HIBEY: Hide the Keyboard in Augmented Reality

Lik Hang LEE[*], Kit Yung LAM[*], Yui Pan YAU[*], Tristan BRAUD[*], and Pan HUI[*†]

[*]The Hong Kong University of Science and Technology - Hong Kong
[†]University of Helsinki - Finland
Email: lhleeac@connect.ust.hk, kylambd@connect.ust.hk, arthuryaucs@gmail.com, braudt@ust.hk, panhui@ust.hk

***Abstract*—Text input is a very challenging task in Augmented Reality (AR). On non-touch AR headsets, virtual keyboards are counter-intuitive and character keys are hard to locate inside the constrained screen real estate. In this paper, we present the design, implementation and evaluation of HIBEY, a text input system for smartglasses. HIBEY provides a fast, reliable, affordable, and easy-to-use text entry solution through vision-based freehand interactions. Supported by a probabilistic spatial model and a language model, a three-level holographic environment enables users to apply fast and continuous hand gesture to pick characters and predictive words in a keyboard-less interface. Through the pilot study and a thorough evaluations lasting 8 days, we show that HIBEY leads to a mean text entry rate of 9.95 word per minute (WPM) with 96.06% accuracy, which is comparable to other state-of-the-art approaches. After 8 days, participants can achieve an average of 13.19 WPM. In addition, HIBEY only occupies 13.14% of the screen real estate at the edge region, which is 62.80% smaller than the default keyboard layout on Microsoft Hololens.**

I. Introduction

Smartglasses overlay virtual content directly on top of the user's physical surroundings [33]. The virtual content can take various forms including windows, menus, and icons [18]. The default interaction approaches, such as controlling a cursor with a mini-touchpad wired to the smartglasses or vision-based hand gesture recognition, allow users to interact with the virtual content. The windows, menus and icons are large in size and thus easy to locate. However, these approaches are insufficient for text input, which involves small-sized content selection. On smartglasses, selecting character keys on virtual keyboards becomes error-prone and inefficient [14] as users may experience difficulties to locate small character keys for a highly repetitive task. Alternatively, speech recognition is a major input method for smartglasses but has limitations such as being subject to environmental noise and presents issues with social acceptance [20][25]. Apart from the above approaches, researchers have proposed other text input methods. These methods include adding external proprietary sensors or re-configuring keyboard arrangement. However, adding proprietary sensors leads to unfavorable setup times and additional hardware costs. Instead, vision-based approaches do not require external proprietary sensors but employ virtual keyboards that occupy a large surface on the screen space. For instance, the virtual QWERTY keyboard in Microsoft Hololens uses the majority of the screen's center area, which leads to obstructed screen real estate at the expense of other Augmented Reality (AR) applications.

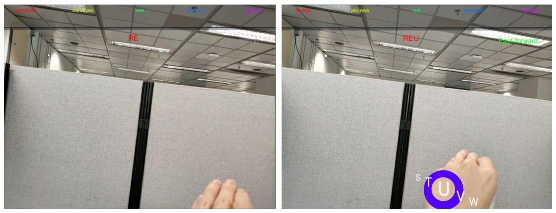

Fig. 1. Keyboard-less interface of HIBEY in the holographic environment, where spectator mode is switching on to aid revealing character positions; Picking character through hand movement from preparation zone to fast-forward zone

Considering the drawbacks of the aforementioned approaches, we propose HIBEY, a convenient and unobtrusive solution. HIBEY enables smartglasses users to input characters and words in AR without adding any proprietary sensors. By providing a keyboard-less text input experience, HIBEY reserves the majority of the screen's real estate for applications in the holographic environment of Microsoft Hololens. As users rarely invest time in learning new keyboard layouts [6], HIBEY leverages the advantages of arranging the characters in alphabetical order such as performance improvement [42] and better usability [10] to novice users [21]. HIBEY applies continuous hand gesture interactions in a holographic environment using a keyboard-less approach. Our solution only requires the user to perform a single discrete hand gesture, i.e. mid-air tap, to initialize the character selection. The user then holds a pointing gesture throughout the text input. During the character selection, the user targets characters arranged along a single line through horizontal hand movements. To terminate the process, the user just releases the hand gesture. Compared with intensive mouse cursor movement and mid-air taps on a virtual keyboard, our solution can preserve the unique advantages of gestural input, such as no additional sensor and natural interaction, while maintaining the screen's real estate.

We implement HIBEY on Microsoft Hololens, which is an AR headset with a wide field of view supporting holographic experiences. HIBEY consists of three key procedures for typing words by mid-air freehand input: 1) the Pointing Gesture enables users to choose and grab the target character in mid-air; 2) the planar boundary between the preparation zone and the fast-forward zone acts as a virtual touchscreen in mid-air, where users can type on the keyboard-less environment (Figure 1); 3) The coordinates on this planar boundary are then recorded by the system and passed to the statistical decoder. This module, which lies on both a language model and a spatial model, computes the most probable words, with the assistance of a word disambiguation algorithm.

We recruited 18 participants to evaluate the performance of HIBEY on Microsoft Hololens and 7,200 word phases were tested throughout the 8-day session. In our experiments, the participants were able to achieve an average text entry speed of 9.95 word per minutes (WPM) with an average accuracy of 96.06% across all the trials. During the final trial, participants achieved an average speed of 13.19 WPM, which is comparable to other state-of-the-art methods. In addition, the majority of participants thinks our proposed system is preferable to mid-air tap on virtual keyboard for text input. We also show comparable results with other state-of-the-art approaches such as touch-on-device input and touch-on-skin input. To the best of our knowledge, this paper presents the first holographic text entry system for smartglasses, which reduces the burdens of selecting character keys on virtual keyboard. HIBEY introduces a novel design of keyboard-less interface, yielding the following contributions and improvements over other existing text entry systems:

- The paper demonstrates the potentials of mid-air hand gestural inputs for text input in AR without sacrificing a large proportion of screen area.
- HIBEY applies a spatial model and a language model to improve text input over discrete mid-air tap on virtual keyboard or sign languages.
- HIBEY enables smartglasses to achieve reliable and fast text input whichever the external environment (e.g. noisy environment).

The rest of this paper is organized as follows. We first summarize the major related studies in Section II. We then describe HIBEY's system design in Section III. We validate our intuitions in a pilot study described and then analyze the imprecision and word disambiguation process within the holographic environment in Section IV. Finally, we describe our implementation and evaluate the proposed solution in Section IV-B.

II. Related Work

In this section, we review the main text entry techniques on mobile devices with small interaction areas, as well as text entry techniques for smartglasses. These techniques can be categorized as follow: on-device interaction, body-centre interaction, and freehand interactions.

A. *Text input on size constrained devices*

Typing on the limited space of mobile devices is an age-old problem since the launch of feature phones. Letterwise [27] proposes dictionary-based disambiguation to support full character input with only 12 keys on feature phones. The condensed two-line mini-physical keyboard [12] has several mode switchers to enable 10 keys to map with various symbols and characters. A prior study [8] compares the performance of blind typing with Twilddler and a physical mini-keyboard. The study shows that blind-typing with Twilddler is faster and more accurate than the mini-keyboard.

The rise of smartphones has accelerated research related to text input on touchscreen soft keyboards. We only focus on the works attempting to reduce the soft keyboard space and increase the screen free space. 1-line keyboard [24] restructures the full QWERTY layout into an 8-key ambiguous keyboard on tablets, which reduces the keyboard size to 32% of the typical QWERTY soft keyboard size. Commercial keyboards such as Minum [2] and ASTETNIOP [1] also use one-line keyboards and assume the users can fix their ten fingers on the dedicated position for rapid typing. Another commercial keyboard – FLEKSY [3] – as well as an experimental prototype [4] enable users to type on a translucent keyboard.

An alternative approach to maximizing the touchscreen free space is to leverage the rear area of mobile devices [39][34][19]. Addendum sensors are installed on the back of mobile device and leave the entire screen for content display. Users put their hands on the touch sensors at the rear of smartphones for text input and achieve around 15 WPM.

HIBEY is similar to the above techniques using blind typing. However, the character keys layout in HIBEY is hidden. We leverage the free space in the holographic environment to input text under the supports of a spatial model as well as the disambiguation algorithm.

B. *On-device and Body-centric interaction*

On-device interaction refers to the interactions on a sensible surface of various devices such as the spectacle frame of smartglasses and peripheral sensors on external devices. Swipe-based gesture are developed for text entry using the spectacle frame on Google Glass [13][41]. In Yu et al's work [41], each character is represented by a set of bidirectional unistrokes. For instance, the character 'a' is composed of three swipes – 'down-up-down' – that mimic handwriting strokes. In SwipeZone [13], the touchable spectacle frame is divided into three zones. Users can choose one of the zones and subsequently target the character inside the zone. Other works focus on the optimal use of an external controller wired with smartglasses to achieve off-hand text entry, which allows users to operate a cursor and select keys on a virtual on-screen keyboard such as Dasher input system [29]. A ring wearable [32] enables two-stage character selection on a virtual QWERTY keyboard in which characters are grouped into a sequence of 3 consecutive keys. In general, on-device interaction approaches are precise and responsive. However, the major drawbacks are the existence of the device itself and the preparation time for putting on the device [9].

Body-centre interaction refers to interfaces located on the user's body. Wang et al. [38] propose an imaginary palm keyboard for text entry. Supported by infrared sensors located on the wrist, the user can touch the target key shown on the virtual keyboard through the optical display, which is faster than a touchpad wired with smartglasses.

Speech recognition is becoming the major text input method on Google Glass and Microsoft Hololens. However, it might be not appropriate in shared or noisy environments, for example, causing disturbance and obtrusion, disadvantages to mute individuals, or being accidentally activated by environmental noise [25], and is less preferable than body gestures or handheld devices input approaches [20]. In contrast, our work exhibits an alternative approach with no addendum sensors and none of the aforementioned social acceptance issues.

C. Freehand interactions

Freehand interactions have exhibited their outstanding capabilities in 3D interfaces. Most of the users prefer interacting with 3D objects through hand gesture over the touch input approaches. Indeed, performing hand gesture in front of facial area is natural and straightforward [36]. The current works of vision-based freehand interactions are primarily interested in the manipulation of 3D objects [23] and physical environments in Augmented Reality [15]. Vision-based sign language using iconic-static gestures suffers from low entry rates, due to long dwelling times of recognizing every single hand sign, and results in unproductive input speed [17]. Another prior work on mid-air text input on a virtual keyboard achieves 23.0 – 29.2 WPM [40] but the majority of screen space is occupied by the virtual keyboard and the LEAP Motion sensor is not available on most of the smartglasses [22].

Contrary to the above studies, our work addresses the text input on smartglasses under the constraint of limited screen size. The key challenge of text input on smartglasses is that the on-screen keyboard on the small display is inconvenient and space-consuming, thus violates the intention of interacting with the physical environment. Also, it is challenging to design a minimized interface that addresses the usability issue, as it is subject to imprecise hand gesture input and the uncertainty of character selection without visual clues. To the best of our knowledge, we are the first work to get rid of the space-consuming on-screen keyboard and enable the users to type avoiding ambient occlusion in the holographic environment.

III. System Design

In this section, we explain the system design and how the users accomplish the character input in the holographic environment. The statistical decoders, word disambiguation algorithm supporting the interaction will be discussed in Section IV.

A. Interaction Overview

HIBEY relies on three connected zones in the 3D space as shown in Figure 2. The user moves his hand to choose the invisible characters configured in 1-line formation, and to confirm or recall the characters through traversing the zones.

a) Preparation zone: This zone serves as a preparation area in which the user selects the characters among the horizontal line of available alphabet. The user selects characters by moving his hand forward to the fast-forward zone.

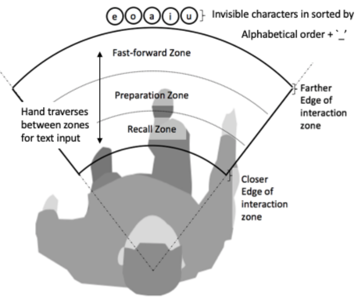

Fig. 2. Three connected zones in interaction area

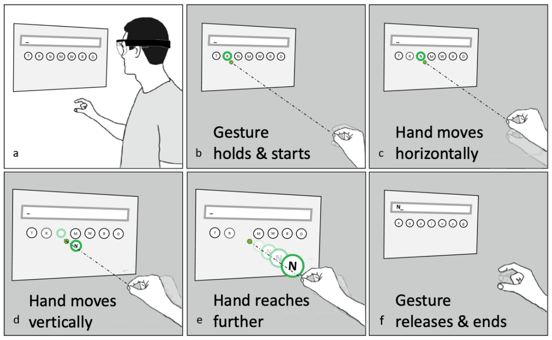

Fig. 3. Pictorial description of picking an invisible character in HIBEY

b) Fast-forward zone: This zone is designed for facilitating the character input. The user's hand moves horizontally to select a character and afterwards moves forward from the preparation zone to the fast-forward zone to confirm the character selection. The selected character will move from the farther edge to the closer edge of the interaction area accordingly. The character's movement speed is directly proportional to the relative depth position of the user's hand in this zone. As such, the user gets a control of his typing speed which allows him to focus on other tasks within the holographic environment.

c) Recall zone: This zone provides a backspace function for character input. Contrary to the Fast-forward zone, the user moves the character from the closer edge to the farther edge of the interaction area to recall an input when the user's hand is located within this zone.

Figure 3 shows the detailed procedures of character selection in the holographic environment. The user places his hand in front of the embedded camera on smartglasses (Figure 3.a) and makes a pointing gesture. The text input will start once a mid-air tap is performed (Figure 3.b). The user's hand horizontal movements allow choosing the target character (Figure 3.c). When the user's hand reaches the boundary between the preparation and fast-forward zone, the target character appears translucently in text box as visual feedback (Figure 3.d). In addition, moving the user's hand in the Fast-forward zone can adjust the character influx speed in case the user feels difficult to follow the current pace. For example, the hand further moves forward to increase the influx speed

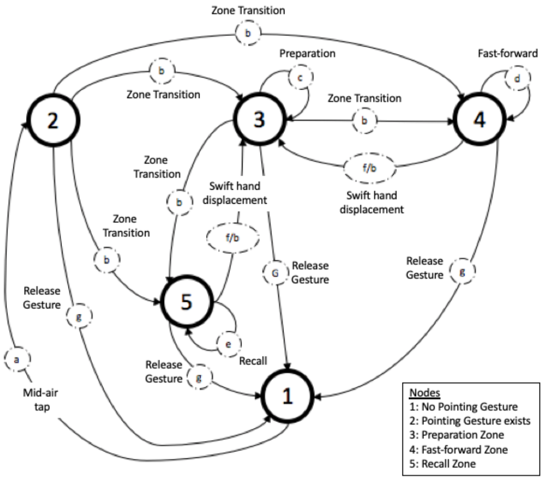

Fig. 4. Transition of interaction operations

(Figure 3.e). Finally, the selected character is confirmed in the text box (Figure 3.f). The user can proceed to the next character through holding the gesture, or release the gesture to end the process (Figure 3.g).

B. Character keys

Character keys are initially located at the farther edge of the holographic environment. We arrange the 27 characters including the 26 characters from the Roman alphabet and the white space '_' (positioned after the alphabet) in a horizontal line formation, in alphabetical order. Users know the character order instinctively, which leads to performance improvement [42] and better usability [10] to novice users [21]. Prior studies of 1-line layouts [43] show that the alphabetical order outperforms the QWERTY and ENBUD layouts. ENBUD [37] has optimized character arrangement but is impractical for finding the characters when the keys are hidden.

Regarding the movement of the characters, α is the initial flying time of character and its change is subject to the length I of prefix (typed substring) in a word, with a discounted factor β. This means the basic velocity will speed up when a substring with more characters results in a smaller number of next possible characters. Thus, the basic velocity of the character is calculated as $V = \frac{D_z}{\alpha-(I\beta)}$. At time frame J, the basic velocity is further accelerated by the relative position of the user's hand H_j in the Fast-forward zone or Recall zone. The farther the hand from the Preparation zone, the faster the movement of the characters. The final velocity V_j is computed as $V_j = V + \frac{V}{\gamma}\int\limits_0^\gamma H_j$, where γ is the number of sub-zones in Fast-forward zone or Recall zone.

C. Continuous Pointing Gesture

In contrast to physical input devices (e.g. mouse and stylus) which feature a high level of precision, pointing gestures in mid-air are relatively coarse and unstable [9]. Therefore, direct positioning operations on small and dense items, such as character input on virtual keyboards with hand gestures are difficult. Instead, our system enables the user to start the text input by performing a mid-air tap. The user then holds a pointing gesture that serves as a pointing token hovering between the connected zones. We select the pointing gesture for its similarity to selecting objects in the physical world that makes it intuitive for users [31]. Figure 4 illustrates the five transition states of continuous pointing gestures: 1) *No pointing gesture is detected*, 2) *Pointing gesture is detected*, 3) *Pointing gesture in Preparation zone*, 4) *Pointing gesture in Fast-forward zone*, and 5) *Pointing gesture in Recall zone*. The transitions (a - g) between states are described as follows. Hold (a): A mid-air tap gesture is maintained, which is interpreted as the initialization of text input. Enter (b): The pointing gesture moves to a new zone. Select (c): The pointing gesture in the Preparation Zone chooses the neighboring characters. Facilitate (d): The pointing gesture in Fast-forward Zone moves the character forward. The movement speed increases or decreases by respectively shifting the pointing gesture forward or backwards. Recall (e): The pointing gesture in Recall Zone makes a backspace function to the selected characters. Flip (f): The user can do a large horizontal displacement of pointing gesture to drop the chosen character key. Release (g): The user releases the pointing gesture or the camera cannot find the user's hand.

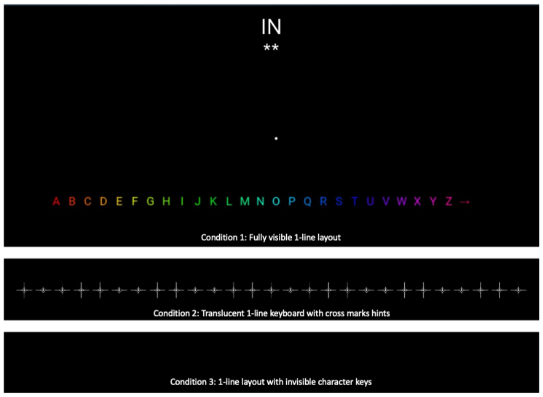
IN
**
A B C D E F G H I J K L M N O P Q R S T U V W X Y Z →
Condition 1: Fully visible 1-line layout
Condition 2: Translucent 1-line keyboard with cross marks hints
Condition 3: 1-line layout with invisible character keys

Fig. 5. The three experimental interfaces

IV. Uncertainty on Keyboard-less Environment

In this section, we first conduct a pilot study to understand the user behavior of keyboard-less typing in the holographic environment. This experiment aims at studying the performance of text entry in three visual conditions and validate the feasibility of keyboard-less text entry in the holographic environment. We collect and use the data of position displacements to manage the imprecision through the proposed probabilistic models.

A. Design of Pilot Test

We evaluate our text entry holographic environment in three visual feedback settings presented in Figure 5: (1) fully visible

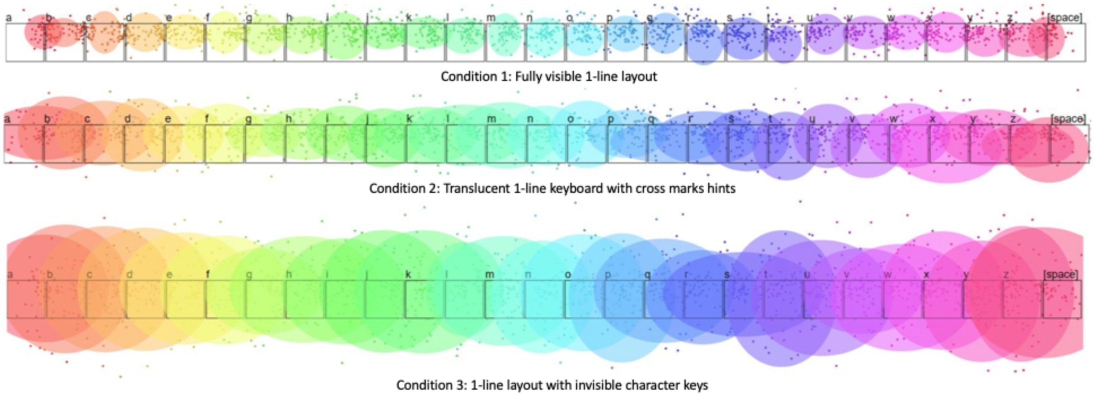

Fig. 6. The distribution of coordinates with the three experimental interfaces

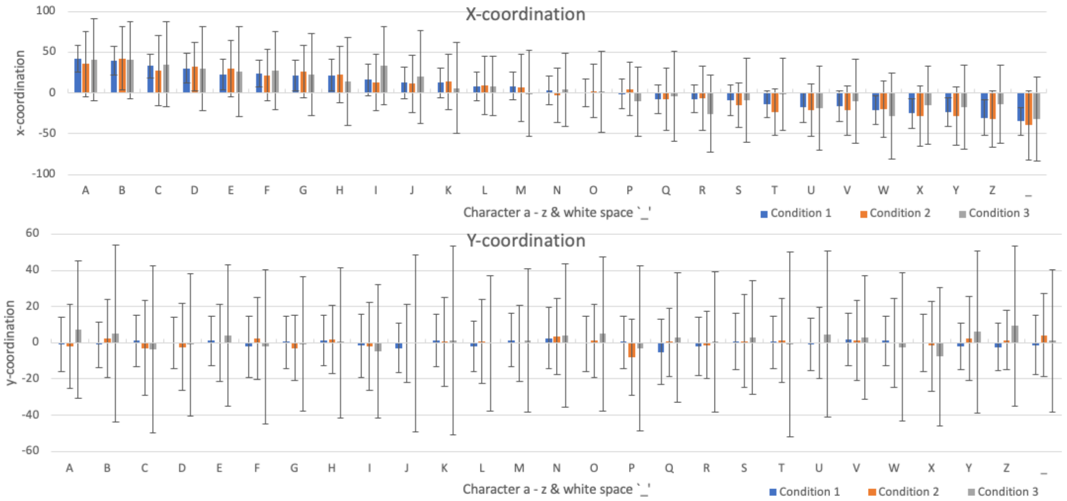

Fig. 7. The distribution of x- and y-coordinates with the three experimental interfaces (upper: x-coordinates & lower: y-coordinates)

1-line layout (top), (2) translucent 1-line keyboard with cross marks hints (middle), and (3) 1-line layout with invisible character keys (bottom). The 1-line layout is configured to enable the users to select the characters in fixed position for easy memorization. We recruit 15 participants from the local universities. The participants had no prior experience in mid-air text input. 4 out of 15 participants had tried Microsoft Kinect application. None of them is a native English speaker but all are familiar with the alphabetical order. The experiment is conducted on Microsoft Hololens. We instruct participants to input word phrases as accurately as possible, i.e. locate the character keys, without correction. The output of text entry is represented by asterisks to avoid bias towards our keyboard design and force the participants to pay attention to the next character entry. The three layouts are tested in alternative order. For each layout, the participants complete 5 sessions featuring 15 phrases from MacKenzie & Soukoreff phrase set [28] for a total of 2700 phrases (3 layouts * 15 participants * 5 blocks * 15 phrases). In order to alleviate the imbalance on the least frequent characters such as q, x , and z, we handpick and balance the word phrases. For each character input, we record the x and y coordinates located across the boundary between the preparation zone and the fast-forward zone.

B. *Results and Implication of the Pilot Test*

Figure 6 shows the distribution of coordinates on the boundary between the preparation zone and the fast-forward zone. The ellipses enclose the recorded coordinates within 95% confidence. The geometric centers of character keys are shown within the squares. The three distributions represent the 3 layout according to Figure 5: (1) fully visible 1-line layout (top), (2) translucent 1-line keyboard with hints of cross marks (middle), and (3) 1-line layout with invisible character keys (bottom). We define the offset (in pixel) as the measured coordinates minus the geometric center of the character key (the center of the square).

Regarding the horizontal offset, ANOVA demonstrates a significant effect of the visual feedback on the horizontal offset ($F_{2,69}$ = 209.448, p <0.0001) and pairwise comparison between each layout shows a significant difference (p <0.0001). The mean offsets for visual conditions (1), (2) and (3) are respectively 22.97 (std. = 16.31), 32.12 (std. = 26.90) and 40.41 (std. = 36.51). The offset for layout (3) (invisible character keys) is 75.93% larger than for the fully visible layout, while the standard deviation of the layout (3) is 138.12% greater than the fully visible layout. According to Figure 6, we observe that layouts (1), (2) and (3) respectively display an approximate offset length of 0–1, 1–2 and 2–3. For all three layouts, the common tendency is that the measured centers of the leftmost 9 characters and rightmost 9 characters are shifted to the center of the screen while the middle 9 characters show random centers of measured horizontal coordinates.

Regarding the vertical offset, ANOVA demonstrates a significant effect of the visual feedback on the vertical offset ($F_{2,69}$ = 446.891, p <0.0001). Pairwise comparison between each layout shows a significant difference (p <0.0001). The mean offsets for visual conditions (1), (2) and (3) are 11.28 (std. = 9.77), 15.54 (std. = 15.67) and 29.16 (std. = 29.76). The vertical offset between condition (1) and (2) shows only 37.74% and 60.31% difference in the values of mean and standard deviation. In contrast, the offset of the layout with invisible character keys is 158.49% larger than the fully visible layout, while the standard deviation of the layout with invisible character keys is 2 times larger than the fully visible layout. We observe that users in condition (3) have a greater vertical movement, which aligns with our findings shown in Figure 6.

In the study, we investigate the possibility of text input under the keyboard-less condition. Under condition (3), the overlapping of x-coordinates across keys is generally no bigger than 2 character keys. The primitive approach considering an offset of fixed size 2–3 characters is feasible but deteriorates the performance of word disambiguation [43]. Instead, we apply a probabilistic approach to handle the uncertainty issue due to imprecise character selection. Figure 7 summarizes the offset of coordinates, where μ_x and μ_y are the mean offset values of x and y coordinates, σ_x and σ_y are the standard deviation of x and y coordinates, and ρ is the correlation between x and y coordinates.

C. Probabilistic Model for Handling Imprecision

The imprecision in hand gestural text input, especially in holographic environments, can be handled by statistical decoding. Note that swipe-based trajectory hovering over the needed character keys on the virtual keyboard is not recommended because hand gesture detection is coarse and not as accurate as the touchscreen on smartphones [26]. Usual statistical decoders for touchscreen interfaces are supported by both the language model and spatial model [7][11]. In order to simplify the computational workloads in the holographic environment, we design a transformed coordinate system from 3D into 2D. The boundary between the preparation zone and the fast-forward zone serves as a 'touchscreen' in mid-air, and the hand gestures traversing this 2D plane are computed as 'touch points' in the statistical decoder. The traversing locations on the imaginary 2D plane are further interpreted by the probabilistic distribution as shown in Figure 7. As shown in Figure 7, the ends of bars and the error bar show the mean coordination and corresponding standard deviation, respectively. Both x and y bar plots show a general trend in which the lesser the visual clues, the higher the imprecision value obtained, as indicated by the standard deviation values. The x-coordinations for the characters have demonstrated consistent and close positions among three conditions. In contrast, the y-coordination has shown random vertical direction in most of the characters. The major reason is that the characters arranged in horizontal position and therefore the users carefully pick the x-coordinates, while the vertical positions do not have a significant impact on the character selection. Thus, the users choose the characters at their most convenient positions. Bayes' theorem computes the probability of a word and recommends the most probable words in the word suggestion list. Given a set of 2D coordinates in mid-air $C = \{c_1, c_2, c_3, ..., c_n\}$, the decoder searches for the optimal word W_{Opt} inside the lexicon X satisfying

$$W_{Opt} = arg \max_{\forall W \in X} P(W|C)$$

According to the Bayes' rule, we have

$$W_{Opt} = arg \max_{\forall W \in X} P(C|W)P(W)$$

where $P(W)$ and $P(C|W)$ are respectively computed by the language model [11] and the spatial model. Given that W consists of n characters: $L = \{l_1, l_2, l_3, ..., l_n\}$, the spatial model computes the $P(C|W)$ as follows.

$$P(C|W) = \prod_{i=1}^{n} P(c_i|l_i)$$

Prior research [5] shows that the character selection on 2D interfaces follows the Bivariate Gaussian distribution. The x and y coordinates of c_i are x_i and y_i and hence,

$$P(c_i|l_i) = \frac{1}{2\pi\sigma_{i_x}\sigma_{i_y}\sqrt{1-\rho_i^2}} exp[\frac{z}{2(1-\rho_i^2)}]$$

and

$$z \equiv \frac{(x_i - \sigma_{i_x})^2}{\sigma_{i_x}} - \frac{2\rho_i(x_i - \mu_{i_x})(y_i - \mu_{i_y})}{\sigma_{i_x}\sigma_{i_y}} + \frac{(y_i - \sigma_{i_y})^2}{\sigma_{i_y}}$$

where (μ_{i_x}, μ_{i_y}) is the geometrical center of the character key l_i; σ_{i_x} and σ_{i_y} are the standard deviations of x and y coordinates for character key l_i; ρ_i is the correlation value of the x and y coordinates. The collected data from the pilot study (the trends of $\mu_{i_x}, \mu_{i_y}, \sigma_{i_x}, \sigma_{i_y}$), as summarized in Figure 6 and Figure 7, are applied to the above equations to determine the most probable word W_{Opt} in the keyboard-less text input environment.

D. Word Disambiguation

Word disambiguation happens when the recorded x_i and y_i coordinates point to an overlapping area between two or more characters. In this scenario, word disambiguation is necessary for efficient text input. We implement our algorithm for word

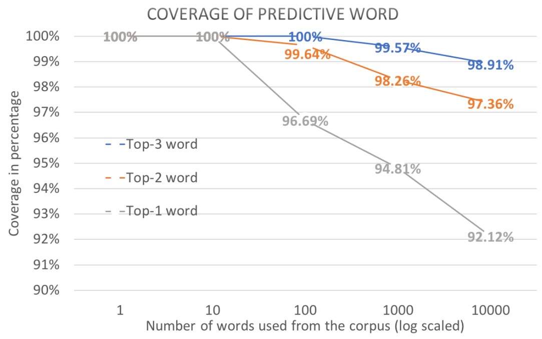

Fig. 8. Coverage of word disambiguation algorithm

disambiguation based on a corpus [35]. We first delete all the text strings containing non-alphabetic characters. Then, we build a hash table of all the valid words. In this hash table, the keys represent the coordinates in the 1-line alphabetical configuration and the words are the values. We sort all the words that have the same coordinate sequence by their frequency in the corpus. Therefore, our word disambiguation algorithm suggests the most frequent words (top-k word) on the basis of coordinate sequence in the mid-air.

Regarding the performance of word disambiguation in the keyboard-less configuration, we measure the ratio of the words in the dictionary that appear in the top k candidates under the given user's input. We simulate the inputs under the proposed statistical decoder using the dataset of keyboard-less configuration from the pilot study. Figure 8 shows the ratio of the words in the top k candidate word list.

V. Implementation and User Evaluation

We implement HIBEY on Microsoft Hololens. Similarly to the third experimental interface in Section IV, we tackle the uncertainty of keyboard-less environment and imprecision in mid-air. In our proposed system, the character keys are hidden. A single column of predicted words is positioned at the top edge of the interface. The predicted words are located on the top of Preparation and Fast-forward zone. In other words, we dedicate the small portion of the top area among these two zones to the word prediction function. Figure 9 is an illustrative interface of HIBEY. In the illustrative interface, we include the spectator mode to aid the explanation of the implemented system. The spectator mode is a colorful circle showing the hidden characters interacting with the user and the color hints follow the color arrangement in the layout (1) of the experimental interface (Figure 5 (top)). Note that the characters in the spectator mode should not be shown in the usual circumstance as well as the evaluation.

Figure 9.a shows what happens when the system does not detect a hand in the holographic environment. Figure 9.a – h demonstrate the procedures for typing the word 'IN'. The user initially puts his hand in the preparation zone and hovers over the desired character. Afterwards, the user moves his hand forward in fast-forward zone to pick the character. As the user mistypes the character 'H' instead of 'I'. The user selects the predicted word (the 2nd choice) to accomplish correction. Figure 9.i – l demonstrates the backspace function of the Recall zone. The word "ENGLAND" is mistyped as 'ENGLAC'. The user pulls his hand to the recall zone to perform the backspace function and thus delete the mistyped character 'C'. To speed up the text entry, the user can also use the predicted word (the 3rd choice).

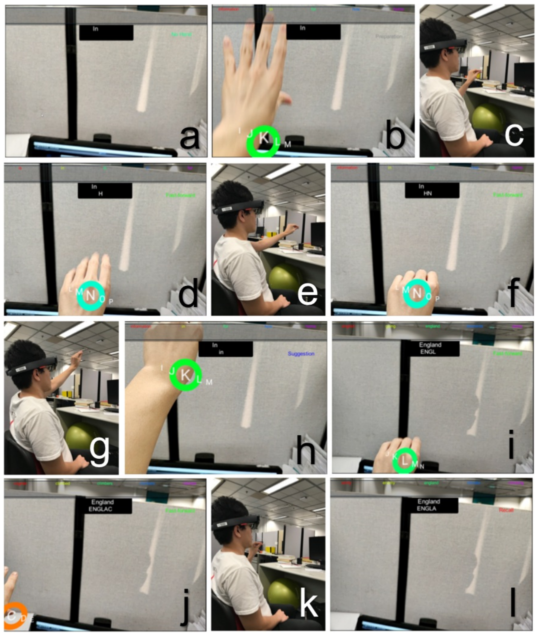

Fig. 9. Text entry illustration– a: Hand-off, b: Starting Mid-air tap in preparation zone, c: Hold the hand gesture in preparation zone (side view), d: Move the hand gesture horizontally to 'N', e: Hold the gesture in Fast-forward zone to pick a character (side view), f: the character 'N' is selected, g: Move the hand gesture to predicted word, h: the word changes from 'HN' to 'IN', i: another illustration snapshot, j: mistakenly pick character 'C', k: Move the hand gesture to Recall zone for backspacing the character 'C', l: removal of character 'C'

We design a text entry task to evaluate the system performance of HIBEY in terms of text entry speed and error rate. 18 Participants are invited to perform 8-day text input tasks under two text input conditions: 1-line and Non-key (HIBEY). The translucent layout is excluded as the study goal is to evaluate the keyboard-less approach. As such, we use the 1-line layout as our baseline. In the 1-line condition, the participants are able to see the 27 character keys and the predicted words. In the Non-key condition, the system only displays predicted words while the character keys are hidden. We further ask another 8 participants to perform 8 sessions of mid-air text input with the default QWERTY keyboard on Microsoft Hololens. For each condition, we show 25 word phrases in the optical screen of Microsoft Hololens and ask the participants to type the target words. During a briefing session of around 15 minutes, we explain the configurations in the three typing interfaces. The participants are instructed at the beginning of each session to type as fast as possible, and can correct typing mistakes only for the current word. Both

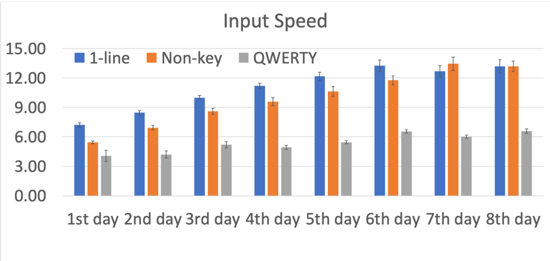

Fig. 10. Text input speed under three text input conditions

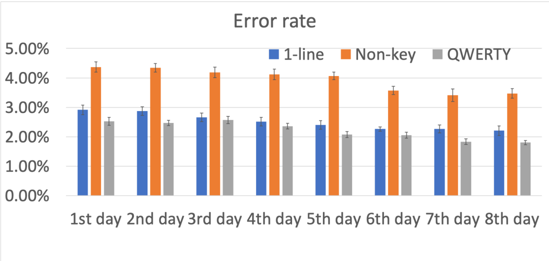

Fig. 11. Error rate under three text input conditions

single-character input and predictive word input (automatic word completion) are allowed. We extract the text material from MacKenzie & Soukoreff phrase set [28]. The material is written in a sophisticated yet recognizable style so the experiments are reasonable mock-ups of everyday typing tasks. We measure and compare the participants' typing speeds and error rates. On the first day and the eighth day of the experiment, we ask the participants to answer the NASA TLX [16] survey assessing the user experience of 1-line and Non-key configurations through six qualitative metrics: Mental Demand, Physical Demand, Temporal Demand, Performance, Effort, and Frustration.

a) Text entry speed: Figure 10 shows the word-level text entry rate [43] under the two conditions, where the error bars represent the standard deviation. Two-way repeated measures ANOVA yields a significant effect of the Text entry condition and Session ($F_{2,7}$ = 339.044, p <0.0001), indicating that different text entry conditions produce a different performance of text entry speed, accompanied with a learning effect between sessions on the new layout. Participants achieve 11.01 WPM (std. = 2.29) on average with the 1-line condition over the 8-day sessions. The average text entry rate increases to 13.18 WPM (std. = 1.32) on the 8th day from 7.21 WPM (std. = 0.41) on the 1st day, showing a 82.82% speed improvement. In contrast, the participants achieve 5.44 WPM (std. = 0.28) with the Non-key condition on the 1st day. The average text entry rate on the 8th day improves by 142.3% to 13.19 WPM (std. =1.10). Our results show that the initial performance of participants with the Non-key condition is only 75.48% of the 1-line condition. The performance of the Non-key condition surpasses the 1-line condition on the 7th day. The steep learning curve shows that the participants are still learning about HIBEY throughout the study. The baseline results of QWERTY keyboard on Microsoft Hololens shows an average 5.38 WPM (std. = 1.09), which starts from 4.07 WPM (std. = 1.15) in the 1st session and reaches 6.59 WPM (std. = 0.49) in the 8th session. The QWERTY keyboard relies on the head pointing technique to choose the characters on the keyboard, which leads to a slower speed due to ergonomic restriction of head movements.

b) Error rate: Figure 11 shows the word-level error rate [7] under the two text entry conditions, where the error bars represent the standard deviation. Two-way repeated measures ANOVA demonstrates a significant effect of the Text entry condition and the Session ($F_{2,7}$ = 70.353, p <0.0001), which indicates the significance of text entry conditions on error rates and the learning effect between sessions on similar text entry conditions. 1-line condition achieves a mean error rate of 2.51% (std. = 0.0038), which improves from 2.91% (std. = 0.0032) on the 1st day to 2.21% (std. = 0.0033) on the 8th day. In comparison, the Non-key condition achieves a mean error rate of 3.94% (std. = 0.0049). As expected, the initial high error rate of 4.37% (std. = 0.0035) on the 1st day is mainly caused by the unfamiliarity of the layout of the hidden character keys. Throughout the 2nd day and 5th day, we observe that the Non-key condition catches up the 1-line condition. On the 8th day, the error rate of Non-key condition decreases to 3.47% (std. = 0.0032), as the participants are able to memorize the relative position of the hidden character keys. The baseline results of QWERTY keyboard on Microsoft Hololens shows a mean error rate of 2.21% (std. = 0.0035), which starts from 2.53% (std. = 0.0026) in the 1st session and reaches 1.81% (std. = 0.0015) in the 8th session. The user familiarity to the QWERTY keyboard contributes to the consistent error rate lower than the above conditions.

c) NASA Task Load Index: Figure 12 shows the results of the user perception to the text entry conditions. One-way ANOVA with Bonferroni and Holm methods between sessions under two text entry conditions shows significant effects of the Text input conditions and the Session (p <0.0001) except for the physical demand between 1-line on the 1st day and Non-key on the 8th day (Bonferroni p-value = 0.0716), and the frustration metric between 1-line on the 8th day and Non-key on the 8th day (Bonferroni p-value = 0.6985). From the user rating, we can conclude that the participant's perceived load significantly decreased over the 8-day sessions. On the 1st day, participants are more predisposed to the 1-line condition than the Non-key condition. On the 8th day, the gap between two text entry conditions has narrowed, especially the frustration of participant that reaches a similar value for both the 1-line and Non-key conditions.

At the end of the study on the 8th day, we further show the text input interface to the participants and ask the following question: *Which interfaces do you prefer for typing tasks?*. 13 out of 17 participants choose Non-key text input due to the increased screen real estate, while the remaining 4 participants prefer to use the Microsoft Hololens' default keyboard because of the familiarity with the QWERTY soft keyboard layout. These four participants reflect that the Non-key text input approach is more counter-intuitive than the QWERTY soft

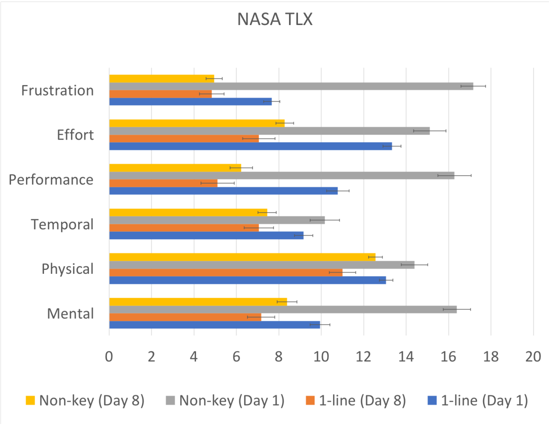

Fig. 12. Qualitative measures of user perception

keyboard layout. However, HIBEY takes only 13.14% of screen area at the edge position while the default QWERTY keyboard occupies 35.33% of screen area at the center position. The default QWERTY keyboard therefore needs 168.84% more space than HIBEY and meanwhile HIBEY reserves the center position in AR.

d) Discussion and Limitation: Regarding the text entry speed, HIBEY achieves a comparable performance to the existing works of text entry on smartglasses. We compare the text entry rate of HIBEY with other recently proposed selection-based methods on smartglasses: 1) PalmType [38], 2) 1Line keyboard [41], 3) 1D Hand writing [41], 4) Typing Ring [32], 5) External touch controller [30]. These solutions achieve typing speeds ranging from 6.47 to 10.01 WPM, while Non-key text entry approach has an average of 9.95 WPM over the 8 sessions and reaches 13.19 WPM on the last trial. Another prior work [40] using the full QWERTY soft keyboard achieves 23.0 – 29.2 WPM. Our work is far slower but unleashes most of the screen's real estate for the interaction in augmented reality.

As for the error rate, HIBEY results in an average of 3.94% error rate, which is slightly higher than the above works, for instance, PalmType (0.87%) and Typing Ring (1.34%). The presence of tactile feedback on the touch interface enables users to achieve more accurate input [38]. In fact, we are constrained by the hardware configurations and at a disadvantage of the absence of tactile feedback. This results in a more uncertain environment than the above approaches. In addition, picking a character key accurately in a primitive 1-line keyboard is difficult and 2 or 3 character offsets are considered as a comfortable option without paying visual attention to the keys [43]. Our proposed statistical decoder supports reliable character selection under the constrained environment. In addition, the tracking sensitivity on the pointing gesture may impact the user performance, due to the limitations of camera sensitivity and the computation resources on smartglasses. The users may not be able to effectively perform a sharp jump from one character to another, in order to make a last minute change in character selection. Typing mistakes result in unproductive times in the Recall zone during the task. Despite the higher error rates than concurrent solutions, users manage to achieve higher typing speeds thanks to the predictive word completion and the backspace function that allow to quickly correct typing mistakes.

Our work serves as the groundwork showing that the keyboard-less text entry does work in AR. Compared with the existing works, the key advantages of HIBEY are: 1) The lower disturbance to the physical environment as the reserved area for text input is significantly reduced and 2) No addendum sensors is required. The main limitation of HIBEY is the requirement of a depth camera to detect the traversing position across zones in a vertical arrangement, which is costly and not available on the lower-end AR headsets. More sophisticated approaches such as deep learning, and more advanced language models and hand gesture recognitions, should be conducted in the future to gain better insights on the performance of the proposed system in comparison to other existing methods in the literature. It would also be interesting to test the performance of HIBEY when the magnified effects (e.g. MacOS magnified icons) on the chosen characters under the 1-line invisible configuration in future works. Note that HIBEY is not limited to character selection, and can potentially extend the human-smartglasses interactions in multitudinous ways. For example, word disambiguation can be applied to ambiguous pronunciation in speech recognition, with corrections managed through freehand interactions.

VI. Conclusion

In this paper, we present HIBEY, a vision-based text entry system using one continuous gesture in the holographic environment of smartglasses without any additional ambient sensor and instrumental glove. Our work was implemented on Microsoft Hololens and thoroughly evaluated. Our evaluation shows that HIBEY is an easy-to-use and reliable solution achieving an average of 9.95 WPM with error rate of 3.94%, a comparable performance to other state-of-the-art methods. After 8 trials, users halved the perceived task load, reaching levels similar to the 1-line visible layout. Furthermore, HIBEY occupies only 13.14% screen area that is 62.80% less than the default virtual keyboard on Microsoft Hololens.

In future works, we plan to enhance the capabilities of HIBEY in several aspects. First, we will introduce ten-finger pointing gestures to improve the typing speed. Second, we will extend the system with speech recognition. Instead of typing the characters, the user will select ambiguous words in output sentences from voice input. Third, we will conduct a longitudinal study to improve our understanding of the long-term text entry performance of HIBEY.

VII. Acknowledgements

The authors thank the anonymous reviewers for their insightful comments. This research has been supported, in part, by projects 26211515, 16214817, and G-HKUST604/16 from the Research Grants Council of Hong Kong, and the 5GEAR project from the Academy of Finland ICT 2023 programme.

REFERENCES

[1] Asetniop keyboard, 2012. Retrieved from http://asetniop.com/.

[2] Minuum keyboard by whirlscape, 2015. Retrieved from http://minuum.com/.

[3] Fleksy keyboard, 2016. Retrieved from http://fleksy.com/.

[4] A. S. Arif, B. Iltisberger, and W. Stuerzlinger. Extending mobile user ambient awareness for nomadic text entry. In *Proceedings of the 23rd Australian Computer-Human Interaction Conference*, OzCHI '11, pages 21–30. ACM, 2011.

[5] S. Azenkot and S. Zhai. Touch behavior with different postures on soft smartphone keyboards. MobileHCI '12, pages 251–260. ACM, 2012.

[6] X. Bi, B. A. Smith, and S. Zhai. Quasi-qwerty soft keyboard optimization. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '10, pages 283–286. ACM, 2010.

[7] X. Bi and S. Zhai. Bayesian touch: A statistical criterion of target selection with finger touch. In *Proceedings of the 26th Annual ACM Symposium on User Interface Software and Technology*, UIST '13, pages 51–60. ACM, 2013.

[8] J. Clawson, K. Lyons, T. Starner, and E. Clarkson. The impacts of limited visual feedback on mobile text entry for the twiddler and mini-qwerty keyboards. In *Proceedings of the Ninth IEEE International Symposium on Wearable Computers*, ISWC '05, pages 170–177. IEEE Computer Society, 2005.

[9] B. Ens, A. Byagowi, T. Han, J. D. Hincapié-Ramos, and P. Irani. Combining ring input with hand tracking for precise, natural interaction with spatial analytic interfaces. In *Proceedings of the 2016 Symposium on Spatial User Interaction*, SUI '16, pages 99–102. ACM, 2016.

[10] J. Gong and P. Tarasewich. Alphabetically constrained keypad designs for text entry on mobile devices. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, CHI '05, pages 211–220. ACM, 2005.

[11] J. Goodman, G. Venolia, K. Steury, and C. Parker. Language modeling for soft keyboards. In *Proceedings of the 7th International Conference on Intelligent User Interfaces*, IUI '02, pages 194–195. ACM, 2002.

[12] N. Green, J. Kruger, C. Faldu, and R. St. Amant. A reduced qwerty keyboard for mobile text entry. In *CHI '04 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '04, pages 1429–1432. ACM, 2004.

[13] T. Grossman, X. A. Chen, and G. Fitzmaurice. 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, pages 144–152. ACM, 2015.

[14] J. D. Hincapié-Ramos, X. Guo, and P. Irani. The consumed endurance workbench: A tool to assess arm fatigue during mid-air interactions. In *Proceedings of the 2014 Companion Publication on Designing Interactive Systems*, DIS Companion '14, pages 109–112. ACM, 2014.

[15] Z. Huang, W. Li, and P. Hui. Ubii: Towards seamless interaction between digital and physical worlds. In *Proceedings of the 23rd ACM International Conference on Multimedia*, MM '15, pages 341–350. ACM, 2015.

[16] N. A. R. C. Human Performance Research Group. Nasa task load index (tlx), 1999.

[17] H. Istance, R. Bates, A. Hyrskykari, and S. Vickers. Snap clutch, a moded approach to solving the midas touch problem. In *Proceedings of the 2008 Symposium on Eye Tracking Research &#38; Applications*, ETRA '08, pages 221–228. ACM, 2008.

[18] R. J. Jacob, A. Girouard, L. M. Hirshfield, M. S. Horn, O. Shaer, E. T. Solovey, and J. Zigelbaum. Reality-based interaction: A framework for post-wimp interfaces. In *Proceedings of SIGCHI Conference on Human Factors in Computing Systems*, CHI '08, pages 201–210. ACM, 2008.

[19] H. Kim, Y.-k. Row, and G. Lee. Back keyboard: A physical keyboard on backside of mobile phone using qwerty. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, pages 1583–1588. ACM, 2012.

[20] B. Kollee, S. Kratz, and A. Dunnigan. Exploring gestural interaction in smart spaces using head mounted devices with ego-centric sensing. In *Proceedings of the 2Nd ACM Symposium on Spatial User Interaction*, SUI '14, pages 40–49. ACM, 2014.

[21] D. L. G. T. K. A. Kushler. Reduced keyboard disambiguating computer, May 7 1995. US Patent 5818437A.

[22] L. Lee and P. Hui. Interaction methods for smart glasses: A survey. *IEEE Access*, 6:28712–28732, 2018.

[23] T. Lee and T. Hollerer. Handy ar: Markerless inspection of augmented reality objects using fingertip tracking. In *2007 11th IEEE International Symposium on Wearable Computers*, pages 83–90, Oct 2007.

[24] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong. The 1line keyboard: A qwerty layout in a single line. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 461–470. ACM, 2011.

[25] S. Li, A. Ashok, Y. Zhang, C. Xu, J. Lindqvist, and M. Gruteser. Whose move is it anyway? authenticating smart wearable devices using unique head movement patterns. pages 1–9, 03 2016.

[26] S. Ma, Q. Liu, C. Kim, and P. Sheu. Lift: Using projected coded light for finger tracking and device augmentation, 03 2017.

[27] I. S. MacKenzie, H. Kober, D. Smith, T. Jones, and E. Skepner. Letterwise: Prefix-based disambiguation for mobile text input. In *Proceedings of the 14th Annual ACM Symposium on User Interface Software and Technology*, UIST '01, pages 111–120. ACM, 2001.

[28] I. S. MacKenzie and R. W. Soukoreff. Phrase sets for evaluating text entry techniques. In *CHI '03 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '03, pages 754–755. ACM, 2003.

[29] R. McCall, B. Martin, A. Popleteev, N. Louveton, and T. Engel. Text entry on smart glasses. In *2015 8th International Conference on Human System Interaction (HSI)*, pages 195–200, June 2015.

[30] R. McCall, B. Martin, A. Popleteev, N. Louveton, and T. Engel. Text entry on smart glasses. In *2015 8th International Conference on Human System Interaction (HSI)*, pages 195–200, June 2015.

[31] T. Ni, D. A. Bowman, C. North, and R. P. McMahan. Design and evaluation of freehand menu selection interfaces using tilt and pinch gestures. *Int. J. Hum.-Comput. Stud.*, 69(9):551–562, Aug. 2011.

[32] S. Nirjon, J. Gummeson, D. Gelb, and K.-H. Kim. Typingring: A wearable ring platform for text input. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, MobiSys '15, pages 227–239. ACM, 2015.

[33] I. Poupyrev, D. S. Tan, M. Billinghurst, H. Kato, H. Regenbrecht, and N. Tetsutani. Developing a generic augmented-reality interface. *Computer*, 35(3):44–50, Mar. 2002.

[34] J. Scott, S. Izadi, L. S. Rezai, D. Ruszkowski, X. Bi, and R. Balakrishnan. Reartype: Text entry using keys on the back of a device. In *Proceedings of the 12th International Conference on Human Computer Interaction with Mobile Devices and Services*, MobileHCI '10, pages 171–180. ACM, 2010.

[35] T. Segaran and J. Hammerbacher. *Beautiful Data: The Stories Behind Elegant Data Solutions*. Theory in practice. O'Reilly Media, 2009.

[36] Y.-C. Tung, C.-Y. Hsu, H.-Y. Wang, S. Chyou, J.-W. Lin, P.-J. Wu, A. Valstar, and M. Y. Chen. User-defined game input for smart glasses in public space. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3327–3336. ACM, 2015.

[37] W. S. Walmsley, W. X. Snelgrove, and K. N. Truong. Disambiguation of imprecise input with one-dimensional rotational text entry. *ACM Trans. Comput.-Hum. Interact.*, 21(1):4:1–4:40, Feb. 2014.

[38] C.-Y. Wang, W.-C. Chu, P.-T. Chiu, M.-C. Hsiu, Y.-H. Chiang, and M. Y. Chen. Palmtype: Using palms as keyboards for smart glasses. MobileHCI '15, pages 153–160. ACM, 2015.

[39] D. Wigdor, C. Forlines, P. Baudisch, J. Barnwell, and C. Shen. Lucid touch: A see-through mobile device. In *Proceedings of the 20th Annual ACM Symposium on User Interface Software and Technology*, UIST '07, pages 269–278. ACM, 2007.

[40] X. Yi, C. Yu, M. Zhang, S. Gao, K. Sun, and Y. Shi. 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 &#38; Technology*, UIST '15, pages 539–548. ACM, 2015.

[41] C. Yu, K. Sun, M. Zhong, X. Li, P. Zhao, and Y. Shi. One-dimensional handwriting: Inputting letters and words on smart glasses. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, CHI '16, pages 71–82. ACM, 2016.

[42] S. Zhai and B. A. Smith. Alphabetically biased virtual keyboards are easier to use: Layout does matter. In *Extended Abstracts on Human Factors in Computing Systems*, CHI '01, pages 321–322. ACM, 2001.

[43] M. Zhong, C. Yu, Q. Wang, X. Xu, and Y. Shi. Forceboard: Subtle text entry leveraging pressure. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*, CHI '18, pages 528:1–528:10. ACM, 2018.