseline conditions, respectively.

Finally, we conducted a validation study, which demonstrated that the VISAR keyboard is a viable text entry method for typical text entry tasks anticipated for productive use of AR HMDs. The user experience of the system is examined in four sub-tasks involving transcription, composition, replying to a message, and freely annotating real-world objects.

1.1 Outline

This article begins by reviewing work relevant to the design of an effective text entry method for AR HMDs. A set of six key solution principles is distilled from the literature and prior experience derived from AR interface design. These solution principles inform the design of the VISAR keyboard, which is described in detail in Section 4. The results of the five experiments briefly described above are then presented. Finally, the article is concluded with a discussion of the limitations and open issues related to designing and deploying a fully featured AR keyboard based on a touch-driven paradigm.

2 RELATED WORK

The technological advancements in HMDs are frequently accompanied by research seeking to empower their users with productive text entry methods. In this section, we review the literature relevant to developing efficient text entry for HMDs. Text entry is an active field of research that has produced a plethora of input techniques. We constrain our scope here to work that informs the design of text entry techniques in AR, both from an interaction and language perspective.

First, we review efforts to support productive text entry in circumstances where user selections are subject to high levels of error, such as are potentially encountered in HMD contexts exploiting coarse hand tracking. We then examine research that specifically targets productive text entry for HMDs in both Virtual and AR. Last, we look at work which explores providing mid-air text input not necessarily accompanied by a head-worn display.

2.1 Intelligent Text Entry

A distinction can be drawn between text entry methods that simply insert a selected letter versus those which provide intermediary intelligence to infer user intent. Patterns in language and knowledge of human behaviour can be exploited to improve text entry performance (both in terms of entry rate and error rate) for a given input technique. Adding intelligence to a text entry method is of particular value when the input process is constrained by some physical limitations.

Small keyboards, constrained by the size of device on which they are deployed, are an obvious target for application of smart methods for inferring input and correcting errors. Goodman et al.

(2002) demonstrated the potential of a character-level language model to help reduce error rate by inferring user's intended key presses on a PDA soft keyboard. Thumb typing on small keyboards is similarly error prone due to key occlusion. Faster entry rates typically also exacerbate error rates. For example, Clarkson et al. (2005) observed that participants could thumb type on a small QWERTY keyboard at 31.72wpm in session 1 with error rates of 6.12% but that this then grew to 60.03wpm with error rates of 8.32% after 20 sessions of 20 minutes. Kim et al. (2006) evaluated a small wrist worn keyboard intended for a wearable computing environment that allowed users to achieve 18.9wpm after five 20 minute sessions. They also noted an issue with small keyboard size and error rates, which were consistently averaging above 6% over the five sessions. To address such error rates in small keyboards, Clawson et al. (2008) applied a decision tree approach to correct 32.37% of user errors on a miniature physical thumb typing keyboard. Kristensson and Zhai (2005) use an alternative approach based on pattern-matching to compare user's tapped points to ideal point templates. In a pilot evaluation of the entry technique on a stylus keyboard, peak entry rates in the range of 37.7 to 51.8wpm were achieved.

Rather than relying on a character or symbolic approach to text input, it is possible to exploit people's most efficient language communication channel: voice. In situations where there is low environmental noise and privacy is not a concern, speech recognition offers the potential for very fast input. Accuracy has historically been a problem for speech-to-text input but recent advances have substantially lowered error rates (Hinton et al. 2012). Entry rates can also be improved by providing effective interfaces to support rapid correction of recognised speech. Pick et al. (2016) investigated text entry in a CAVE virtual environment using speech recognition with correction taking place via a hand-held point-and-click device. Participants achieved an average of 23.6wpm with word error rates of 0.56%. The SpeeG2 (Hoste and Signer 2013) interface allows text input via speech recognition with correction taking place via gestures sensed by a depth sensor. SpeeG2 supported entry at 21wpm with users interacting in front of a wall-sized display.

In general, user performance can be considerably enhanced by exploiting the predictability of language and behaviour. There is, however, an inevitable tradeoff against agency and false positives should the user intent and inferred intent diverge in an intrusive fashion.

2.2 Text Entry for HMDs

Determining how best to enter text or other symbolic input in VR has been a long standing research problem. Early systems allowed short text input or annotation of the virtual environment via handwritten graphical notes/drawings (e.g., Poupyrev et al. (1998)), audio annotations (e.g., Harmon et al. (1996) and Verlinden et al. (1993)), or via hand gestures sensed with a glove (e.g., Kuester et al. (2005) and Rosenberg and Slater (1999)). Bowman et al. (2002) compared the following four different input methods for entering text in VR while wearing an HMD: speech recognition (performed by a human), pinch gloves, a pen and tablet, and a chord keyboard. Entry rates are as follows: 4wpm chord keyboard, 6wpm pinch gloves, 10wpm pen and tablet, and 13wpm speech.

Yu et al. (2017) also evaluated three alternative text entry methods for VR by exploiting the fine head-motion tracking capability of modern HMDs. Using a head-fixed gaze cursor, the following three methods examined alternative approaches to indicate a key selection: dwell, button-press on a game pad, and gaze-based-gesture path (also using a game pad to indicate gesture start and finish events). In a comparative study, the three methods achieved average entry rates of 10.59, 15.58, and 19.04wpm, respectively, in the sixth session (eight phrases per session). After further refinement of the gaze-based-gesture entry method, including correcting for an observed performance difference between head movement up and down versus left and right, an average entry rate of 24.73wpm was achieved after 8 sessions of typing the same 10 phrases repeatedly.

Text entry for AR has received less attention in the literature. The augmented reality keyboard (ARKB) (Lee and Woo 2003) uses a stereo visible light camera on an HMD to track coloured markers attached to a user's fingers. ARKB detects collision with a virtual QWERTY keyboard display in the HMD. No user trial results were reported.

SwipeZone (Grossman et al. 2015) exploits the touch region of the side of Google Glass to deliver a text entry method involving swiping to select key groups then the desired letter. In a controlled experiment involving entry of 10 five-letter words per block for 20 blocks, participants achieved a mean entry rate of 8.73wpm in the final block. Also focusing on the Google Glass, Yu et al. (2016) present a one-dimensional unistroke gesture technique that allows text input. The input system made use of a probabilistic stroke and language model. In their second study session, participants were able to enter words at 9wpm.

The PalmType system (Wang et al. 2015) used Google Glass to display a QWERTY keyboard interface on a user's palm. In a user study that used a Vicon tracking system, users were able to type on the palm of their hand at 8wpm. Using a wrist-worn IR sensor users typed at 5wpm. Input was literal with no auto-correction algorithm.

The various studies of text entry methods specifically targeting HMDs suggests typical entry rates in the range of 5 to 25wpm without use of a physical keyboard. It should be noted, however, that the approaches that prove effective for VR may not transfer well to AR and vice versa. The distinction between AR and VR is not always meaningful, particularly in terms of the physical execution of interactions and more generally in aspects of software architecture. There are, however, prominent distinctions between AR and VR that should not be ignored. For example, the fact that many VR systems are tethered or for other reasons preclude extended user mobility means that the use of controllers or other input devices and fixed sensing infrastructure may be appropriate. Tracking provided by input devices or fixed infrastructure is likely to be characterised by higher accuracy and lower system delays. Similarly, the lack of control over the background scene in AR means that the visual features of a text entry interface may need to be considerably different from the same approach used in VR.

2.3 Mid-Air Text Entry

A variety of work has looked at how to capture input for text entry with unobtrusive sensing rather than with input-specific devices such as a glove or strap-on mini-keyboard. DigiTap (Prätorius et al. 2014) uses a wrist-mounted camera to detect the thumb touching 12 different points on a user's finger. No formal user study was presented, but one of the authors was able to achieve 10wpm entering text via a literal multi-tap keyboard mapped to the finger locations.

Markussen et al. (2013) evaluate three alternative text input methods for interacting with large wall displays. Hand tracking was provided by an OptiTrack system and a glove with reflective markers. Hand movements were mapped onto the wall display and 'taps' were recognised based on an angle threshold of the vector between the hand and fingertip. After six 45 minute sessions, participants recorded a mean entry rate of 13.2wpm in the best performing condition: A projected full QWERTY keyboard implementation in which users would select individual keys by moving their hand and then 'tapping'.

Iterating on these techniques, Markussen and colleagues produced Vulture (Markussen et al. 2014), a word-gesture keyboard (Zhai and Kristensson 2012) designed to operate in mid-air. Again using a wall-sized display and an OptiTrack motion-capture system, users wrote at 21wpm. As is typical with word-gesture keyboards, a probabilistic language model and template matching algorithm was used.

AirStroke (Ni et al. 2011) allows users to type using freehand gestures based on the Graffiti alphabet. The non-gesturing hand can also be employed to select word predictions. When word predictions were included, participants were able to achieve entry rates of approximately 13wpm with error rates under 5%.

The emergence of sensors that support fine hand and finger articulation tracking, such as the Leap Motion, has enabled the investigation of a variety of free-hand mid-air text entry techniques. Sridhar et al. (2015) demonstrate text entry using a multi-finger gesture set. Using a repeated word evaluation as proposed in (Bi et al. 2012), participants achieved a mean peak performance of 22.25wpm. The air typing keyboard (ATK) (Yi et al. 2015) allows 10-finger typing in mid-air with the location of fingers detected by a leap motion depth sensor. A probabilistic decoder is used to infer the user's typing, although ATK did not model insertions or deletions. The ATK thus allows users to type as they would on a typical mechanical keyboard by extending their fingers as if pressing keys. Users were reportedly able to type at 29wpm after an hour of practice; however, stimulus phrases were purposely selected to only include words within the vocabulary.

Fully articulated hand tracking to support 10 finger mechanical-keyboard-like text entry is perhaps the holy grail for mid-air text entry. The work of Yi et al. (2015) suggests that this may be achievable but currently available sensor technologies face difficulties supporting such an interaction in a fully mobile AR use case. The requirement for on-body sensors to perform hand localisation and articulation estimation is challenged by observability constraints and independent, non-rigid movement of appendages. These issues may be remedied to some extent through careful sensor placement and improved sensor design with the AR HMD use case specifically in mind. As it stands, however, the various studies suggest that hand-based mid-air text entry, even with external fixed sensors, is typically in the range of 10 to 30wpm. The interactions that are feasible under external tracking of hand position and articulation are not necessarily feasible in a body-fixed sensing scenario.

3 SOLUTION PRINCIPLES

The VISAR keyboard is intended to satisfy the design goal of providing an efficient and accurate text entry method for use in AR. The design of the VISAR keyboard is guided by six key principles that are proposed as critical features to an effective AR text entry solution. These solution principles are derived with reference to the recognised features that make conventional two-dimensional text entry methods effective as well as the unique requirements of immersive interfaces for AR.

It is envisioned that the majority of mobile AR text entry use cases will be light text entry only, i.e., occasional entry of usernames, passwords, search terms, or short phrases. This imagined usage behaviour also informs the following solutions principles.

SP 1. Rapid Input Selection

Presenting a virtual keyboard using a HMD enables a variety of potential selection methods. Text entry may be thought of as a connected sequence of discrete key selections. In order to maximise overall entry rates, interaction techniques that support rapid selection are preferable. The speed of key selection does, however, expose a potential tradeoff against input error, hence, the following corresponding solution principle, tolerance to inaccurate selection.

SP 2. Tolerance to Inaccurate Selection

Mid-air keyboard text entry by hand tracking is an inherently noisy process with a high risk of false identification of intended keys. Unmitigated, the error prone user is likely to fall back on a closed loop strategy involving selection then review. This strategy is highly detrimental to entry rate. To mitigate the inaccurate selection process, it is possible to interpret the user's input via a probabilistic decoder that treats each attempted key press by the user as an uncertain *observation*. The decoder then decodes an *observation sequence* of such key presses into individual words by assigning a posterior probability distribution over candidate words. Performance of the decoder is related to the span-length of the statistical language model and the size of the text corpora.

SP 3. Minimal Occlusion of Field-of-View

AR optical see-through displays are designed to allow users to be highly mobile and maintain visibility of the real environment. The current commercially-available devices provide fairly limited display regions that are unavoidably located in the centre of the user's field-of-view. This constrained display real estate introduces unique considerations related to the placement and styling of content. Future iterations of the AR HMDs will likely seek to expand the usable display region. While there are, no doubt, technical challenges that will make this difficult to achieve, AR delivered over the user's full field-of-vision is an eventuality for which HCI researchers should prepare. Even in this eventuality, however, it will be preferable to avoid obscuring the user's central field-of-view when non-essential so that they may continue to attend the physical environment. Supporting text entry under this constraint is thus a necessary but difficult to accommodate design objective. Where possible, text entry methods should seek to minimise the occlusion of the real world.

SP 4. Intelligent Word Predictions

When key selection speed is limited due to physical constraints on the interaction method, entry rates may be significantly increased by exploiting word predictions based on probabilistic language models. The user may only need to type several characters before the desired word is presented as a predicted option based on the prefix. Fortunately, users are increasingly exposed to such models within many mobile text entry keyboards, and so the same approaches may be readily applied in AR.

SP 5. Fluid Regulation between Input Modes

The application of a decoder to auto-correct typing mistakes and errors due to sensor noise is discussed within *SP 2*. However, sometimes users intend to write text that is unlikely to be predicted by a statistical decoder, for instance usernames or passwords, which are often intentionally designed to exhibit high perplexity under a statistical language model. It is therefore important to support *fluid regulation* of the user's uncertainty that allows users to easily indicate to the system whether they desire their key presses to be decoded or to be interpreted literally.

SP 6. Walk-Up Usability and Acceptance

It is notoriously difficult to design a text entry method that will be adopted by users. Hundreds of text entry methods are proposed in the HCI literature but very few achieve mainstream adoption. A theory in economics known as *path dependence* helps explain this phenomenon (David 1985) (for an alternative view, see (Liebowitz and Margolis 1990); see also (Kristensson 2015; Zhai et al. 2005)). In order to use a new text entry method users need to invest learning time. If the text entry method is radically different from a QWERTY keyboard, this learning time can be substantial.

Also relevant in the context of text entry applications for AR HMDs is the requirement to smoothly transition into other tasks. For example, users may wish to effortless transition between labelling an object with text and adjusting its position. Modes of interaction that minimise the effort required to switch between discrete tasks are thus desirable.

In summary, the above solution principles guide the design decisions behind the development of the VISAR keyboard. The following section describes the system design, and where relevant, reference is made to the corresponding solution principle that has informed the choices made.

4 VISAR SYSTEM DESIGN

The VISAR system splits function between two principle components: the decoder and the mid-air virtual keyboard. The mid-air virtual keyboard provides the visual interface and supports the direct-touch interaction technique. The decoder provides corrections of noisy key selections made on the virtual keyboard. Together they deliver a natural and immersive interaction method that enables text entry at moderate speed with acceptable error rates.

4.1 Decoder

Due to inaccuracies in the tracking of a user's hand and in the perceived location of the virtual keyboard, the recorded tap location and that of the user's actual intended key target will very likely differ. To infer a user's intended text from this noisy input data, we extended the VelociTap decoder (Vertanen et al. 2015). Enabling error-tolerant typing is anticipated to allow users to maintain higher entry rates according to *SP 2*. We repeat details from (Vertanen et al. 2015) for completeness.

The decoder allows users to enter text by tapping out all the letters of a sentence on a touchscreen virtual keyboard. After entry, the entire sentence of noisy touch locations is provided as the input observations for decoding. The decoder searches for the most probable sentence that is consistent with the observations but that is also probable under a language model.

The goal of the decoder's search is to find the sequence of actions that consumes all the observations and does so with the highest probability. The first action the decoder can take is to generate a character from the keyboard. The probability of generating a character is based on the likelihood of the observation's location under a two-dimensional Gaussian centred at the key labelled with that character. The two-dimensional Gaussians are axis-aligned with two separate parameters controlling the x- and y-variances. All keys share the same two parameters. This action prefers characters near a tap's location, but generates new hypotheses for all possible keyboard characters with probability diminishing for further away keys.

The input sequence may contain an extra observation (e.g., if a user accidentally taps a key twice). The second decoder action is to delete an observation without outputting a character. A deletion penalty is assessed whenever this action is taken.

The input sequence could also be missing an observation (e.g., if a tap fails to register). The third decoder action inserts an output character without advancing in the observations. This action proposes the insertion of all possible characters. Each new hypothesis pays an insertion penalty.

In the above actions, whenever a hypothesis proposes the generation of a character, including space, the hypothesis is assigned an additional penalty based on the probability of that character given the previously written text under a character n-gram language model. Whenever a space is output, the hypothesis is further penalised by a word language model based on the previous words. The character and word language model probabilities are multiplied by either a character or a word scale factor. Words that are not in a known vocabulary list incur an additional out-of-vocabulary penalty.

The tradeoff between speed and accuracy is controlled by a beam width. The decoder tracks the highest probability hypothesis seen at every point in the observation sequence. Hypotheses passing through that observation that are too improbable compared to this previous best probability are pruned. The decoder's search proceeds in parallel with multiple threads extending hypotheses. Once a hypothesis consumes all observations, it represents some possible text with a given probability. The decoder remembers the n-best finishing hypotheses for a given observation sequence.

To date, VelociTap has only been used for decoding an entire sentence of input. We wanted the VISAR keyboard to allow users to perform word-at-a-time entry. We extended the decoder to utilise known text context to both the left and to the right of the noisy input sequence. In our case, the left context is the previously written words for a given sentence. If no text has been written yet, the left context is the language model's sentence start pseudo-word. The left text provides context to the character and word language models during the decoder's search, i.e., it biases the search towards text that makes sense given what was previously written.

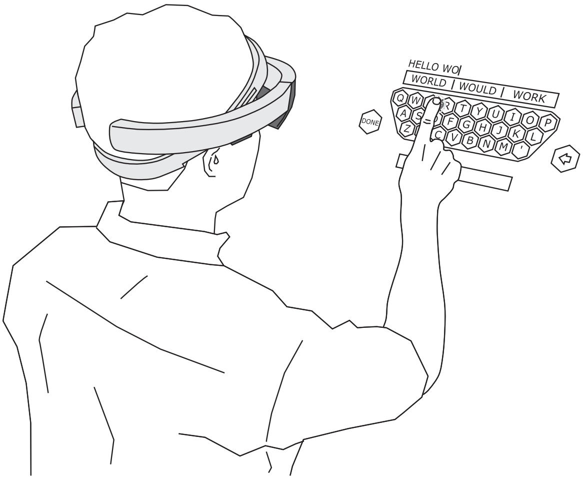

Fig. 1. Illustration of a user typing on the VISAR keyboard.

If right context is given, a hypothesis before finishing is assessed with character and word language model probabilities based on generating this right text. This makes sure the hypothesis makes sense given whatever comes after the word. For our application, we always set the right context to be a space. This had the effect of biasing the search towards complete words.

Our language models were trained using SRILM (Stolcke 2002) on billions of words of twitter, usenet, blog, social media, movie subtitle, and web forum data. We trained a 12-gram character model using Witten–Bell smoothing. We trained a 4-gram word model using interpolated modified Kneser–Ney smoothing. We used entropy pruning to reduce model size to 578MB for the character model and 1.0GB for the word model. The decoder incorporated a vocabulary of 64,000 words.

The free parameters of the decoder such as the variances, deletion penalty, insertion penalty, out-of-vocabulary penalty, and language model scale factors were optimised with respect to development data recorded by two of the authors and two other volunteers who did not take part in any of our user studies. The development data consisted of sentences not used in the user studies. As will be discussed, we tested a normal virtual keyboard with key outlines and labels as well as keyboards with reduced visual features. We optimised two sets of parameters, one for the normal keyboard and one for keyboards with reduced visual features.

4.2 Mid-Air Virtual Keyboard

The Microsoft HoloLens provides the hardware platform for the implementation of the mid-air virtual keyboard. The HoloLens is a head-mounted see through display that also provides coarse hand-tracking. The HoloLens constructs and maintains a spatial map of the environment. This map can subsequently be exploited to maintain fine tracking of the user head location and orientation within the environment. Virtual objects can then be placed in the user's view such that they appear fixed within the local environment.

The virtual keyboard in this study is generated as a two-dimensional panel of keys. Figure 1 illustrates the virtual keyboard concept. The keyboard layout employed in this study was simplified to contain only characters *A* to *Z* and apostrophe (total of 27 character keys). The *SPACE* key is used as the trigger to activate the decoder on the most recent observation sequence. The *DONE* key is used in the experiments described to indicate completion of the set phrase. The *BACKSPACE* key removes previous touch input unless a space was entered and the word decoded. If the previous

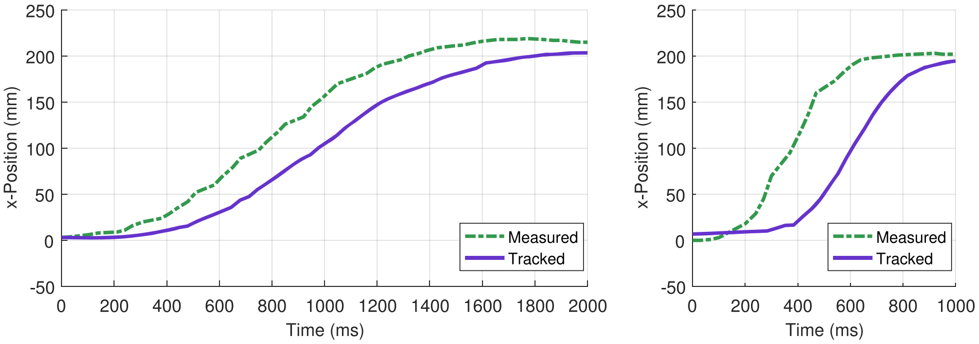

Fig. 2. Typical tracking delay observable in index cursor positioning as derived from the reported hand location. The step movement was generated by moving the hand approximately 200mm along a single axis. Measured position was approximated from the on-device video recording. Tracked position was logged and synchronised with the video. Left plot shows a slow hand movement lasting approximately 2 seconds. Right plot shows a fast hand movement lasting approximately 1 second.

user action was pressing the *SPACE* key (and hence, a decode), then pressing *BACKSPACE* would remove the whole previous word allowing the user to re-enter the desired word from scratch. The *SPACE*, *DONE*, and *BACKSPACE* keys were always treated deterministically in that user selections immediately activated their corresponding behaviours (as opposed to being inferred). For this reason, the three control keys are distinctly separated from the 27 character keys as shown in Figure 1.

The experiment application runs on the HoloLens and communicates with the decoder over a dedicated wireless network. This system architecture was chosen to enable concise separation of functionality and support parallel development. There are, however, no known obstacles to performing the decoding step on-device, and this will be an objective of future work.

4.3 Virtualised Touch Key Selection

The VISAR keyboard seeks to minimise learning time (*SP 6* in Section 3) by exploiting an interaction technique that is compatible with people's pre-existing keyboard typing skills and experience. The HoloLens provides access to the hand position, as tracked by the on-device sensors. The documentation does not explicitly define the point being returned as the tracked hand position but visual inspection suggest that it approximates the centre of the dorsum of the hand, i.e., surface opposite the palm. No hand orientation information is available. We exploit the tracked hand position to place a cursor approximating the tip of the index finger. It is important to understand that the pointer finger is not tracked and so this cursor placement is only approximate. The cursor remains at a fixed offset and orientation with respect to the tracked hand position, and does not adjust for joint articulation or hand orientation changes.

The tracked hand position visibly lags behind the true hand position. The typical lag in hand position tracking (approximately 220ms) as experienced by the user in this study is shown in Figure 2. Nevertheless, the reported tracked hand position shows good robustness to pose and joint articulation changes. The lag also appears to be largely constant, allowing the user to accommodate the delay in their pointing behaviour. Although testing an indirect control task, Hoffmann (1992) suggests that the transition between a continuous and a 'move-and-wait' control strategy occurs at around 700ms. Chung et al. (2011) evaluate actual hand movements under time delay in a virtual environment. They find that below 440ms, the target width dominates the movement amplitude in determining movement time under delay. The application of the decoder to mediate

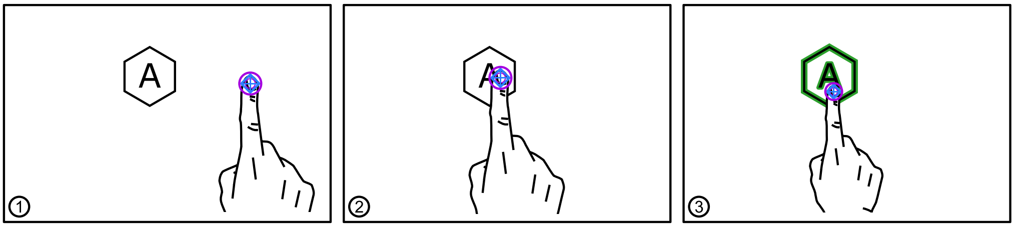

Fig. 3. Virtualised touch driven key selection sequence. (1) The hand is in the ready position with the index cursor showing. (2) The user moves their hand to the desired key location. (3) The user moves their hand forward towards the key to generate an intersection between the key and the index cursor. The key is selected.

touch inaccuracies can be thought of as increasing the effective target width of a key, and thus also acts to reduce the detrimental effects of the tracking delay. More generally, provided tracking delay remains within the range associated with a continuous control strategy, the effect of delay can largely be reduced by ensuring key sizes are of a reasonable size.

It is important to highlight that the hand tracking behaviour resulting from the various device limitations is not ideal. An ideal system for single-finger typing would provide robust index finger position and articulation tracking with minimal delay. Superior multi-finger tracking might also enable 10 finger two-handed typing. As the enabling tracking technology develops for integration with head-mounted AR, such enhancements to the system presented here are worthy of examination. Nevertheless, even more advanced technologies are unlikely to deliver perfect tracking in the fully mobile use case with no off-body sensors. There is thus likely to be ongoing demand for intelligent mediation of mid-air touch based interactions. The approach and design presented in this article for touch-driven interaction facilitated by intelligent decoding is sufficiently flexible and extensible to accommodate advancements in the underlying tracking technology. While the current iteration is subject to several key limitations, it nevertheless provides a valuable baseline and foundation for future development.

To perform direct touches on the keyboard, the user moves their hand to generate an intersection between the keyboard plane and the cursor approximating the tip of their index finger. Upon intersection, the user's touch location is indicated by a small circular marker while the nearest key flashes to green then fades back to white. The intersection point is added to the trace point list, and the nearest key is added as a key press. This key selection approach is illustrated in Figure 3.

4.4 Experimentally Driven Design Iteration

The sequence of four experiments reflect several key design iterations of the VISAR keyboard. Each experiment is described in detail in the remainder of this article but as an aid to the reader, we briefly outline the design journey taken with the VISAR keyboard.

Experiment 1 (see Section 5) evaluates the virtualised touch driven approach against an existing text entry method derived from the standard HoloLens system keyboard. This experiment principally seeks to investigate walk-up usability and acceptance (*SP 6* in Section 3) in comparison with an established baseline.

The results of Experiment 1 demonstrate only marginal differences in net entry rate but the time between key selections using the VISAR keyboard is significantly faster. This observation is in alignment with *SP 1* in Section 3 but the failure of more rapid key selections to translate into faster entry rates highlighted a potential design flaw. We hypothesised that this flaw related to the frequency of error correction undertaken by users. One interesting source of errors, and a unique consequence of decoder based touch mediation, is the inability to distinguish between unknown or unusual words and an erroneous input. As an example, one participant correctly input the key sequence 'D-Y-N-E-G-Y' corresponding to the company name 'DYNEGY' but was overridden in the decode step to the word 'SUNG'. This observation, among others, suggested the investigation of methods for allowing users to indicate a literal interpretation of their input sequence.

In broad terms, the user can provide such guidance to the decoder according to two alternative strategies: proactively or reactively. Fortunately, the use of one strategy does not preclude the other and both can be combined to provide multiple correction pathways for the user. A proactive approach may involve the provision of some additional information during input that the decoder can exploit to better distinguish intent. In Experiment 2, we specifically explore a proactive approach to literal input disambiguation. Our implementation takes significant inspiration from Weir et al. (2014) who augment touchscreen text input by incorporating touch pressure as an indicator of the degree of confidence associated with an input event. Instead of pressure, we use a depth-based transition to switch fluidly between input modes in alignment with *SP 5* in Section 3.

In contrast, a reactive approach is used in many conventional mobile keyboards and implemented as a set of several alternative word predictions, one of which may be the as-input literal word. The user may then either opt-out of replacing their literal input with one of the predictions or choose the literal input from among the displayed options. Note that this strategy is actually introduced as part of the addition of word predictions to VISAR and tested in Experiment 4. The results of Experiment 2 show that the proactive strategy employed did not substantially impact the performance of the keyboard but was rated as useful by a majority of participants.

It is not uncommon for users to type with a single finger on their smartphone, tablet, or even on large interactive information displays such as are found in shopping centres. Broad familiarity with single finger typing led us to hypothesise that users may be able to exploit this prior experience, and possibly any ingrained muscle memory, to touch type on the virtual keyboard with reduced visual features. This commonality with the physical interaction paradigm also promotes walk-up usability and acceptance (*SP 6* in Section 3). The exploration of reduced visual features specifically is motivated by the objective of minimising the occlusion of the physical world by virtual content (*SP 3* in Section 3). We found that the majority of participants were in fact able to type effectively even when all key outlines and key labels were removed.

Encouraged by the faster inter-key timings observed in Experiment 1, the flexible design space demonstrated in Experiment 2, and the performance achieved with the minimal occlusion keyboard in Experiment 3, we applied further design refinements to the VISAR system. Most significantly, we added word predictions as per *SP 4* in Section 3, and re-evaluated performance against a similarly revised baseline in Experiment 4. As briefly described above, the implementation of word predictions also supported reactive disambiguation of intent by presenting the literal input as one of the alternative word panels. This functionality is an alternative but complementary reflection of *SP 5* in Section 3. These various design improvements result in higher entry rates and lower error rates with one participant achieving a peak mean typing speed of 23.38wpm in an experimental block.

Finally, we demonstrate the VISAR keyboard in a range of envisaged short text entry tasks conducted by participants while standing and independently moving through a physical environment. Participants achieved tolerable entry rates in these conditions despite very little prior training and indicated good acceptance of the system design.

5 EXPERIMENT 1: SELECTION METHOD EVALUATION

Experiment 1 examines the hypothesis that allowing users to engage with the keyboard through direct touch is more intuitive and will deliver higher text entry rates than a gaze-then-gesture

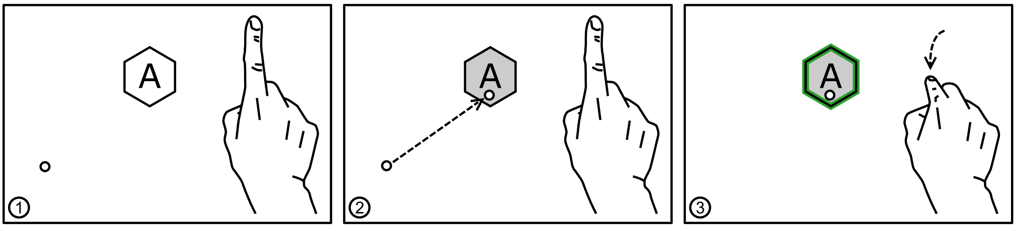

Fig. 4. Gaze-then-gesture key selection sequence. (1) The user is not looking at the key and the gaze cursor is in bottom left of frame. (2) The user looks at the desired key and the key highlights when the gaze cursor enters its region. (3) The user performs the *air-tap* gesture and the key is selected.

interaction. The gaze-then-gesture interaction method is the primary selection paradigm exploited on the HoloLens.

The gaze-then-gesture interaction method leverages the *gaze cursor* for pointing. Strictly, the *gaze cursor* does not reflect the user's eye movements but rather the orientation of the head-fixed frame. The *gaze cursor* is placed at the first intersection of the ray emanating from the head-fixed frame along the principle forward axis. For conciseness and consistency with the Microsoft documentation, we use the term *gaze* in this article to refer to this head-tracked vector at the cost of semantic accuracy. The *gaze cursor* is thus the point of intersection of this vector with objects in the scene.

The gaze-then-gesture interaction method is based on designating focus with the gaze cursor and making a selection using the *air-tap* gesture. The *air-tap* gesture involves placing the hand first in the neutral position with only the index finger raised. The index finger is subsequently pressed down then raised again to indicate a selection. The gaze-then-gesture paradigm is used in the system keyboard provided by default in HoloLens applications. To specify a key, the user focuses the gaze cursor on the desired letter then performs an air-tap gesture to make the selection. The visual appearance of the key in focus changes to provide feedback on which letter will be selected when the selection gesture is performed. The gaze-then-gesture interaction sequence is illustrated in Figure 4.

The gaze-then-gesture based keyboard evaluated in this experiment serves as an established baseline method against which we compare the VISAR system. To provide a valid reference point, we deliver an experience that replicates use of the HoloLens system keyboard while at the same time standardises certain design features for the sake of a meaningful comparison. In particular, we chose not to integrate the decoder with the gaze-then-gesture based keyboard as the underlying paradigm implies a discrete and two-step process of selection then confirmation.

5.1 Method

Experiment 1 is a within-subject experiment comparing the two conditions:

- Baseline: Gaze-then-gesture interaction in which user moves the gaze cursor to the desired key, then performs the *air-tap* gesture with their index finger to make the selection.
- VISAR: Error-tolerant mid-air touch keyboard in which the user moves their hand to generate an intersection between the index finger cursor and the virtual keyboard plane to type a key. After the entry of each word, the *SPACE* key activates the decode method to replace the literal entry with the most probable word.

The appearance of the Baseline and VISAR keyboards as viewed through the HoloLens are shown in Figures 5 and 6, respectively. Note that for all experiments reported in this article, the

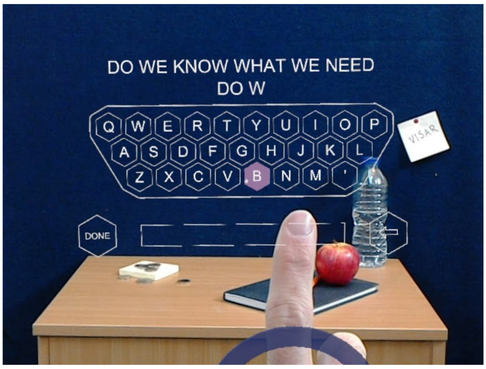

Fig. 5. Appearance of the Baseline keyboard condition as viewed through the HoloLens.

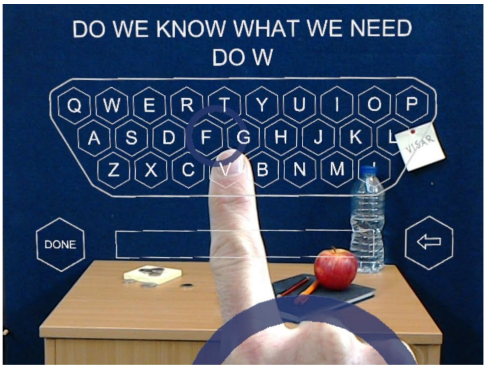

Fig. 6. Appearance of the VISAR keyboard condition as viewed through the HoloLens.

potentially confounding variable of background scene colouration and clutter was controlled for by seating participants in front of a flat colour poster board. The default sizing and distance of the system keyboard is approximately replicated for the Baseline condition: The keyboard is placed at a distance of 1.2m, and the apparent key diameter was set to approximately 45mm. The VISAR keyboard was placed within reaching distance (approximately 0.5m) and scaled down to fit within the display such that the apparent key diameter was approximately 22mm.

As described in Section 4.2, selecting the *BACKSPACE* key on the VISAR keyboard would either remove the previous character or the previous word depending on whether a decode step was previously triggered. Given that no decodes were performed in the Baseline condition, the *BACKSPACE* key would always just remove the previous input key.

We recruited 12 participants for a single 1-hour session (4 female, 8 male). Participants received a £5 Amazon voucher in compensation for their time. We briefed participants on the experimental protocol and fitted them with the AR headset. The order of the two conditions was counterbalanced.

Participants were instructed to type provided phrases as quickly as possible while maintaining low error rates. To compute entry rate in wpm, we measured entry time from the first key press until the selection of the *DONE* key to submit the sentence. The numerator, i.e., the effective word count, is calculated based on the entered phrase length minus one (since entry time starts at first key press) divided by a nominal word length of five characters. We measured the error rate using CER. CER is the minimum number of character-level insertions, deletions and substitutions required to transform the response text into the stimulus text, divided by the number of characters in the stimulus text. After selecting the *DONE* key to submit, participants would see a brief dialog showing their entry rate (wpm) and their CER for the phrase just entered. We instructed participants that more care should be taken if their reported error rate was consistently above 10%.

The stimulus sentences were taken from the memorable phrases subset of the Enron mobile message dataset (Vertanen and Kristensson 2011). The 200 sentences in the memorable phrases subset were filtered to those with 40 or fewer characters, 4 words or more, and that contained only the letters *A* to *Z* and apostrophe. The character limit was imposed to ensure that all phrases would appear on a single line and within the visible display region at the nominal keyboard placement location. The word limit was imposed to ensure a base length and complexity in stimulus phrases. The letter constraints were necessary to ensure all phrases could be typed using the simplified keyboard layout. All sentence terminating punctuation was removed. The resulting set used in Experiment 1 contained 90 distinct phrase. As described in Section 4.1, the decoder incorporated

Table 1. Entry Rate (wpm) and Character Error Rate (CER%) Descriptive Statistics from Experiment 1

Condition	Entry rate (wpm)	Error rate (CER)
Baseline	5.86 ± 1.12 [3.92, 7.48]	1.10 ± 1.30 [0.00, 3.75]
VISAR	6.45 ± 1.83 [3.54, 9.60]	1.40 ± 1.48 [0.00, 4.59]

Results show mean ±1 standard deviation [min, max].

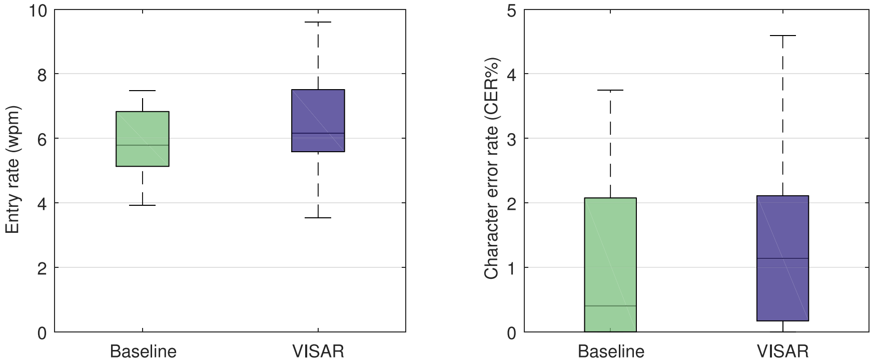

Fig. 7. Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 1.

a vocabulary of 64,000 words. The out of vocabulary percentage for the Experiment 1 phrase set was 0.56%.

At the start of each condition, the participant was instructed to type five practice sentences. During this time they were encouraged to ask questions and make sure they understood the interaction mechanism and keyboard functionality. Upon completing the practice phase, users began typing sentences taken at random from the stimulus set. The test phase would run for 15 minutes of cumulative entry time (sum of time between first key press on a new phrase to selection of the *DONE* key).

5.2 Results

We report results on only the phrases entered during the 15 minute test period (the initial five practice sentences are excluded from the analysis). Unless otherwise specified, we performed statistical significance tests using within-subjects repeated measures analysis of variance at an initial significance level of $\alpha = 0.05$ for entry rate and using Friedman's test for error rate (since errors are count data). We made adjustments for multiple comparisons using Holm-Bonferroni correction.

The group descriptive statistics in each condition are shown in Table 1. Figure 7 presents boxplots of both entry rate and CER. The mean entry rate in VISAR is faster than Baseline (6.45 versus 5.86wpm), however, the result is not significant ($F_{1,11} = 2.160$, $\eta_p^2 = 0.164$, $p = 0.170$). Participants achieved acceptable error rates in both conditions, although Baseline yielded marginally higher accuracy. This difference was not statistically significant ($\chi^2(1) = 1.600, p = 0.206$).

Although the net performance difference between the two conditions is not significant, it is useful to examine the learning effect associated with each interaction technique. A text entry method that is intuitive and easy to gain proficiency in is more likely to gain traction as highlighted by *SP 6* in Section 3. Figure 8 shows the boxplots of participant entry rate corresponding to the beginning, middle and end of the 15 minute test block. Interestingly, the boxplots illustrate a more marked improvement in entry rate between the first and last interval in the VISAR condition compared with the Baseline. The increase in mean entry rate between the first and last intervals

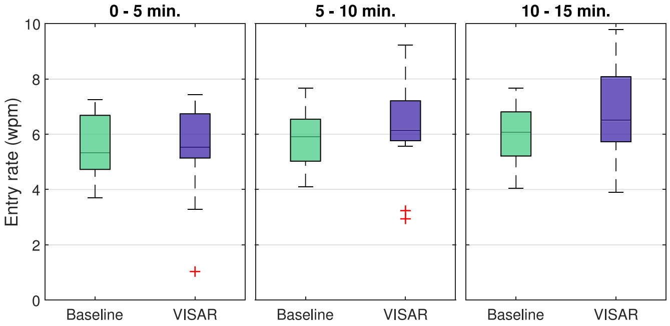

Fig. 8. Boxplots of entry rate (wpm) over time intervals 0–5 minutes, 5–10 minutes and 10–15 minutes in Experiment 1. Red crosses indicate outliers based on $q_1 - 1.5 \times (q3 - q1)$.

Table 2. Median Questionnaire Response to Questions 1 to 3 in Experiment 1

Statement		BASELINE	VISAR
Q1	The keyboard made it easy to type quickly.	2.5	4.0
Q2	The keyboard made it easy to type accurately.	4.0	2.0
Q3	The keyboard was comfortable to use.	3.5	3.0

Responses were recorded on a five-point Likert scale from 1-strongly disagree to 5-strongly agree.

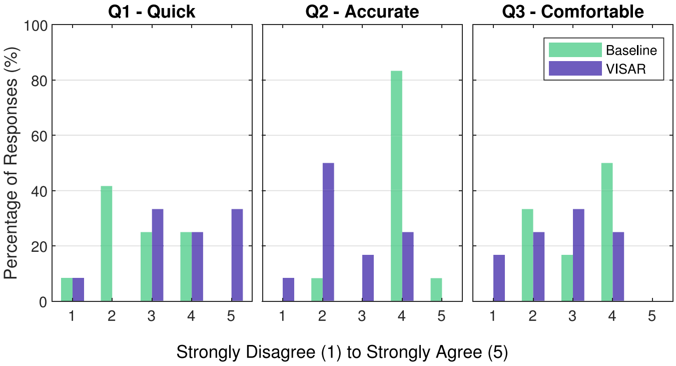

Fig. 9. Distribution of participant responses to Experiment 1 questionnaire. The question statements Q1–3 are defined in Table 2.

is 19.7% for VISAR versus 8.1% for the BASELINE. Also observable in Figures 7 and 8, however, is a larger variance in participant entry rate compared with the BASELINE. This suggests that individual user characteristics may have a more prominent influence on performance due to certain attributes of the VISAR keyboard design.

At the completion of both conditions, participants responded to three statements in a short questionnaire targeting their experience with the two conditions. The three statements are included in Table 2 and examined perceptions of typing speed, accuracy and comfort. Responses were recorded on a Likert scale from 1-strongly disagree to 5-strongly agree. The full response distribution is shown in Figure 9 while median scores are presented in Table 2.

The condition effect on these responses is examined using a Wilcoxon signed rank test. We observe that the participant median perception of typing speed was significantly higher in the

Table 3. Inter-key Timing (s) Descriptive Statistics from Experiment 1

Condition	Inter-key timing (s)
Baseline	2.01 ± 0.40 [1.57, 2.86]
VISAR	1.66 ± 0.47 [1.06, 2.74]

Results show mean ± 1 standard deviation [min, max].

Table 4. Entry Rate (wpm) Descriptive Statistics Based on Phrases Not Requiring Whole-Word Deletions in Experiment 1

Condition	Minor revision entry rate (wpm)
VISAR	7.13 ± 2.24 [3.54, 10.63]

Results show mean ± 1 standard deviation [min, max].

VISAR condition ($Z = -2.365$, $p = 0.018$). This perception is consistent with actual performance in that entry rates were on average higher in the VISAR condition though not significantly so.

The participant median perception of accuracy was significantly higher in the Baseline condition ($Z = 2.889$, $p = 0.004$). This perception is also consistent with actual performance in that lower (though not significantly lower) mean error rates were observed in the Baseline condition. There is no significant difference in participant perception of comfort although several participants did comment on some shoulder discomfort in the VISAR condition.

Despite the marginal difference observed in raw entry rate, during the experiment we observed the rate of key presses appeared faster in the VISAR condition. This suggested an analysis of inter-key timing, that is, the time between discrete key selections. We computed the mean inter-key timing per participant based on the inter-key times across all test phrases, including presses of control keys and subsequently deleted letters. We examined inter-key timing as a proxy for the upper-bound entry rate potential of the interaction method, independent of error rate.

Table 3 summarises the group inter-key timing results. The inter-key timing was 17.4% faster in VISAR and this difference was statistically significant ($F_{1,11} = 7.600$, $\eta_p^2 = 0.409$, $p < 0.05$). This result suggests that, while the VISAR text entry method supports more rapid key selection, it loses significant speed due to the frequency of re-corrections required due to incorrect decoder results. Recall that pressing backspace in the VISAR condition after the decoder prediction was returned would result in the whole word being deleted. This was a simple solution intended to allow users to quickly retype mistaken or incorrectly decoded input. Clearly such actions introduce a corresponding performance penalty. The mean frequency of whole word deletions across participants in Experiment 1 was 7.8 and the median was 6. If we assume an error free entry rate of 7wpm, 6 whole word deletions is approximately equivalent to a time penalty of 50 seconds, or roughly 5% of the total experiment duration. While improvements to the error correction procedure are certainly necessary, the inter-key timing result is promising in terms of *SP 2* as described in Section 3 which suggests that rapid selection may ultimately help realise faster entry rates.

While re-corrections are an unavoidable reality in any recognition-based approach, it is worth evaluating the upper-bound potential for the VISAR method that might be achieved through decoder and/or interface improvements such as the provision of a literal entry fall-back method. To evaluate this potential, we recomputed the mean entry rate for participants after removing entries in which one or more whole-word deletions occurred. This analysis can provide some insight on the upper-bound potential of the method among the novice participant group. Table 4 shows the

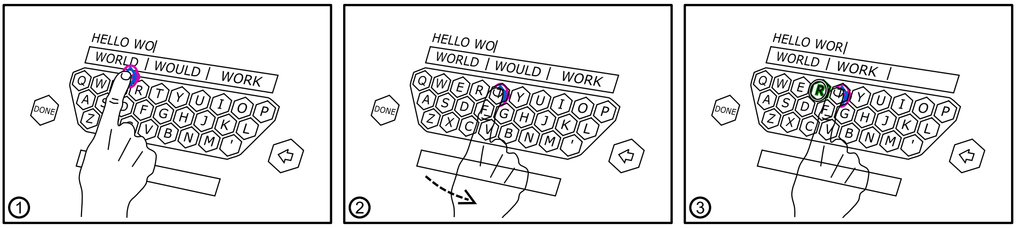

Fig. 10. Precision key selection sequence. (1) Hand is in ready position with index cursor showing. (2) User moves hand so that index cursor is inside the keyboard plane. (3) After 1 second, a secondary cursor appears attached to the keyboard plane. The user moves their hand to place the secondary ring cursor over the desired key. The key is selected based on a 1 second dwell period.

results based on this post-analysis. The difference in entry rate between all entries using VISAR and only entries without whole-word deletions is statistically significant ($F_{1,11} = 10.006$, $\eta_p^2 = 0.476$, $p < 0.01$). There is no parallel to a whole word deletion in the Baseline condition, and therefore it is not included in Table 4.

6 EXPERIMENT 2: FLUID FALL-BACK TO PRECISE KEY SELECTION

Experiment 1 showed that users on average typed individual keys faster using VISAR but the overall text entry rate was not significantly improved due to the need to correct errors. In Experiment 2, we investigate how to mitigate this problem by allowing users to seamlessly combine inferred and literal keyboard entry. We achieve this by providing users with an optional fall-back method for precisely selecting keys as informed by *SP 5* in Section 3. This *precision selection mode* can be optionally activated by users and does not interfere with the normal direct touch interaction of VISAR. Letters entered using this mode are not subject to change during the decode step.

6.1 Implementing Precise Key Selection

To activate the precision key selection mode, the user pushes the index finger cursor into the keyboard and holds it on the far side of the keyboard plane. After one second, a new ring cursor appears, attached to the keyboard plane. The user can make fine adjustments to move the ring cursor so that it highlights their desired letter. The ring cursor is held on the desired key for a further one second period until the selection is confirmed by an *accept* tone. If only a single key selection is desired, the user can then retract their hand such that the pointer finger cursor exits from behind the keyboard plane and the interaction method returns to standard discrete touches. If multiple key selections are desired, subsequent letters can be chained together by dragging the ring cursor to a new key. This allows the user to completely specify a series of letters or an entire word without having to reactivate the precision selection mode.

The precision selection mode applies only to the letters *A* to *Z* and apostrophe. Making a precise key selection using this feature informs the word correction decoder that the specified letter cannot be changed or deleted, that is, the individual letter key selection has 100% certainty. During the briefing of the experiment, we demonstrated this functionality to participants via a video and informed them that they could specify any or all letters in a word in this manner. The precise key selection interaction process is illustrated in Figure 10.

6.2 Method

A further 12 participants were recruited for a single 2-hour session (2 female, 10 male). None of the participants had taken part in Experiment 1. Note that the session time was split between

Table 5. Entry Rate (wpm) and Character Error Rate (CER%) Descriptive Statistics from Experiment 2

Condition	*Entry Rate (wpm)*	*Error Rate (CER)*
VISAR WITHOUT FALL-BACK	8.67 ± 1.05 [6.45, 10.09]	2.01 ± 1.90 [0.21, 6.42]
VISAR WITH FALL-BACK	8.34 ± 1.74 [6.43, 11.13]	1.36 ± 1.55 [0.00, 4.40]

Results show mean ± 1 standard deviation [min, max].

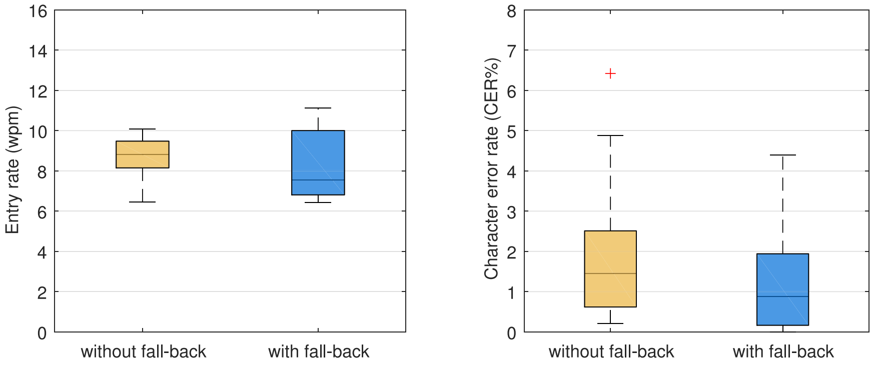

Fig. 11. Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 2. Red cross indicates outlier based on $q_3 + 1.5 \times (q3 - q1)$.

execution of Experiment 2 and Experiment 3 (described later in Section 7). Participants always carried out Experiment 2 before Experiment 3. We compensated participants with a £20 Amazon voucher for their time.

Experiment 2 examined the impact of providing a precise key selection fall-back method on text entry performance. This was a within-subject design with two conditions:

- **VISAR WITHOUT FALL-BACK OPTION**: Participants touched keys to type out the phrase. The word correction decode was triggered when the participant touched the *SPACE* key. There was no provision for specifying that particular letters should be unchanged by the correction step.
- **VISAR WITH FALL-BACK OPTION**: Identical to the previous condition but with addition of the optional precise key selection mode.

The decoder's parameters were re-tuned prior to Experiment 2 based on the trace logs captured as part of Experiment 1. A new distinct set of 115 stimulus phrases was extracted from the Enron dataset for Experiments 2 and 3. Again all phrases were constrained to be 40 characters or less, four words or more, and containing only the letters *A* to *Z* plus apostrophe. The out of vocabulary percentage for the phrase set in Experiments 2 and 3 was 0.54%.

The order of the two conditions was counter-balanced. Participants began each condition by entering five practice sentences. After completing the practice phase, the test phase began and participants were presented with stimulus phrases in random order. The test phase ran for 15 minutes of cumulative entry time. Participants were encouraged to take a five minute break before moving on to the next condition.

6.3 Results

Participant entry rates (wpm) are summarised in Table 5 and Figure 11. The precise key selection fall-back modality was used at least once by 11 out of the 12 participants. The difference in entry

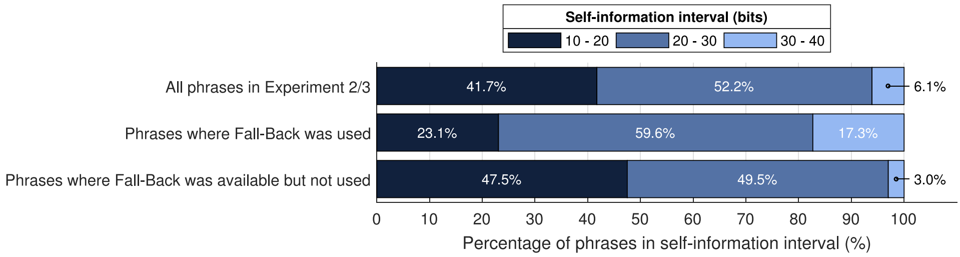

Fig. 12. Precision fall-back usage profile according to phrase self-information interval.

rate between the with and without conditions was negligible and not significant ($F_{1,11} = 0.379$, $\eta_p^2 = 0.033$, $p = 0.551$). The very low effect size suggests that the provision of the precision fall-back functionality is unlikely to have significantly influenced text entry performance.

Table 5 also shows the CER% descriptive statistics. The mean CER is approximately 30% less in the VISAR WITH FALL-BACK OPTION condition, however, this difference is not significant based on a Friedman's test ($\chi^2(1) = 3.000$, $p = 0.083$). It is thus useful to inspect the median CERs in the two conditions, which were 1.46% and 0.88% for without and with fall-back respectively.

The fall-back method was used on average 5.1 (median = 2.5) times on distinct words by participants. Out of those usages, 75.4% were pre-emptive in that there was no prior word correction failure to prompt the need for precision input. In other words, participant's most often employed the precision fall-back method when they expected the decoder to fail to return the correct word.

We were also interested in investigating whether the performance of the fall-back method was affected by the average uncertainty (information entropy) of a phrase. We conjecture that the fall-back method is likely to be more useful for phrases with higher self-information since these are harder to predict by the decoder. We determine the self-information I, expressed in bits, of a phrase by computing

$$I = \log_2(1/P), \tag{1}$$

where P is the probability of the phrase under the decoder's character language model.

Figure 12 shows the usage profile of the precision fall-back method according to the self-information of the stimulus phrase set. It lists the percentage of phrases that fall within a given self-information interval for: bar 1) all phrases in the Experiment 2/3 phrase set, bar 2) phrases where the fall-back method was used, and bar 3) phrases where the fall-back method was available but not used.

The usage profile indicates that the fall-back method was used more frequently in the mid and high self-information phrase intervals and less frequently in the low self-information phrase interval. This is an intuitive result given the intended purpose of providing a precision fall-back method is to assist in typing unusual or out-of-vocabulary words (i.e., words that would increase the self-information of a phrase) that might otherwise be falsely corrected by the decoder.

Figure 13 shows the entry rate and CER for the VISAR WITHOUT FALL-BACK OPTION condition and the subset of the VISAR WITH FALL-BACK OPTION condition where the fall-back method was actually used. The phrase sets are binned according to the three intervals of phrase self-information and the average entry rate and character rate are computed. In Figure 13, we see that the entry rate is distinctly lower than that observed in the VISAR WITHOUT FALL-BACK OPTION condition when the fall-back method is used in the low (10–20) and mid (20–30) phrase self-information intervals. This can be explained by the inherent time penalty introduced by the dwell period required to activate and select in the precision fall-back mode. However, the entry rate is only marginally slower in the VISAR WITHOUT FALL-BACK OPTION condition subset for phrases in the high (30–40)

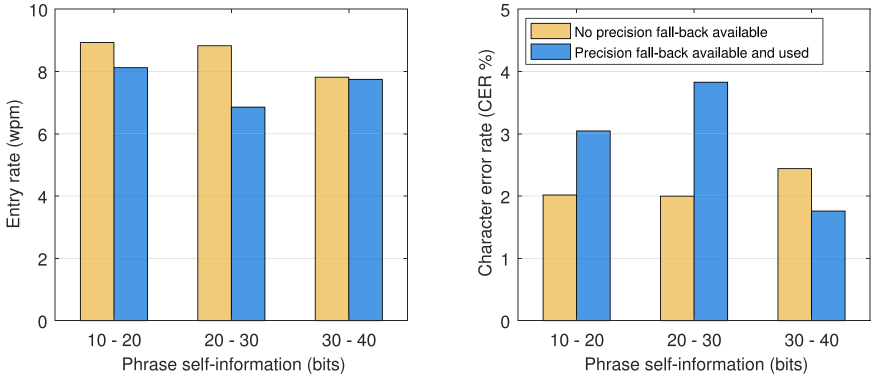

Fig. 13. Text entry performance in the VISAR without Fall-Back Option condition compared against the subset of phrases typed in the VISAR with Fall-Back Option condition where the fall-back method was actually used. Entry rate (wpm) (left) and CER% (right) for phrases binned according to self-information.

Table 6. Median Questionnaire Response in Experiment 2 on a Five-Point Likert Scale from 1-strongly Disagree to 5-strongly Agree

Statement		Median response
Q1	The keyboard made it easy to type quickly.	3.0
Q2	The keyboard made it easy to type accurately.	3.5
Q3	The keyboard was comfortable to use.	3.0
Q4	The precision selection method was useful.	4.0

self-information interval. This result suggests that the fall-back method can assist in maintaining entry rates when difficult phrases are encountered.

The CERs shown in Figure 13 indicate that the precision fall-back method was not always effective at reducing errors. Indeed, we observed that some participants would mistakenly add additional characters using the precision fall-back method, and (by design) these would then not be corrected by the decoder. Figure 13 does, however, suggest that the fall-back method was effective at reducing errors in the high self-information interval (30–40). The results presented in Figure 13 highlight that further refinement is required to ensure the fall-back method can be reliably leveraged by users. Additional training and practice in use of the method is also likely to improve performance. Nevertheless, the performance observed in the high phrase self-information interval does suggest that the precision fall-back method can help to reduce error rates without a significant cost to entry rates.

Participant subjective feedback was obtained in the form of a summative questionnaire which gauged overall impressions of the keyboard and queried specific distinctions where relevant. This short questionnaire was completed after participants finished both Experiment 2 and 3. The statements relevant to Experiment 2 are summarised in Table 6 along with the participant median response. The implementations of VISAR with and without Fall-Back differed only in the provision of the fall-back functionality. For this reason, we considered it reasonable to examine participant's overall impressions of using the keyboard (Q1–3) and then specifically target their experience of the fall-back functionality (Q4). Responses were recorded on a Likert scale from 1-strongly disagree to 5-strongly agree. The full distribution of responses are shown in Figure 14.

The responses to Q1–3 provided limited information in an absolute sense due to the lack of an alternative condition to compare against. The response to statement Q4 indicates that participants

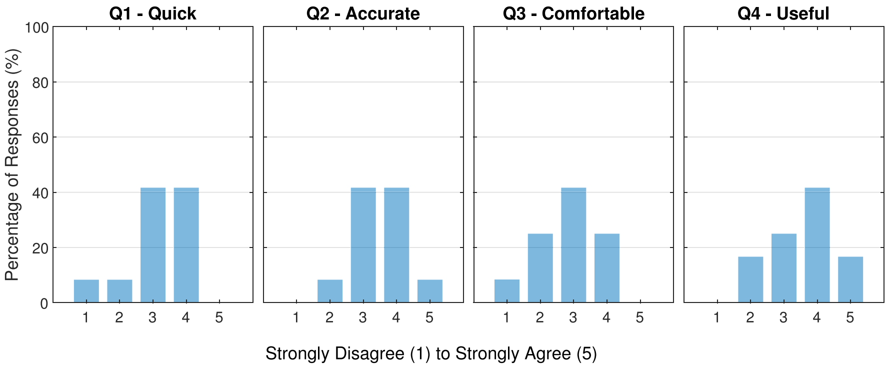

Fig. 14. Distribution of responses to Experiment 2 questionnaire. The question statements Q1–4 are defined in Table 6.

generally found the fall-back method to be useful (58.3% indicated agree or strongly agree). Only two (16.67%) participants specifically felt that it was not useful.

In summary, the provision of the fall-back mechanism did not deliver the increase in entry rate that we anticipated. Clearly there are other aspects to the task of dealing with unusual vocabulary and error corrections that require further investigation. Nevertheless, the negative impact on entry rate is negligible (<4%) and the questionnaire results indicate that it was considered useful by the majority of participants. This suggests that the precision fall-back approach delivers some valuable functionality and may ultimately serve as a useful component within a more complete suite of error correction interactions. We conclude that the fall-back method does not adversely affect entry rates and can help to reduce error rates when employed effectively.

7 EXPERIMENT 3: MINIMISING KEYBOARD OCCLUSION

In Experiment 3 we investigate the implications of solution principle *SP 3* which is critical for AR HMD text entry—minimising field-of-view occlusion. By reducing the number of visual features of the keyboard which are displayed in the HMD, the user can focus their attention on the AR scene, rather than on the text entry interface. We hypothesise that we can easily train users so that they can type using the VISAR keyboard by just providing an outline of the keyboard and hiding both the outlines of the individual letter keys and the labels on the individual keys.

7.1 Method

As described in Section 6.2, Experiment 3 was conducted with the same participant group as Experiment 2 but in the second half of the participant's 2 hour session. The same phrase set covered both experiments, with stimulus phrases randomly presented without replacement. To manage fatigue, participants were encouraged to take a 5 minute break between conditions.

Experiment 3 was a within-subject design with two conditions:

- **VISAR Reduced Occlusion**: Letter labels were removed from keys as shown in Figure 15. All other keyboard features remained the same. The optional precision fall-back method was also available.
- **VISAR Minimal Occlusion**: Identical to the previous condition but with the hexagonal key outlines also removed as shown in Figure 16.

The order of conditions was fixed to deliberately exploit the learning effect associated with a sequential reduction of visual features, i.e., participants would perform the test in the VISAR

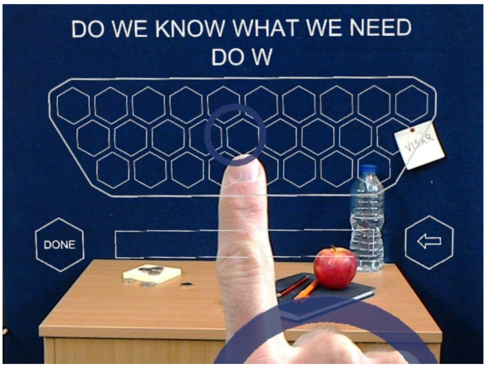

Fig. 15. The VISAR Reduced Occlusion condition with no key labels.

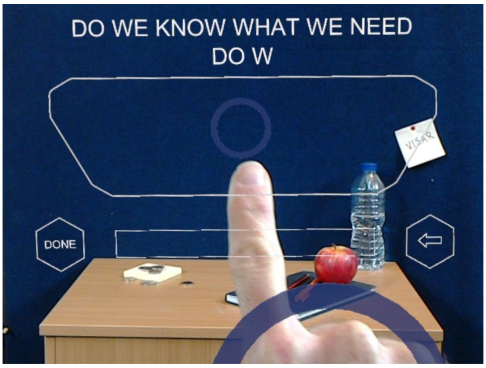

Fig. 16. The VISAR Minimal Occlusion condition with no key labels or outlines.

Table 7. Entry Rate (wpm) and Character Error Rate (CER%) Descriptive Statistics from Experiment 3

Condition	Entry rate (wpm)	Error rrate (CER)
VISAR Reduced Occlusion	10.39 ± 2.48 [7.02, 14.90]	2.19 ± 1.64 [0.00, 4.57]
VISAR Minimal Occlusion	10.56 ± 2.59 [6.52, 15.26]	2.91 ± 1.97 [0.28, 5.54]

Results show mean ± 1 standard deviation [min, max].

Reduced Occlusion condition first and the VISAR Minimal Occlusion condition second. The keyboard outline was held constant across all conditions, as was the position and visual appearance of the *BACKSPACE*, *SPACE*, and *DONE* keys.

Users continued to receive visual feedback of the detected touch intersection with the keyboard plane via the small circular marker indicating the most recent touch location. In the VISAR Reduced Occlusion condition, the nearest key outline would flash to green then fade back to white. No letter labels were shown in either condition in response to the basic touch event. Activation of the optional precision fall-back method would show the letter label and outline of the key currently in focus only. The key would fade back to its original visual configuration (depending on the condition) upon change of focus or deactivation of the precision fall-back mode.

7.2 Results

The key performance metrics for the two conditions in Experiment 3 are presented in Table 7. The difference in text entry rate between the reduced and minimal conditions is marginal and not significant ($F_{1,11} = 0.428$, $\eta_p^2 = 0.037$, $p = 0.526$). The reader is reminded that condition order was not balanced in this experiment and so the asymmetric learning effect represents a second plausible explanatory variable. The significance tests presented in this section thus reflect the coupled influence of minimising visual features and additional practice. The null result is still interesting as the practical application of a minimal occlusion interface would likely be introduced with a similar process of staged reduction of visual features. The entry rate result suggests that minimal visual features combined with additional practice yields performance largely indistinct from the keyboard with just key labels removed.

Interestingly, out of the 12 participants who completed the four conditions in Experiments 2 and 3, 10 achieved their highest entry rate performance in either the minimal or reduced occlusion configuration despite the lack of visual features. The maximum mean entry rate across all

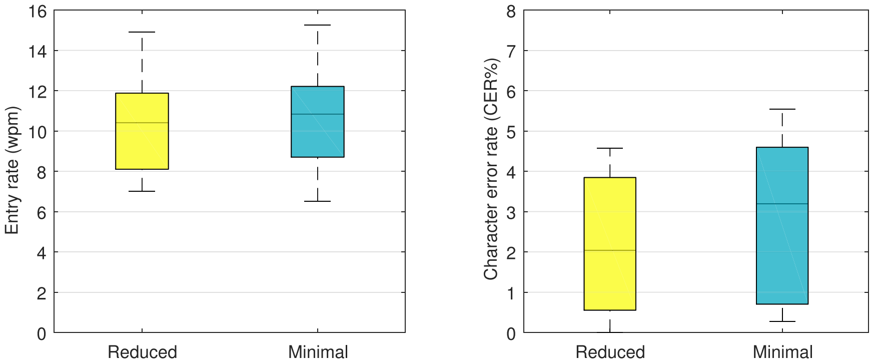

Fig. 17. Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 3.

Table 8. Median Questionnaire Response in Experiment 3 on a Five-Point Likert Scale from 1-Strongly Disagree to 5-Strongly Agree

Statement		Median response
Q5	It was still possible to type effectively without key labels.	5.0
Q6	It was still possible to type effectively without key labels and outlines.	5.0

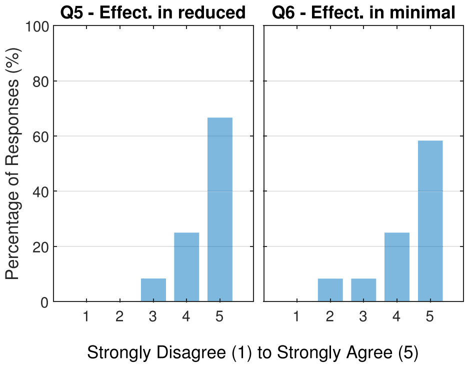

Fig. 18. Distribution of responses to Experiment 3 questionnaire statements Q5–6 as defined in Table 8.

the conditions that comprised Experiments 2 and 3 was achieved by participant 6 in the VISAR Minimal Occlusion condition: 15.26wpm with a CER of 0.35%. Figure 17 provides a boxplot of these performance metrics.

There was a small error rate difference between reduced and minimal occlusion, though not significant ($\chi^2(1) = 3.000, p = 0.083$). As a point of comparison, VISAR with no reduction in visual features in Experiment 2 yielded a mean CER of 1.36%. Therefore we find that minimising field-of-view occlusion by removing visual keyboard features does increase error rate, however, the resulting error rate with minimal occlusion is still below a tolerable threshold for CER (<5%).

After completing Experiment 3, participants were asked to respond to a short questionnaire. The two statements Q5 and Q6 in Table 8 examined perceived typing effectiveness under the reduced and minimal occlusion conditions. The median responses are presented in Table 8 while the full distributions are shown in Figure 18.

Interestingly, participants were overwhelmingly positive in their self-assessment, with 91.67% either agreeing or strongly agreeing that they could type effectively without key labels. This

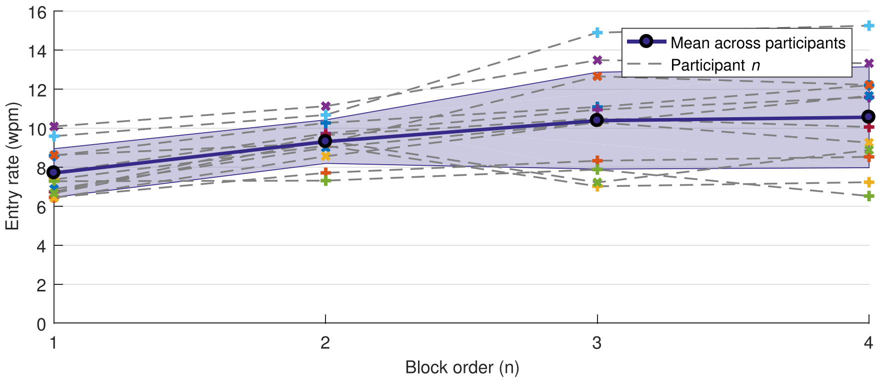

Fig. 19. Mean entry rate (wpm) across participants plotted in chronological order of test block in Experiments 2 and 3, irrespective of condition. Shaded region shows ±1 standard deviation.

proportion was only slightly lower for the statement corresponding to no key labels or outlines (83.33%).

The result that group mean entry rate was highest in the two conditions with the least visual features raises the question: to what extent does learning influence performance improvement? Figure 19 plots the mean entry rate for each condition performed during the single session that comprised Experiments 2 and 3. Recall that Experiment 2 balanced the order of presentation of the with and without fall-back conditions. This has been taken into account, such that Figure 19 shows the results of each participant as they were performed in chronological order and irrespective of keyboard condition. The positive gradient observable across blocks 1 to 3 indicates the presence of a distinct learning effect and suggests a correlation between practice and performance.

8 EXPERIMENT 4: DESIGN ITERATION AND EVALUATION UNDER EXTENDED USE

This section describes several refinements of both the VISAR and Baseline keyboard interaction methods and interface designs. The main design change involves the addition of word predictions based on the current input text. This design modification stems from *SP,4* described in Section 3. The revised VISAR and Baseline conditions are subsequently evaluated in an extended-use experiment in which participants are exposed to each keyboard condition over a 1.5 to 2 hour session. The revised conditions are subsequently referred to as Baseline* and VISAR*.

8.1 Word Predictions and Decoder Refinement

The relatively low mean entry rates observed in the previous experiments (6wpm in Experiment 1 and up to 10wpm in Experiment 3) suggests that users are inherently rate limited by the two selection methods available. A potential strategy for improving entry rates is the provision of word predictions. Here, we define word predictions as the presentation of the *n* most likely words based on the currently entered characters and sentence context. The user selects from among these presented word predictions to insert the word rather than typing all the remaining characters. Indeed, the system keyboard on the Microsoft HoloLens does provide word predictions and so the investigation of their potential effect is particularly relevant.

A trigram language model was integrated into both keyboard implementations. Preliminary testing indicated that a trigram language model provided comparable predictive power with the HoloLens system keyboard for the typical phrases used in the experiment. This trigram language model was trained in the same manner and on the same data as the 4-gram model used by the

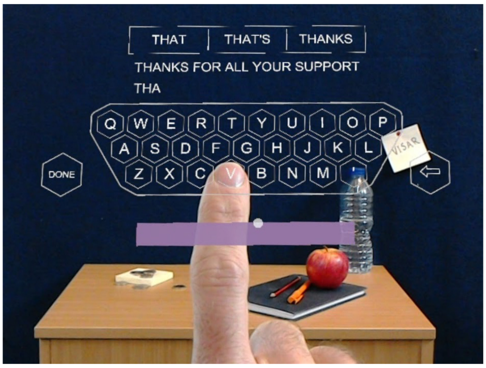

Fig. 20. The Baseline* keyboard condition.

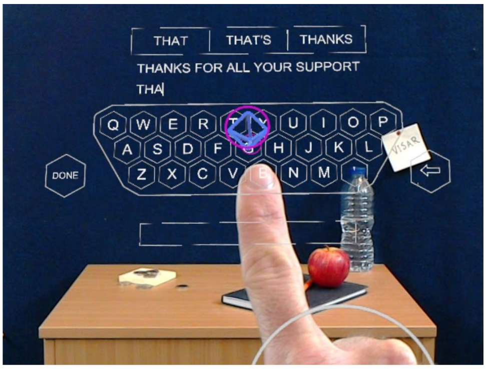

Fig. 21. The VISAR* keyboard condition.

decoder. In our implementation, three alternative word predictions were presented above the keyboard and updated as the user typed. If the user made a selection from among these alternatives, predictions for the next word in the sentence were shown.

The participant entry logs from Experiments 2 and 3 were used to further refine the decoder parameters. Furthermore, we extended the decoder in VISAR* to provide relevant word predictions even under input error. For example, a user might have typing errors in the literal interpretation of the current word's prefix. Our algorithm provides the most probable word predictions taking into account the distribution over possible word prefixes and the language model probability of possible word predictions.

Typing a full word and entering a space on the VISAR* keyboard initiated the standard correction behaviour as described in Section 4.2: the most likely word given the observation sequence replaced the typed string. Importantly, however, when the decoder is activated by entering a space, the VISAR* keyboard temporarily re-purposes the three slots introduced to present the word predictions. The three next most likely decoder results are presented in these slots and can be selected to replace the inserted word. If the literal string typed is not already among these three alternatives, it is included and replaces the least likely of the presented words. This approach enabled users to select and re-insert the literal input in circumstances where the decoder incorrectly replaced the typed string.

8.2 Interface Design Changes

The appearance of the Baseline* and VISAR* keyboard conditions as seen through the HoloLens are shown in Figures 20 and 21, respectively. The following interface changes described were made based on observation of participants using the keyboard and informal qualitative feedback obtained during the previous experiments. Several participants observed that it was at times difficult to be close enough to reach the keyboard comfortably while maintaining a sufficient amount of the keyboard within view. Note that the Microsoft HoloLens provides a somewhat limited display window which means that near objects are prone to extending outside the render region and so appear cut-off. A decision was thus made to bring the keyboard closer to the user so that it was easier to reach, while also reducing it in size so that it was fully rendered. Adding the word prediction selection functionality also required that the position of the *BACKSPACE* and *DONE* keys be adjusted to make more efficient use of the available display region. These changes resulted in an apparent key diameter of approximately 17.5mm and a key layout (keys *A* to *Z* plus apostrophe) of width 175mm by height 525mm.

During previous experiments, several participants also complained that typing on a vertically oriented keyboard plane was uncomfortable after extended use. It was suggested that the keyboard plane might be tilted and lowered to better map with how the hand traverses the space with minimal shoulder movement. This proved an effective suggestion and was incorporated into the revised design. The centre of the top keyboard row was thus positioned relative to the headset origin (a point approximately located slightly in front and above the user's eyebrows) with an offset of 500mm away and 70mm down and the whole layout was inclined at 20°.

The index cursor was also modified from a single flat circle to a pair of circles and a wireframe pyramid (compare the original cursor shown in Figure 6 with the revised cursor shown in Figure 21). This change was made in response to feedback from participants that they had difficulty judging the depth of the original index cursor.

Note that the precision fall-back method introduced in Experiment 2 was also included in the VISAR* condition. It behaved in the same way as described in Section 6.1.

8.3 Method

In this experiment and in contrast with Experiment 1, the size and location of the keyboard was held constant in both conditions. This was done to reduce potential confounding effects associated with interactions between the selection method and the keyboard placement and/or sizing.

The test protocol was also revised from that used in the previous experiments to assess performance over blocks of a fixed number of phrases, rather than fixed time periods. This was done in part to encourage users to maintain high entry rates but also to ensure that participants would type the same number of phrases over the full session in both conditions. Participants would thus type 20 phrases per block for eight blocks to complete one session. Participants were encouraged to take a short break between each block. The eight blocks of 20 phrases that constituted a single session were all completed in the same keyboard condition. The two conditions were thus allocated to their own individual sessions, and were conducted on different days. Sessions were scheduled such that there was no more than one day break between each session. The order of conditions experienced by the participants was counterbalanced. Each session would typically last between 1.5 and 2 hours depending on participant typing entry rates. Participants were compensated with a £30 Amazon voucher for their time.

An introductory familiarisation task was also added to the experiment protocol to ensure participants achieved basic competence with the relevant selection method before beginning to type on the keyboard. The task serves to separate the device and interaction familiarisation from the keyboard familiarisation in order to better isolate the specific learning and performance effects associated with the two test conditions. The task involved selecting targets in a fixed sequence (a simplified circular target acquisition task). Participants were required to select all 10 targets within 15 s. If all targets were not selected within 15 s, the task would reset and repeat until this was achieved to ensure a minimum level of selection method proficiency was achieved. Following the familiarisation task, participants were then instructed to type five practice sentences. As in Experiments 1 and 2, participants were encouraged during this practice period to ask questions and make sure they understood the interaction mechanism and keyboard functionality. Finally, one additional block was added to the VISAR* condition to investigate how predictions and extended use influenced performance in the minimal occlusion configuration.

A total of 340 distinct phrases (20 phrases per block with eight blocks in the Baseline* condition and 9 blocks in the VISAR* condition) were selected from the wider Enron dataset. These phrases were then randomly allocated over the conditions, blocks, and participants. Consistent with previous experiments, all phrases were constrained to be 40 characters or less, four words or more, and containing only the letters A to Z plus apostrophe. The out of vocabulary percentage

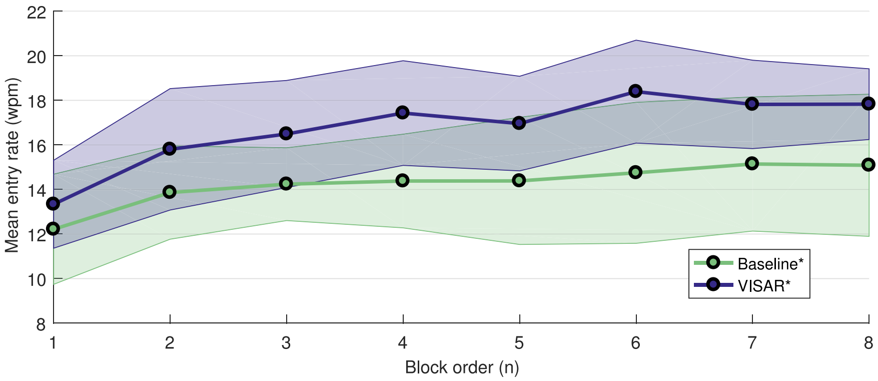

Fig. 22. Mean entry rate (wpm) across participants over the eight experimental blocks. Shaded region shows ±1 standard deviation.

Table 9. Entry Rate (wpm) and Character Error Rate (CER%) Descriptive Statistics from Experiment 4

Condition	Entry rate (wpm)	Error rate (CER)
BASELINE*	14.26 ± 2.12 [11.22, 18.24]	0.36 ± 0.26 [0.02, 0.85]
VISAR*	16.76 ± 1.67 [14.43, 19.11]	0.63 ± 0.38 [0.26, 1.72]

Results show mean ± 1 standard deviation [min, max].

for the phrase set in Experiment 4 was 1.18%. None of the phrases had been used in the previous experiments.

8.4 Results

A new group of 12 participants were recruited for the experiment (7 female, 5 male). None had participated in any of the previous experiments or had any experience with the Microsoft HoloLens.

Figure 22 shows the mean entry rate across participants over the eight blocks in each condition. The shaded region shows the standard deviation across participants in the given block number. The gradient is steepest in both conditions between blocks 1 and 2 then increases more gradually. This suggests a pronounced initial learning effect before a transition to a more gradual performance improvement with increased exposure and experience.

Table 9 shows the descriptive statistics for Experiment 4 following the completion of all eight test blocks. Entry rate and error rate results are also presented in Figure 23. The mean entry rate (wpm) over all eight blocks was 14.26 in the BASELINE* condition and 16.76 in VISAR*. Both methods achieved this with mean CERs under 1%. The difference in text entry rate over all eight blocks between the BASELINE* and VISAR* conditions is significant ($F_{1,11} = 9.014$, $\eta_p^2 = 0.450$, $p < 0.05$).

Another relevant point of analysis is the performance difference once the dominant learning effect has subsided. This is examined by computing entry rates in the final four blocks only. Figure 23 also presents boxplots based on the final four blocks only. The mean entry rate in the final four blocks was 14.84 and 17.75 for the BASELINE* and VISAR* conditions, respectively. This represents a speed increase from the baseline of 19.6%. This difference is significant ($F_{1,11} = 8.237$, $\eta_p^2 = 0.428$, $p < 0.05$).

The highest mean entry rate for a given 20 phrase block for VISAR* was achieved by participant 7 during block 6 with 23.38wpm at a character error rate of 0.24%. The best performing block for BASELINE* was participant 2, also in block 6, with 20.55wpm at an error rate of 0.00%.

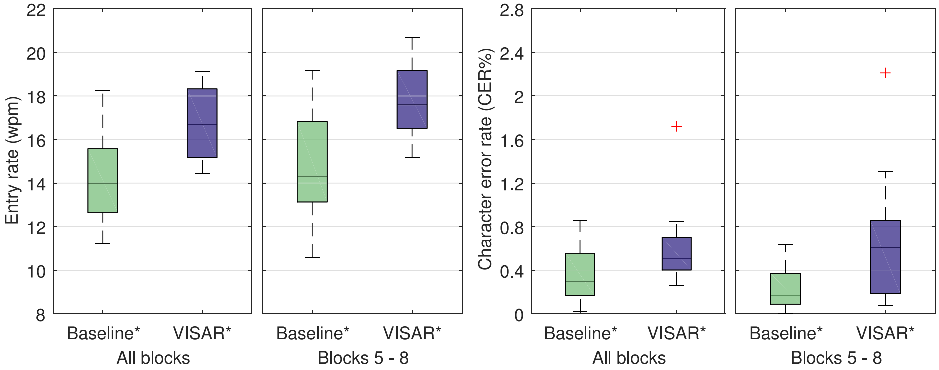

Fig. 23. Boxplots of entry rate (wpm) (left) and character error rate (CER%) (right) in Experiment 4. Plots show performance over all eight blocks as well as in just the final four blocks (blocks 5–8).

In addition to net entry rate, it is useful to examine the underlying efficiency of both selection methods. An interesting first point of comparison is the number of repetitions required to complete the initial target acquisition familiarisation task. Recall that this task required participants to select 10 targets appearing at opposing points on a circle of fixed radius within 15 seconds. The mean number of executions in the gaze-then-click and touch based selection techniques were 7.3 and 2.2, respectively. These results are distorted by some participants who found the air-tap gesture particularly difficult to perform reliably. The median number of repetitions, 5 for gaze-then-click and 2 for touch, is perhaps a better reflection of the relative efficiency of the two techniques. This result is likely a consequence of the following two key factors: (i) the touch driven interaction technique exploits a familiar paradigm allowing users to apply already established motor skills; and (ii) discrete target selection is inherently more efficient and reliable without the use of a hand gesture. The first factor was observed to some degree in both Experiment 1 and Experiment 4, where a steeper learning effect was observed in the VISAR conditions. The second factor is further explored in more detail below through an analysis of input efficiency based on Fitts' law.

Typing on a keyboard may be abstracted to a sequential target acquisition task. Fortunately, we can leverage established analytical approaches to estimating the underlying qualities of a selection technique that are scale independent.

The key log data collected during this experiment was post-processed to extract the key-to-key transitions. Only those input phrases that contained no interim selection errors were included in this analysis. The key-to-key transitions form the basis for estimating throughput according to Fitts' law.

Fitts' law predicts that movement time (*MT*) in making a selection is linearly proportional to index of difficult (*ID*), a non-dimensional metric representing the difficulty associated with a selection.

Thus, movement time can be defined as

$$MT = a + bID, \tag{2}$$

where *a* and *b* are regression coefficients and *ID* is (according to the Shannon formulation of Fitts' law)

$$ID = \log_2\left(\frac{D}{W} + 1\right), \tag{3}$$

where *D* is the movement distance, and *W* is the target (key) width.

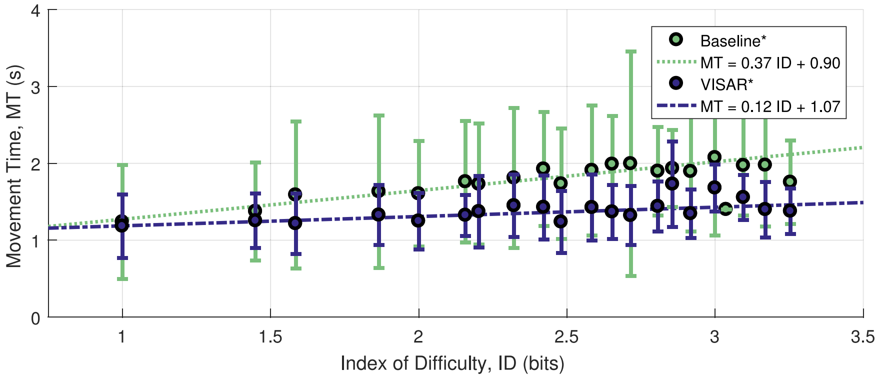

Fig. 24. Movement time (*MT*) versus index of difficulty (*ID*) based on key transitions encountered while typing. Error bars show ±1 standard deviation. The dashed lines show a linear regression of the two conditions.

We seek to extract the movement times associated with the different key transitions encountered while typing (only considering keys *A* to *Z* and apostrophe). The regular tessellation of the keyboard layout means that there are only 23 distinct key-to-key movement distances. Furthermore, the key sequences corresponding to two of the extreme movement distances (e.g., *Q* to *P* and *Q* to apostrophe/*Z* to *P* as well as their inverses) do not occur in the stimulus phrase set. This leaves 21 distinct key-to-key movement distances that form the basis for the *ID* sample points (*W* is constant).

Figure 24 shows the movement time versus index of difficulty for the two keyboard conditions. We compute throughput as $TP = 1/b$, where b is the gradient of the regression line. The computed throughput was 8.26bps for VISAR* and 2.68bps for the BASELINE*. The time taken to make a discrete key selection is clearly an emergent property and influenced by multiple factors such as the human perception, motor control, and processing systems as well as device attributes, such as tracking accuracy and latency. It is reasonable to assume that many of these factors are consistent across both conditions, and this is corroborated by the similar intercept values, a, shown in Figure 24. The difference in slopes (and hence difference in throughput) visible in Figure 24 indicates that the negative effect of increasing task difficulty (*ID*) is lower in the VISAR* condition. In other words, participants could select distant keys nearly as well as nearby keys in the VISAR* condition whereas the BASELINE* condition saw a more prominent negative effect as the distance between keys increased. This may stem for inherently superior motor control of the hand in contrast to head movements. The throughput values reported here should, however, be interpreted with caution given that the free-form typing task does not closely replicate the traditional protocol used in a typical Fitts' law experiment. Furthermore, constraining the analysis to only phrases with completely accurate selections likely inflates the computed throughput values. Nevertheless, the relative magnitude of the two values does provide an indication of the comparative efficiency of the two selection methods.

The character error rates were significantly higher in the VISAR* condition ($\chi^2(1) = 5.333$, $p < 0.05$), although the mean character error rate over all eight blocks was less than 1% for 11 of the 12 participants. Error rates of this magnitude are typically considered tolerable in most text entry tasks. Nevertheless, this result does highlight the speed-accuracy tradeoff typically observed in alternative text entry methods. The specific usage scenario may dictate whether a user is willing to accept a higher error rate for the sake of higher entry rates. It also highlights the importance of error correction and error prevention functionality being considered in parallel with underlying

Table 10. Median Questionnaire Response in Experiment 4 on a Five-Point Likert Scale from 1-Strongly Disagree to 5-Strongly Agree

Statement		Baseline*	VISAR*
Q1	The keyboard made it easy to type quickly.	4.0	4.0
Q2	The keyboard made it easy to type accurately.	4.0	3.0
Q3	The keyboard was comfortable to use.	4.0	2.5

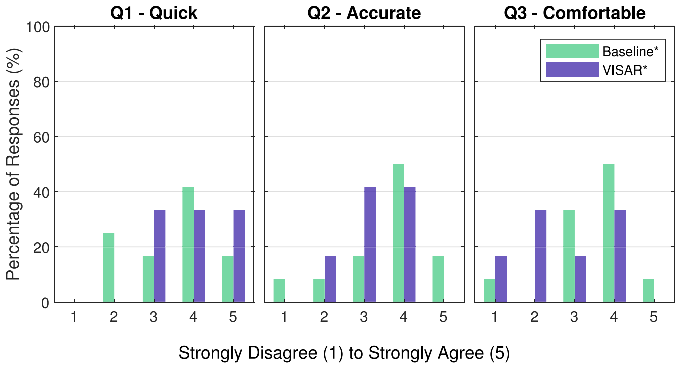

Fig. 25. Distribution of responses to Experiment 4 questionnaire. The question statements Q1–3 are defined in Table 10.

keyboard and interaction design. For example, it was observed that entry errors would occasionally occur when users failed to check the word returned after a decode. Improving the visibility of the decode process through careful interface cues may thus help to further reduce these types of errors.

Although the precision fall-back method was available in the VISAR* condition, it was only used by 5 of the 12 participants. Among these five participants, the fall-back method was used on average 3.0 (median = 1.0) times on distinct words. This rate of usage is a distinct reduction from that observed in Experiment 2, where the average usage per participant was 5.1 distinct words and a median of 2.5. Only one usage of the fall-back method was a response to a decoding failure that the user sought to correct. All other intentional usages of the fall-back method were pre-emptive in that participants had not experienced a prior decoding failure on that word. The reduction in usage of the fall-back method is likely a consequence of both the introduction of word predictions and a reduced explicit emphasis on the feature within the experimental briefing and protocol. Furthermore, we observed that in instances where a possible decode failure was anticipated, several participants found that simply taking more time and care to hit the desired keys was sufficient to correctly type the word.

Participants completed a post-experiment questionnaire targeting impressions of their typing speed, accuracy, and comfort under the two keyboard conditions. The questionnaire statements responded to, and their median responses are presented in Table 10. Note that participants were asked to exclude their experience of the minimal occlusion condition described in the following section when considering their assessment. Figure 25 presents the full distribution of responses to the statements in Table 10. No distinct difference is apparent in terms of typing speed ($Z = -1.730$, $p = 0.084$) or accuracy ($Z = 0.741, p = 0.458$). Participants were generally less willing to agree with the statement that VISAR* was comfortable although the difference from the Baseline* was not significant ($Z = 1.801$, $p = 0.072$). From inspection of Q3 in Figure 25, we see a bimodal distribution of responses for VISAR*. This result is consistent with informal comments from several

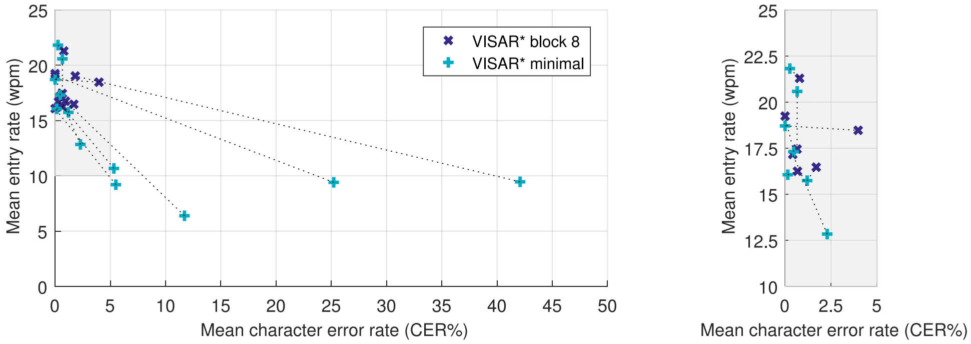

Fig. 26. Mean entry rate (wpm) versus mean character error rate (CER%) for VISAR* in the minimal occlusion configuration and the final block (block 8) of the standard visual configuration. Dashed lines link the two results of individual participants. The left plot shows data points for all participants. The right plot provides an enlarged detail view of the shaded region shown on the left. Only data points for participants with error rates under 5% are shown in the enlarged detail view.

participants that the VISAR* condition caused some discomfort for the shoulder. The final question on the questionnaire asked participants to indicate a preference between the two conditions. VISAR* was preferred by 8 of the 12 participants (67%).

As described in Section 8.3, participants were exposed to one additional block in the VISAR* condition where the visual features of the keyboard were set to the minimal occlusion configuration, i.e., no key outlines or key labels were shown. Figure 26 plots entry rate versus error rate for all 12 participants under this configuration. The performance of each participant in the immediately preceding eighth block of the full visibility VISAR* keyboard condition is also shown for comparison.

The results presented in Figure 26 highlight a distinct split between participants who were unable to effectively use VISAR* in the minimal occlusion configuration and those who were largely unaffected by the removal of key outlines and labels. Seven of 12 participants experienced a reduction in entry rate of between 10.0% and 61.9% against their previous block performance combined with a signification deterioration in character error rate. The other five participants maintained a mean entry rate between 4.4% slower and 17.9% faster than their previous full visibility block while all had character error rates of less than 1.2%. It is suspected that this observation is likely to be correlated with a participant's ability to effectively touch type. It is worth noting, however, that this same distinct split was not observed in Experiment 3. Unfortunately, the pre-experiment experiential survey did not capture touch typing ability, and so investigation of this correlation remains as future work.

We speculate that the introduction of word predictions may also have altered the typing strategy of some users that inadvertently primed them differently for the minimal occlusion block in Experiment 4. The provision of word predictions allows users to obtain near instantaneous feedback on the decoder's best estimates of their intended input. When users can see the keys, they are more likely to touch on or near the intended key. This improves the likelihood that the presented predictions will include the intended word, even if there have been only two or three touches. Under higher levels of input noise such as encountered in the minimal occlusion configuration, more data points (touches) may be required to accurately predict the intended word. However, users accustomed to seeing their intended word among the predictions after very few touches have not built up sufficient trust and confidence in the decoder to continue typing despite apparently erroneous predictions. In contrast, participants in Experiments 2 and 3 were not shown predictions, and so were more likely to type out the full word and rely on the decode step to correct errors.

Positive examples of successful error correction served to reinforce trust and confidence in the decoder. This theorised interaction between interface features and typing strategies requires further investigation. Nevertheless, we do observe that VISAR* does enable a sub-set of the participant group to maintain their entry rate under the minimal occlusion configuration.

9 VALIDATION STUDY

This validation study examines whether VISAR* is a suitable text entry method for typical AR applications. A short experiment, chiefly examining user experience and behaviour, was designed to expose participants to a range of short text entry tasks under conditions relevant to head-mounted AR.

9.1 Method

Four participants were recruited for the study (four male). The experiment session lasted for approximately 1 hour and participants were compensated with a £10 Amazon voucher.

Participants received an introductory briefing before performing the same target acquisition familiarisation task described in Section 8.3. Participants would then also complete five practice phrases while seated (in the same arrangement as described in Section 8.3).

The main exercise in the experiment then required that participants explore the space where they would encounter four different kinds of text entry sub-tasks. Five of each sub-task were presented at locations dispersed throughout the space, resulting in 20 sub-tasks in total.

The four sub-tasks are described as follows:

— Transcription: A short phrase (taken from the Enron dataset) was printed on a page and attached to the wall at the task location. Participants were instructed to transcribe the phrase exactly.
— Description: A simple illustration was printed on a page and attached to the wall at the task location. Participants were instructed to describe the image, e.g., one image showed an image of a man and a woman riding a tandem bicycle.
— Message: A ‘message’ would be received at specific locations in the space asking a simple question. Participants were instructed to respond to the question, e.g., one message asked, ‘What did you have for breakfast?’
— Annotation: Participants were instructed to pick an object at the task location and annotate or describe it, e.g., one free annotation task was located where several portable fire extinguishers were located.

Upon finding a sub-task in the space, participants would select the corresponding virtual marker as shown in Figure 27. This would then bring up the keyboard (see Figure 28) allowing them to complete the task. Participants were encouraged to use at least four words when crafting their text in the three sub-tasks involving composition. Upon selecting *DONE*, the entered text would be submitted, and the keyboard would close.

Participant entry rates were recorded. Following the experiment, participants were also requested to complete a system usability scale (SUS) survey (Brooke 1996) targeting their experience of the system as a whole. A short semi-structured interview was also conducted to obtain participant feedback on the user experience in the spatial annotation exercise.

9.2 Results

The results of this experiment should be interpreted with some caution given the limited number of participants involved. The quantitative results are presented to provide an indication of what might be achievable rather than as an attempt to describe typical user performance.

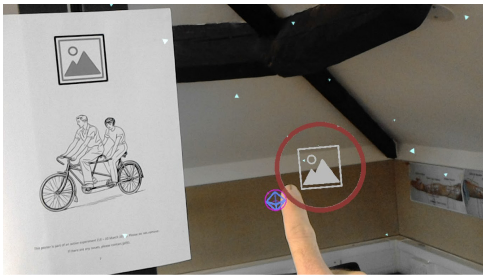

Fig. 27. A DESCRIPTION sub-task requiring the user to describe an image in four words or more.

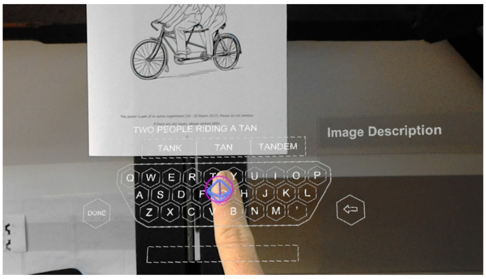

Fig. 28. The keyboard appears after selecting the task, allowing the user to enter the description.

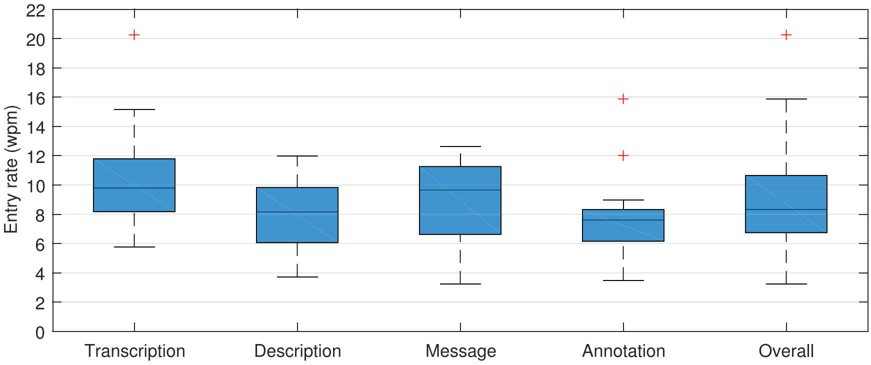

Fig. 29. Boxplots of entry rate (wpm) for each sub-task and over all sub-tasks in the validation study. Red crosses indicate outliers based on $q_3 + 1.5 \times (q3 - q1)$.

Figure 29 presents boxplots of the entry rates achieved by the participants in the four different sub-tasks as well as over all sub-tasks irrespective of task type. As might be expected, we see that different sub-tasks result in different entry rates. Intuitively, the highest mean entry rate achieved was in the TRANSCRIPTION task (10.32wpm). The overall mean entry rate achieved is 8.74wpm. This is clearly considerably lower than the maximum rates archived in Experiment 4 but tolerable for a casual text entry method. Furthermore, we know from the results of Experiment 4 that entry rates improve significantly with practice. The participants in this experiment only completed five practice phrases before commencing the spatial annotation task.

The four participants scored the system using the system usability scale: 72.5, 75.0, 77.5, and 67.5 (mean 73.1). Bangor et al. (2008) suggest that an SUS score above 70 indicates acceptable system usability. The SUS ratings provided by the participants are thus very promising and appear to indicate the viability of the VISAR keyboard as a text entry method for AR HMDs.

Following the experiment, participants were asked to comment on the aspects of the task that they enjoyed. Responses focused on the positive experience of being able to freely and interactively explore the space ('I liked the fact that it was a mixed reality environment so you had some elements that were virtual and some that were real', 'It was good to explore things') as well as the helpfulness of the directional cues provided ('It was good that it was directing you towards where it was going to ask something and then you are communicating through the system', 'The hint it gives me to find the tasks and also the sound system, it reminds me that there is a message coming in. That's the part that I really liked').

Participants were also asked to comment on any aspects of using the system that they found annoying. Two of the participants commented on comfort, one referring to the shoulder discomfort ('The main thing was the comfort of holding your hand up and especially the finger up as well') and the other referring to general discomfort with the headset ('I think the first problem is that it is not comfortable to wear the headset'). Other issues identified as being annoying were the small size of the headset display region and the lag in hand tracking. Also related to hand tracking, participants were warned in the introductory briefing that the HoloLens reported hand location was less robust when used in close proximity (<0.5 m) to walls or other fixed physical objects. Some of the participants encountered this issue during the task and commented on this in the interview. The keyboard location could, however, be adjusted by looking away from the current location and re-focusing on the new desired location. Participants were thus able to quickly remedy this issue when encountered.

In other miscellaneous comments, one participant observed that, 'I did find that you get drawn into the virtual elements and I had to stop and realise what was around me sometimes'. This observation highlights a broader challenge for AR interface design in terms of managing cognitive tunnelling. Another observation related to the lack of a physical response when interacting with the virtualised input surface of the keyboard, 'I feel like there is auditory and visual feedback but there is no tactile feedback on your finger and it is a bit disconcerting'. Examining the potential for minimal tactile feedback to enhance the experience of interacting with virtual objects in AR, and especially for text entry, offers an exciting avenue of exploration.

Finally, the four participants were asked whether the system worked sufficiently well to effectively complete the task. All answered in the affirmative, thus complementing the SUS score results obtained.

10 DISCUSSION

The five experiments presented in this study represent a structured attempt to apply immersive, and more specifically, AR-focused solution principles to the challenge of supporting efficient text entry for AR HMDs. In Experiment 1, we sought to test the hypothesis that the more natural direct-touch technique would support higher entry rates than a gaze-then-gesture baseline. The results revealed only a marginal improvement in overall entry rate associated with the VISAR keyboard although the time taken to select discrete keys was significantly faster. This finding encouraged us to explore why the more rapid key selection using the VISAR system failed to directly translate into higher entry rates.

An obvious deficiency identified was the lack of a precision fall-back mechanism allowing users to specify that certain letters are inputted with full certainty and need not be changed by the decoder. The lack of this feature resulted in significant time being wasted on correcting incorrect decoder returns. We therefore designed a seamless high-precision fall-back mechanism, which was evaluated in Experiment 2. This experiment indicated that the provision of a fall-back method in VISAR does not adversely affect entry rate and can help to reduce the error rate when employed effectively.

Experiment 3 is motivated by the design objective in AR HMDs to minimise keyboard occlusion of the real-world (*SP,3* in Section 3). We examine two conditions that sequentially removed visual features from the keyboard layout: first no letter labels, then no letter labels or key outlines. Almost all users in the experiment expressed surprise at being able to type effectively without key labels or outlines. The fastest participant in Experiments 2 and 3 achieved an entry rate of 15.26wpm at an error rate of 0.35% in the no keys or outlines condition shown in Figure 16. In general, we found that users could type quickly using no visual features, and although error rates rose significantly, the absolute error rates with no visual features were still far below the maximum tolerable threshold for character error rates (typically set at 5% CER). This finding demonstrates how VISAR can provide superior text entry support for AR HMDs.

The design of VISAR was further iterated upon to improve comfort in use and to include probabilistic, error-tolerant word predictions as suggested by *SP 4* in Section 3. Experiment 4 evaluated the refined VISAR keyboard against a similarly improved baseline condition, again based on the gaze-then-gesture selection paradigm. The results show that VISAR was capable of producing a mean entry rate across participants of 16.76wpm compared with 14.26wpm when completely naive users were exposed to each method and requested to type for between 1.5 and 2 hours. With the dominant period of learning removed, the mean entry rates were 17.75wpm and 14.84wpm for VISAR and the baseline respectively. This finding suggests a significant speed advantage for VISAR of approximately 20% relative to the baseline. The highest mean entry rate among all participants over a distinct experimental block of 20 phrases was 23.38wpm at a character error rate of 0.24%. More generally, the character error rate of the VISAR keyboard was elevated against the baseline condition with a mean of 0.63% across participants. While this is within generally tolerated levels, the result does highlight a speed-accuracy tradeoff. The additional speed provided by the VISAR keyboard comes with the cost of higher error rates.

Finally, the validation study demonstrated the suitability of VISAR for typical mobile AR text entry tasks. Participants were able to achieve mean entry rates in the range of 7 to 10wpm in a variety of transcription and composition tasks that were encountered while walking to explore a physical space. These entry rates were achieved after only a brief introduction on the use of the system and a seated training period involving only five practice phrases.

10.1 Implications for Design

Although we have demonstrated the effectiveness of the VISAR technique, we do not make the claim that entry rates, even at the maximum achieved by a participant in a distinct test block (23.38wpm), should be considered ‘fast’. Nevertheless, we believe that the obtained entry rates are sufficient for casual text entry in an AR HMD environment when a physical keyboard or other human–machine interface device is unavailable. We also anticipate that typical error rates are well within tolerable levels for most casual text entry tasks. The main contribution of this article is in highlighting several unique design requirements and solution principles relevant to AR HMDs.

The direct-touch interaction method exploited in VISAR is based on relatively coarse hand tracking. Despite this, users quickly adapted to the task of controlling the index cursor to touch keys, despite its positioning lag and inability to reflect hand articulation. Although the influence of high-precision hand-tracking in this task is worth exploring, the fact that sub-optimal tracking with a state-of-the-art AR HMD can still be exploited to deliver an immersive interaction experience for text entry should not be overlooked.

This study has also highlighted several design considerations specifically relevant to text entry in AR. First, participants found the interaction method tiring on the arm. This is not surprising given the duration of several of the experiments. We do, however, envisage that the likely use cases of fully hands-free environment-embedded AR text entry are sufficiently sporadic and short to make the VISAR approach worthwhile.

A second observation is that users appeared more prone to mishitting keys lower in the keyboard due to the trajectory followed by the hand during reaching. This has potential implications on the design of the keyboard layout and potential placement of other interaction elements.

The unfortunate inverse relationship between keyboard size and presentation proximity enforced by the constrained display window in current AR HMDs also introduces difficulties in accommodating a wide range of users. Although this statement requires proper investigation to be conclusive, it was casually observed that participants with shorter arms struggled more to find a comfortable position in which they could see a sufficient amount of the keyboard and easily reach the keys.

In terms of error-tolerant touch-based decoding, additional design explorations can possibly improve performance in an AR HMD environment, where interactions are laborious and/or imprecise. We observed deficiencies in the word-by-word decoding approach in instances where there are words closely located in the feature space, e.g., so and do, out and put, toy and you. This is particularly problematic in cases where such words occur early in sentences where the left context does not help narrow the search. A further challenge in deploying advanced decoding is the education of users. Many users expressed surprise at the effectiveness of the decoder but had to be encouraged to leverage its capability in order to increase their entry rate.

10.2 Limitations and Future Work

While the study reported here seeks to provide a thorough overview of the initial design and evaluation of a text entry method specifically design for AR HMDs, several limitations and opportunities for future work are acknowledged.

Experiments 1 and 4 both highlight a speed-accuracy tradeoff for the VISAR keyboard with the higher entry rate also producing higher error rates. Further work is required to determine whether suitable error correction and error prevention functionality might help mitigate this without degrading entry rates.

To test the applicability of the text entry method to fully featured text entry, it will be necessary to evaluate the effect of providing the full complement of punctuation as well as case modification. Furthermore, inclusion of insertion and editing functionality into the keyboard and interaction method would be necessary for a commercial product and also opens up many avenues for future work on investigating efficient correction interfaces for text entry in AR HMDs.

The system may lead to some discomfort during prolonged use due to the need to point with the hand without receiving any force feedback. To mitigate this, it is important to study the effect of keyboard pose and size on user comfort, possibly by employing metrics such as consumed endurance (Hincapié-Ramos et al. 2014). This finding also raises the question as to what text entry uses cases are likely to emerge for AR HMDs and what factors might encourage the user to transition between different input modes.

The result that users can type effectively without any key outlines or labels suggests the system might also support text entry while walking. However, this would need to be carefully evaluated, updating findings from similar investigations for mobile phones (Oulasvirta et al. 2005).

Finally, the system architecture is flexible enough to support decoding of alternative text entry modalities, in particular gesture keyboard decoding (Kristensson and Zhai 2004). Future work could investigate if this would result in any additional performance gain.

11 CONCLUSIONS

Many anticipated applications of AR require the ability to enter text. Text entry methods for AR should exploit the unique advantages of immersive interfaces rather than being cobbled together from paradigms borrowed from two-dimensional interfaces. In this article, we explore the design of an AR text entry method based on error-tolerant mid-air touch interaction with a virtual keyboard. We investigate its effectiveness on the Microsoft HoloLens in a series of five controlled user studies.

The experimental results show that users can select keys more quickly using the direct-touch approach than with the gaze-then-gesture approach. This delivers significantly faster entry rates when combined with probabilistic word predictions. A particularly striking result is that a subgroup of users can maintain and exceed entry rates when all key labels and outlines are removed from the keyboard so that only the keyboard region outline remains.

The key contributions of this article are three-part. First, we present six solution principles informed by the literature and prior interface design experience that inform the design of productive text entry methods for AR HMDs. Second, we describe a novel keyboard system specifically adapted to AR HMDs based on an error-tolerant touch-driven interaction paradigm and incorporating an inferred-to-literal fall-back method, as well as configurable occlusion settings to improve user visibility of the physical environment. Third, we provide empirical results from a comparison with a gaze-then-gesture baseline entry method, and an investigation of the influence of various design decisions. This establishes a useful point of reference for future studies seeking to explore productive text entry in AR. In summary, we show that VISAR can support productive text entry on a head-mounted AR display. We hope that the solution principles upon which the system is based inspire other novel and efficient entry methods for AR.

REFERENCES

Aaron Bangor, Philip T. Kortum, and James T. Miller. 2008. An empirical evaluation of the system usability scale. *International Journal of Human-Computer Interaction* 24, 6 (2008), 574–594. DOI: https://doi.org/10.1080/10447310802205776

Xiaojun Bi, Barton A. Smith, and Shumin Zhai. 2012. Multilingual touchscreen keyboard design and optimization. *Human-Computer Interaction* 27, 4 (2012), 352–382. DOI: https://doi.org/10.1080/07370024.2012.678241

Doug A. Bowman, Christopher J. Rhoton, 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.

John Brooke. 1996. SUS: A 'quick and dirty' usability scale. In *Usability Evaluation In Industry*. CRC Press, 189–194.

K. M. Chung, Jennifer T. T. Ji, and Richard Hau Yue So. 2011. Manual control with time delays in an immersive virtual environment. In *Proceedings of the International Conference on Ergonomics & Human Factors: Contemporary Ergonomics and Human Factors 2011*. CRC Press, 211–218.

Edward Clarkson, James Clawson, Kent Lyons, and Thad Starner. 2005. An empirical study of typing rates on mini-QWERTY keyboards. In *Proceedings of CHI '05 Extended Abstracts on Human Factors in Computing Systems (CHI EA'05)*. ACM, New York, NY, 1288–1291. DOI: https://doi.org/10.1145/1056808.1056898

James Clawson, Kent Lyons, Alex Rudnick, Robert A. Iannucci, Jr., and Thad Starner. 2008. Automatic whiteout++: Correcting mini-QWERTY typing errors using keypress timing. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'08)*. ACM, New York, NYs, 573–582. DOI: https://doi.org/10.1145/1357054.1357147

Paul A. David. 1985. Clio and the economics of QWERTY. *American Economic Review* 75, 2 (1985), 332–337.

Joshua Goodman, Gina Venolia, Keith Steury, and Chauncey Parker. 2002. Language modeling for soft keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces (IUI'02)*. ACM, New York, NY, 194–195. DOI: https://doi.org/10.1145/502716.502753

Tovi Grossman, Xiang Anthony Chen, and George Fitzmaurice. 2015. Typing on glasses: Adapting text entry to smart eyewear. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'15)*. ACM, New York, NY, 144–152. DOI: https://doi.org/10.1145/2785830.2785867

Reid Harmon, Walter Patterson, William Ribarsky, and Jay Bolter. 1996. The virtual annotation system. In *Proceedings of the IEEE Virtual Reality Annual International Symposium, 1996*. 239–245.

Juan David Hincapié-Ramos, Xiang Guo, Paymahn Moghadasian, 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*. ACM, 1063–1072.

Geoffrey Hinton, Li Deng, Dong Yu, George E. Dahl, Abdel-rahman Mohamed, Navdeep Jaitly, Andrew Senior, Vincent Vanhoucke, Patrick Nguyen, Tara N. Sainath, and Brian Kingsbury. 2012. Deep neural networks for acoustic modeling in speech recognition: The shared views of four research groups. *IEEE Signal Processing Magazine* 29, 6 (2012), 82–97. DOI: http://dx.doi.org/10.1109/MSP.2012.2205597

Errol R. Hoffmann. 1992. Fitts' law with transmission delay. *Ergonomics* 35, 1 (1992), 37–48. DOI: https://doi.org/10.1080/00140139208967796

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 (ICMI'13)*. ACM, New York, NY, 213–220. DOI: https://doi.org/10.1145/2522848.2522861

Seoktae Kim, Minjung Sohn, Jinhee Pak, and Woohun Lee. 2006. One-key keyboard: A very small QWERTY keyboard supporting text entry for wearable computing. In *Proceedings of the 18th Australia Conference on Computer-Human Interaction: Design: Activities, Artefacts and Environments (OZCHI'06)*. ACM, New York, NY, 305–308. DOI: https://doi.org/10.1145/1228175.1228229

Per Ola Kristensson. 2015. Next-generation text entry. *Computer* 48, 7 (2015), 84–87.

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.

Per-Ola Kristensson and Shumin Zhai. 2005. Relaxing stylus typing precision by geometric pattern matching. In *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI'05)*. ACM, New York, NY, 151–158. DOI: https://doi.org/10.1145/1040830.1040867

Falko Kuester, Michelle Chen, Mark E. Phair, and Carsten Mehring. 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, 86–95. DOI: https://doi.org/10.1145/1101616.1101635

Minkyung Lee and Woontack Woo. 2003. ARKB: 3D vision-based augmented reality keyboard. In *Proceedings of the 13th International Conference on Artificial Reality and Telexistence*.

Stan J. Liebowitz and Stephen E. Margolis. 1990. The fable of the keys. *Journal of Law & Economics* 33, 1 (1990), 1–25.

Anders Markussen, Mikkel R. Jakobsen, and Kasper Hornbæk. 2013. *Selection-Based Mid-Air Text Entry on Large Displays*. Springer, Berlin, 401–418. DOI: https://doi.org/10.1007/978-3-642-40483-2_28

Anders Markussen, Mikkel Rønne Jakobsen, and Kasper Hornbæk. 2014. Vulture: A mid-air word-gesture keyboard. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 1073–1082. DOI: https://doi.org/10.1145/2556288.2556964

Tao Ni, Doug Bowman, and Chris North. 2011. AirStroke: Bringing unistroke text entry to freehand gesture interfaces. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'11)*. ACM, New York, NY, 2473–2476. DOI: https://doi.org/10.1145/1978942.1979303

Antti Oulasvirta, Sakari Tamminen, Virpi Roto, and Jaana Kuorelahti. 2005. Interaction in 4-second bursts: The fragmented nature of attentional resources in mobile HCI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 919–928.

Sebastian Pick, Andrew S. Puika, and Torsten W. Kuhlen. 2016. SWIFTER: Design and evaluation of a speech-based text input metaphor for immersive virtual environments. In *Proceedings of the IEEE Symposium on 3D User Interfaces (3DUI'16)*. IEEE, 109–112.

Ivan Poupyrev, Numada Tomokazu, and Suzanne Weghorst. 1998. Virtual notepad: Handwriting in immersive VR. In *Proceedings of the IEEE 1998 Virtual Reality Annual International Symposium*. 126–132.

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*. ACM, 9–18.

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.

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, 3643–3652. DOI: https://doi.org/10.1145/2702123.2702136

Andreas Stolcke. 2002. SRILM – An extensible language modeling toolkit. In *Proceedings of International Conference on Spoken Language Processing*. 901–904.

Jouke C. Verlinden, Jay David Bolter, and Charles van der Mast. 1993. *Virtual Annotation: Verbal Communication in Virtual Reality*. Technical Report GIT-GVU-93-40. Georgia Institute of Technology.

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*. ACM, 295–298.

Keith Vertanen, Haythem Memmi, Justin Emge, Shyam Reyal, and Per Ola Kristensson. 2015. VelociTap: Investigating fast mobile text entry using sentence-based decoding of touchscreen keyboard input. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*. ACM, 659–668.

Cheng-Yao Wang, Wei-Chen Chu, Po-Tsung Chiu, Min-Chieh Hsiu, Yih-Harn Chiang, and Mike Y. Chen. 2015. PalmType: Using palms as keyboards for smart glasses. In *Proceedings of the 17th International Conference on Human-Computer Interaction with Mobile Devices and Services (MobileHCI'15)*. ACM, New York, NY, 153–160. DOI: https://doi.org/10.1145/2785830.2785886

Daryl Weir, Henning Pohl, Simon Rogers, Keith Vertanen, and Per Ola Kristensson. 2014. Uncertain text entry on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (CHI'14)*. ACM, New York, NY, 2307–2316. DOI: https://doi.org/10.1145/2556288.2557412

Xin Yi, Chun Yu, Mingrui Zhang, Sida Gao, Ke Sun, and Yuanchun Shi. 2015. ATK: Enabling ten-finger freehand typing in air based on 3D hand tracking data. In *Proceedings of the 28th Annual ACM Symposium on User Interface Software and Technology (UIST'15)*. ACM, New York, NY, 539–548. DOI: https://doi.org/10.1145/2807442.2807504

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, 4479–4488. DOI: https://doi.org/10.1145/3025453.3025964

Chun Yu, Ke Sun, Mingyuan Zhong, Xincheng Li, Peijun Zhao, and Yuanchun Shi. 2016. One-dimensional handwriting: Inputting letters and words on smart glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM, 71–82.

Shumin Zhai and Per Ola Kristensson. 2012. The word-gesture keyboard: Reimagining keyboard interaction. *Communications of the ACM* 55, 9 (Sept. 2012), 91–101. DOI: https://doi.org/10.1145/2330667.2330689

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.

Received September 2017; revised April 2018; accepted June 2018