Received April 22, 2019, accepted July 3, 2019, date of publication July 16, 2019, date of current version August 2, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2929310

Designing Hand Pose Aware Virtual Keyboard With Hand Drift Tolerance

DAEWOONG CHOI[1], HYEONJOONG CHO[1], KYEONGEUN SEO[1], SANGYUB LEE[2], JAEKYU LEE[2], AND JAEJIN KO[2]

[1]Department of Computer Convergence Software, Korea University, Sejong-si 30019, South Korea
[2]Korea Electronics Technology Institute, Seongnam-si 13488, South Korea

Corresponding author: Hyeonjoong Cho (raycho@korea.ac.kr)

This work was supported in part by the IT Research and Development program of MOTIE/KEIT under Grant 10085608 (Developing of independent workable low-power embedded artificial intelligent module and navigation application service.)

ABSTRACT An unintentional hand drift adversely affects the typing performance of conventional virtual keyboards. To overcome this, we proposed to utilize the typing patterns of skilled typists. First, as most typists enter the keys in the same column with a predetermined finger only, we restricted these keys to be typed by their corresponding fingers. Second, our investigation of skilled typists demonstrated that hand poses vary when the typists touch different keys. Thus, rather than locating the touch point as in the case of existing virtual keyboards, we attempted to use unique hand poses to infer the target key. Based on these two techniques, we implemented a novel hand poses aware virtual keyboard that is tolerant of hand drift. Our experimental studies yielded the following results: 1) most of the QWERTY-familiar typists who have varying typing habits were easily adaptable to the proposed keyboard design and 2) the proposed keyboard outperformed existing virtual keyboards in terms of typing speed and several error rates, and eventually achieved a typing speed of approximately 56 WPM.

INDEX TERMS Eyes-free, hand drift, hands, text entry, touch typing, virtual keyboard, virtual reality.

I. INTRODUCTION

Despite the prevalence of touchscreens, ten-finger typing on virtual keyboards is nevertheless slower and less accurate than physical keyboards [8]. The decreased performance of virtual keyboards primarily results from the absence of tactile feedback [1], [3], [8], [32], [36], which occasionally cause hand drifts: users unintentionally move their hands while typing. To improve virtual keyboards, some studies attempted to compensate for the lack of tactile feedback [15], [19], [23], [36] and other studies proposed adaptive keyboard layouts [9], [11], [32], [40]. Although these efforts were able to improve the existing virtual keyboards to a certain extent, the virtual keyboards have not yet reached to their potential maximum typing performance (59 WPM) [8]

In conventional text entry [13], [30], [31] using physical keyboards, typing errors, i.e., a target key is mistakenly replaced with the neighboring key, occurred 43% in the horizontal and 15% in the vertical directions, respectively. Owing to the hand drift, virtual keyboards may be more vulnerable to these typing errors than physical keyboards. Therefore, eliminating the horizontal and vertical typing errors is crucial for improving virtual keyboards. To accomplish this, we focused on the typing patterns of skilled typists.

Figure 1 shows key allocation to each finger for skilled typists. To reduce the horizontal typing errors, we propose to permit each key to be entered by a pre-assigned finger only, as shown in Figure 1. We referred to this technique as *Key Pre-allocation*. As Key Pre-allocation is the same as the common practice adopted by most skilled typists who have typed on QWERTY over the past few decades [5], this technique accommodates most of the skilled typists without extra training. Although some skilled typists type differently from Key Pre-allocation [41], our experimental results showed that 67% of these typists can adapt to this technique in a short time. As Key Pre-allocation limits each key to be entered with the assigned finger only, it prevents most of the horizontal typing errors in advance.

Furthermore, to interpret typists' intentions more precisely, we propose to infer a target key based on hand poses. We referred to this technique as *Key Inference based on Hand Poses*. Our experimental findings supported that when touching a target key, the typist's hand makes a unique pose. Therefore, identifying the unique hand pose for a target key

The associate editor coordinating the review of this manuscript and approving it for publication was An-An Liu.

This work is licensed under a Creative Commons Attribution 4.0 License. For more information, see http://creativecommons.org/licenses/by/4.0/

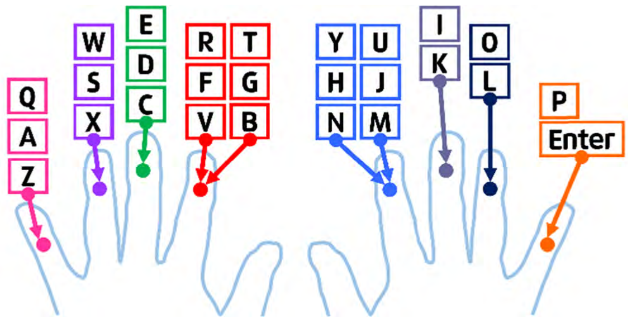

FIGURE 1. Key Pre-allocation that assigns each key column to its corresponding finger. Each key can be entered with its allocated finger only; therefore most horizontal typing errors can be eliminated in advance.

can grasp the typists' intention accurately. As hand pose based inference is independent of where the hand is located, this technique further allows the key inference to be tolerant towards hand drift.

These two techniques allow us to implement a novel virtual keyboard that reduces typing errors and tolerates hand drift as well. Our experimental results showed that the proposed keyboard outperformed existing virtual keyboards in terms of typing speed and several error rates, and eventually reached the typing speed of approximately 56 WPM. Furthermore, we will discuss about the feasible applications of our novel keyboard. We will also discuss how to apply our techniques to other skilled typists who experience difficulty in adapting to Key Pre-allocation.

II. RELATED WORK

In this section, we explore the studies on virtual keyboards that inspired our work.

A. TEXT ENTRY ON TOUCHSCREENS

As touchscreen-based virtual keyboards imitate QWERTY layouts without considering their underlying differences, i.e., there is no tactile feedback, they are suffering from decreased typing performance compared to physical keyboards [3], [8], [33], [37]. However, Findlater *et al.* showed that in an ideal condition where typing errors do not occur, typing on virtual keyboards could be as fast as that on physical keyboards. In addition, they found that even without a keyboard layout provided visually, skilled typists could reach fairly high accuracy (approximately 90%). In conclusion, they emphasized personalization—such as customizing each key size and curved shape of the keyboard layout—as an important factor for future keyboard design. Li and Findlater *et al.* further observed the effect of hand drift on a visible conventional keyboard and an invisible adaptive keyboard [18]. They found that unintentional hand drift occurred on both keyboards and hands moved in the up and left directions.

For improving the virtual keyboards, some researchers suggested providing a modified QWERTY layout. They proposed that the virtual keyboard has a layout adapted to either the user's typing patterns [9], [11] or their natural finger positions on touchscreens [32]. Nevertheless, showing a visually adapted layout to users had a negative impact on performance compared to showing a stable rectangular layout [9]. Shi *et al.* [40] tried to adjust the location and size of the layout in real time based on successive touch inputs. They further used relative locations between the inputs, but their keyboard predicted a target word rather than a target character.

There have been other studies focusing on tactile feedback. Weiss *et al.* [36] suggested putting a keyboard-shaped silicon rubber on a touchscreen; however, performance evaluation has not yet been reported. As alternatives to tactile feedback, haptic and sound feedbacks were proposed as well [19], [23]. However, the auditory feedback did not improve performance; the haptic keyclick feedback was found to be helpful to improve typing speed and error rate.

In addition, Kim *et al.* suggested replacing tactile feedback by changing the text entry process on touchscreen keyboards [15]. They noted that on physical keyboards a character is entered through three stages of Touch-Press-Release, while on conventional soft keyboards a character is entered without the Press stage. To complement the missing Press stage on the touchscreen, they proposed a keyboard, TapBoard. The TapBoard has a certain threshold (< 300 ms) between the Touch and Release stages and the touch gestures released within the threshold were only allowed to enter characters. It prevents users from noticing the difference between the TapBoard and the existing virtual keyboards and at the same time, it enables users to perform additional multi-touch gestures, such as putting all fingers on the touchscreen for resting.

B. TEXT ENTRY ON FLAT SURFACES

The tablet PC's built-in cameras [25], [39] and infrared touch sensing technologies [25] are making it possible to use virtual keyboards on any flat surface beyond the touchscreens. However, several well-known problems of image processing—such as occlusion, lighting condition, etc.—hinder the techniques from being prevalent. Thus, in order to make virtual keyboards on flat surfaces feasible, studies for accurate finger gesture recognition should be preceded. For these reasons, detailed performance assessments of the state-of-the-art gesture recognition based virtual keyboards were not reported, or their performance was shown to be much worse than that of the conventional touchscreen keyboards (1.5–2.5 characters per second) [39].

C. USING FINGER IDENTITY FOR VIRTUAL KEYBOARDS

For providing a richer set of interaction, finger identification on touchscreens has been receiving considerable attention recently [22]. However, although several prototypes capable of finger identification have been proposed, only a few techniques have been applied to virtual keyboards, e.g., a prototype identifying two fingers [14], the other identifying fingers in one hand [22]. It implies that the finger identification techniques are still premature to reliably identify ten fingers in real time.

Rather than directly using finger identification, there have been some approaches to use the mappings between finger identity and QWERTY layout for typing on touchscreens [17], [32]. The 1-line keyboard [17] combines three keys in the same column into a single soft button which results in a total of 10 soft buttons on one line. Then, to solve ambiguity, they suggested looking into the sequences of inputs from the one line buttons to infer words that users want to enter. On Liquid Keyboard [32], when users put their hands on the touchscreen as they do on conventional physical keyboards, each key column is placed according to the direction of their hands and fingers, which is inferred from geometric mappings between touch-points and finger identities. Nevertheless, their proposed keyboard designs were quite slow as well (< 30 WPM) or not reported.

Meanwhile, Choi *et al.* tried to improve the virtual keyboard considering two-dimensional positions of fingertips [4]. Through observing typing patterns of skilled typists, they found that there were 2D positional correlations among all fingers when typing. Based on the 2D positional correlations, they tried to more accurately infer the target key that users wanted to enter. However, their proposed keyboard was slightly limited to showing the actual improvement in typing performance ($M = 31.618$ WPM) for the following two reasons. First, their keyboard was vulnerable to hand drift. Second, their experimental setups had some unreliability, such as an unintended touch recognition and lighting problem of color marker detection.

In this study, we extend their work as follows. First, we improved our experimental setup to detect the marker positions on hands more precisely, which restricted the detecting error to be less than 1 mm. Second, we tried to infer a target key more precisely using hand poses which are tolerant to hand drift. Third, our key inference system was not affected by the difference in hand sizes between typists and was approximately 9.5% more accurate than their key inference models.

III. TOUCH SURFACE

For investigating the typing patterns of skilled typists, we were first required to identify hand poses when they were typing on virtual keyboards. To do this, we implemented a touch surface capable of estimating hand poses during typing.

A. APPARATUS

Figure 2(A) shows our experiment environment. To construct a touch surface, we used seven infrared cameras, named OptiTrack Flex13, and several reflective markers with a 2 mm radius. When the infrared cameras were connected to a hub, the cameras were able to track the infrared reflective markers with an error range of less than 1 mm. Users were requested to sit in front of the touch surface and then enter the phrase presented on display. Figure 2(B) shows the 3D coordinate system on the surface. On the touch surface, the origin was located at the center of the lower edge of the surface. With respect to the origin, the positive directions of x-, y-, and z-axis indicated the leftward, upper, and forward directions, respectively.

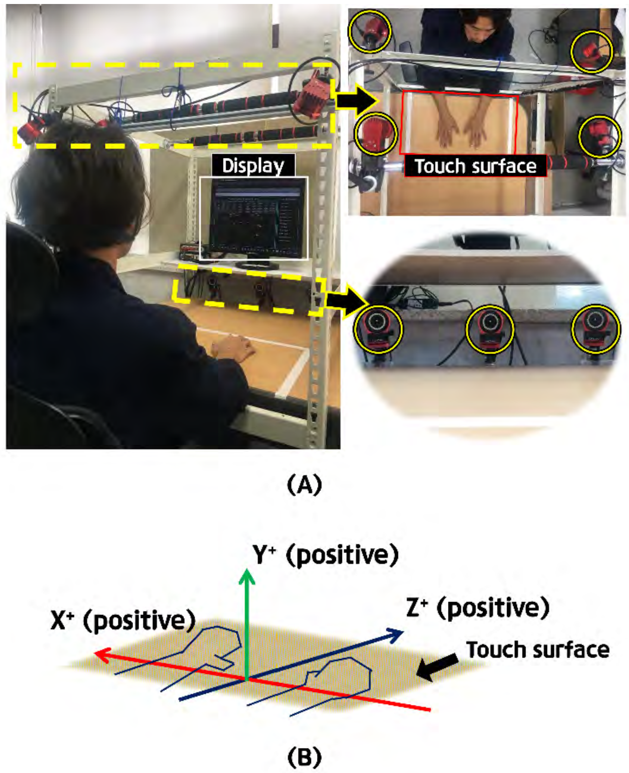

FIGURE 2. (A) Experimental setup for tracking hand poses. Seven infrared cameras were connected to a hub that accurately tracks the infrared reflective markers attached to the hands in real time. (B) 3D coordinate system on our touch surface.

All participants were requested to attach 18 reflective markers on their hands. Figure 3(A) shows the nine markers attached to the left hand. One marker was attached on the wrist, and the other markers were attached on the fingertips and metacarpo-phalangeal (MCP) joints of each finger. As the middle and ring fingers had no markers attached to their MCP joints, we considered those positions to be virtually located at regular spaces between the MCP joints of the index and little fingers. To express the hand pose of one hand, two vectors per finger were calculated. Based on the MCP joint, $\overrightarrow{MW}_n$ and $\overrightarrow{MF}_n$ vectors point at the wrist and fingertip, respectively. Therefore, we expressed a hand pose of one hand with ten vectors, as shown in Figure 3(B).

Figure 4 shows the user interface of our experimental software on an external 19'' monitor (1440×900 resolution). At the top of the screen, the progress bar of the current session, the presented phrase, and the user input stream were displayed. Below them, the keyboard layout and the positions of the joints (red circles) and fingertips (orange circles) were presented. In the right-hand side, User Information area showed several conditions and performance results of the

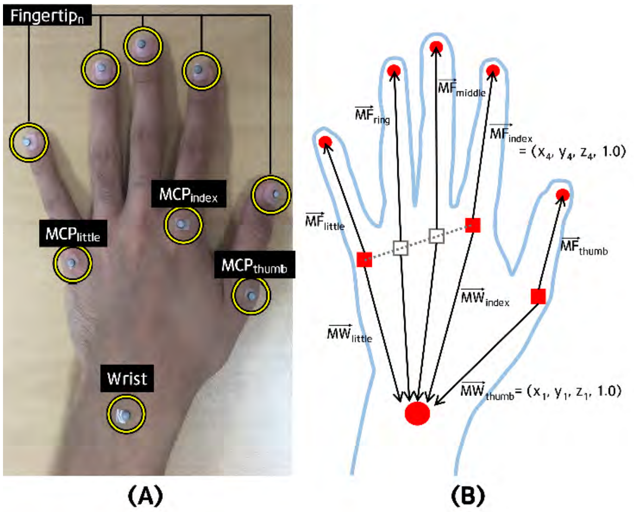

FIGURE 3. (A) Nine infrared reflective markers attached to the left hand and (B) ten vectors for representing a hand pose. Metacarpo-phalangeal (MCP) joints of the middle and ring fingers were estimated by locating at regular spaces between the MCP joints of the index and little fingers.

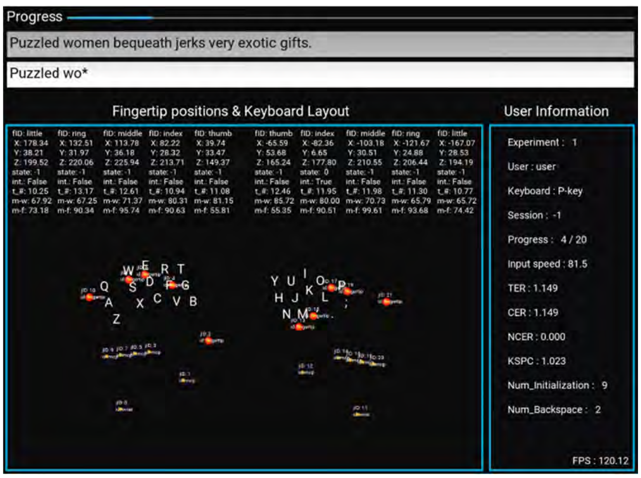

FIGURE 4. Screenshot of the user interface for experiments. It shows some information for typing, such as the progress of the current session, presented phrase and input stream, keyboard layout and positions of joints (orange circles), and fingertips (red circles), etc.

current session. Every time a touch event occurred, the software recorded a set of data, including all the vectors representing the hand poses.

B. TYPING INTERFACE

In our typing interfaces, a touch gesture using one finger was used to enter characters, including 26 English alphabets, Space, and Enter. For better usability, we supported two additional multi-touch gestures based on the results of some previous studies.

First, for providing each participant with a personalized keyboard layout, our interface requested an initialization step at the start of each session. In the initialization step, a participant puts his/her hands on the touch surface naturally and then our interface generated a keyboard layout based on the touch positions of all fingers. As shown in Figure 4, the example of the generated keyboard layout was also mostly curved shaped. As fingers may often rest on the surface [8], we permitted the keyboard to be regenerated whenever the participants put their hands on the surface.

Second, to substitute for the backspace key, we supported a backspace gesture to be performed by touching all fingers of the right hand anywhere on the touch surface. Some of the previous studies on virtual keyboards [4], [8] substituted a right-to-left swipe of right hand for the backspace key. However, as our backspace gesture is performed with one-step motion, we thought that it would contribute to the faster typing speed than the two-stage motion of the swipe gesture. For deleting multiple characters, users needed to repeat the backspace gesture.

To support the multi-touch gestures, including the initialization step and backspace gesture, the touch recognition of TapBoard [15] was used for our typing interface. Whereas the traditional virtual keyboards allow a character to be entered whenever a touch event occurs, the TapBoard allows a character to be entered only if a finger touches and then releases within a certain time threshold ($<$ 300 ms). As this design did not interfere with typing but enabled additional multi-touch gestures, e.g., resting with all fingers touching on the touch screen, we adopted it as the underlying touch recognition for our typing interface.

IV. INVESTIGATION OF TYPING PATTERNS OF SKILLED TYPISTS

As some QWERTY-familiar typists may type differently from our Key Pre-allocation [41], we were first required to check which typists could adapt well to Key Pre-allocation. To do this, we requested 15 skilled typists to type on our touch surface. Then, through observing the typing patterns of their hands, we statistically verified that the use of hand poses to infer a target key helps to improve the virtual keyboard typing performance.

A. PARTICIPANTS

15 participants (5 females), with ages between 23 to 32 years ($M = 26.6$), who had regularly typed on physical and touchscreen keyboards with QWERTY layout were recruited. To verify proficiency in English typing, we ran a simple typing test with physical keyboards before the main experiments, where they were requested to type the Mackenzie phrases set [20]. As a result, they demonstrated fast typing ($M = 65.4$ WPM, $SD = 21.6$) and rarely glanced down at the keyboard layout during typing; we regarded them as skilled typists [5], [13].

B. KEYBOARD DESIGN

The participants were requested to type on two keyboards: Key Pre-allocation-disabled and enabled ones. First, the Key Pre-allocation-disabled keyboard was abbreviated as *NKP keyboard*. The participants could type on the NKP keyboard as they have previously typed on existing

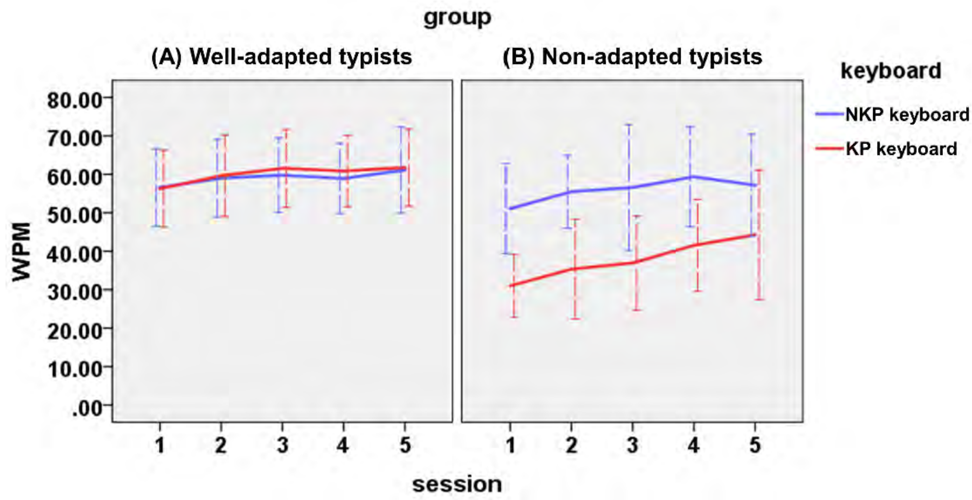

FIGURE 5. On the NKP and KP keyboards, typing speed (WPM) per session for (A) well-adapted typists and (B) non-adapted typists.

QWERTY keyboards. The other keyboard was based on Key Pre-allocation and therefore each finger could enter the pre-allocated keys only. For abbreviation, we called this keyboard as *KP keyboard*.

We wanted to observe hand poses when the skilled typists entered keys correctly. Therefore, on both the keyboards, we requested the participants to use backspace gesture when they made a mistake. In addition, on the KP keyboard, a character was entered only when the key was correctly typed by the assigned finger. When mistakes occurred, the typing data were ignored and a red warning window flickered on the display. Therefore, we considered the collected typing data as the data the typists thought they had entered correctly.

C. PROCEDURE

All the participants typed on the first keyboard and after at least 3 days, they typed on the second keyboard. For each keyboard, we requested the participants to complete 20 practice phrases on our touch surface to make them familiar with the touch surface and keyboard design. After the practice, they were requested to type 200 test phrases over 5 sessions on each keyboard. Half of the test phrases were randomly selected from the Mackenzie phrases set [20] and to consider all letters in the alphabet, the other half consisted of English pangrams. Each phrase of the pangrams contained all the letters of the alphabet in one sentence, such as "the quick brown fox jumps over the lazy dog." Whenever the participants wanted to change the keyboard layout or their hand positions, we permitted them to regenerate the keyboard. Additionally, we requested the participants to type naturally and accurately, with their hands put on the surface.

D. RESULTS

Through the first typing experiment, 90,016 typing data for the NKP keyboard and 90,123 typing data for the KP keyboard were collected. To check the applicability of Key Pre-allocation, we compared the two keyboards in terms of typing speed over five typing sessions. Then, we statistically verified that hand poses became different for different input keys.

1) APPLICABILITY OF KEY PRE-ALLOCATION

To measure typing speed, we calculated words per minute (WPM) following Mackenzie [21]:

$$WPM = \frac{|T| - 1}{S} \times 60 \times \frac{1}{5} \tag{1}$$

where $|T|$ is the length of the final transcribed string and S is the elapsed time in seconds. For analysis, we used two-way repeated measures analysis of variance (ANOVA) followed by the Tukey post-hoc analysis. All significant findings are reported in the 95% confidence interval.

As a result, we divided the participants into two groups. Figure 5 shows a typing speed over the five sessions for each group. As shown in Figure 5(A), the first group consisted of 10 out of 15 participants and they had no difference in typing speed between the two keyboards ($F = 2.019$, $p = 0.189$). We called this group as well-adapted typists. As shown in Figure 5(B), the other group consisted of 5 participants and they had the slower typing speed on the KP keyboard ($M = 37.82$, $SD = 10.284$) than the NKP keyboard ($M = 55.94$, $SD = 9.720$, $F = 36.445$, $p < 0.05$). We called this group as non-adapted typists.

The well-adapted typists maintained their typing speeds high on both keyboards since the first session ($F = 6.324$, $p < 0.05$). There were only three typists in this group whose finger-to-key mapping matched Key Pre-allocation. The other seven well-adapted typists had several inconsistent or different finger allocations from Key Pre-allocation. For inconsistent finger allocation, i.e., an alphabetic key was entered with multiple fingers, we found that on average,

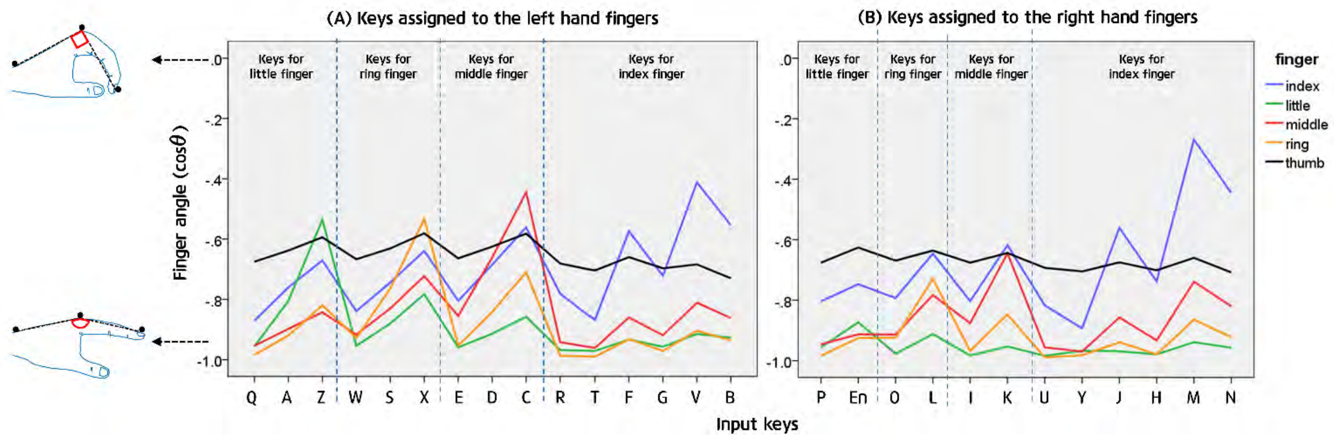

FIGURE 6. Angle (θ) variation of each finger for all input keys. Each figure shows average angles (cos θ) of each finger for input keys assigned to the (A) left hand and (B) right hand. All finger angles varied for each input key.

each participant enters 4.28 keys with multiple fingers inconsistently ($SD = 4.527$). For different key allocations, i.e., an alphabetic key was entered with a different finger from Key Pre-allocation, each participant entered an average of 0.71 keys ($SD = 0.699$) with a different finger.

The non-adapted typists were not adapted well to our Key Pre-allocation within a limited session. Their typing speeds on the KP keyboard increased over the sessions ($F = 10.779$, $p < 0.001$); however, they appeared to require more time for perfect adaptation. In the last session, their typing speeds on the KP keyboard ($M = 44.26$ WPM, $SD = 12.134$) was approximately 77% slower than those on the NKP keyboard ($M = 57.16$ WPM, $SD = 9.542$). On average, they entered inconsistently with multiple fingers for 12.0 keys ($SD = 7.925$) and entered 7.6 keys ($SD = 5.004$) with different fingers.

As the non-adapted typists were adapting throughout each session, it was expected that their typing data were erroneous and noisy. Therefore, we did not use their typing data for further analysis and keyboard configuration. Though, we will discuss how our keyboard design can benefit a variety of typists in Discussion section.

2) UNIQUE HAND POSES FOR EACH KEYS

The use of our hand joint vectors shown in Figure 3(B) enabled us to estimate the unique pose of each finger. By observing the poses of all fingers, we attempted to verify the hypothesis that hand pose for entering each input key is unique.

To estimate the pose of each finger, we measured the angle ($\boldsymbol{\theta}$) between the hand joint vectors as a representative metric. It is indicative of the degree of finger bending. The angle was calculated as a cosine function as follows.

$$cos\theta_n = \frac{\overrightarrow{MW}_n; \overrightarrow{MF}_n}{\left\|\overrightarrow{MW}_n\right\| \times \left\|\overrightarrow{MF}_n\right\|}, \quad n \in \{all\ fingers\} \qquad (2)$$

Each finger was associated with its $\overrightarrow{MW}_n$ and $\overrightarrow{MF}_n$ vectors. In our case, the cos$\boldsymbol{\theta}$ had a range of [-1, 0]. When the cos$\boldsymbol{\theta}$ for a finger is -1.0, the finger is spread out such that its $\overrightarrow{MF}_n$ vector is in line with its $\overrightarrow{MW}_n$ vector. As the cos$\boldsymbol{\theta}$ for a finger increases, the finger becomes more bent.

The analysis was carried out as follows. First, we confirmed that the correlations between the finger angles exist by Pearson's linear correlation analysis ($p < 0.05$). Therefore, a detailed analysis should be carried out to suppress the effect on the correlations when checking whether the finger angles varied depending on input keys. To do this, we used the one-way multivariate analysis of variance (one-way MANOVA) with Pillai's trace criterion. If the results of MANOVA are significant, then one-way ANOVA was used to check which finger angle varied for each key, followed by the Tukey post-hoc analysis.

The result shows that all finger angles varied for input keys (left hand: $F = 22.251$, $p < 0.001$, right hand: $F = 17.204$, $p < 0.001$). Figure 6 shows the average angles of each finger for the keys assigned to the (A) left and (B) right hands. Depending on the keys assigned to each finger, the angle of a touching finger always had the most significant variation compared to the other non-touching fingers. Though, the surrounding non-touching fingers showed correlated movements to the touching finger. For example, when the participants entered the higher row keys, such as the Q, W, and E keys, the non-touching fingers were stretched out together with the touching finger. When entering the lower row keys, all fingers became more bent than when entering the other row keys.

Exceptionally, there were no differences in finger angles between when entering the Y and U keys for all fingers including the touching finger ($p > 0.05$). To differentiate these two keys, we were required to analyze another hand characteristic that affects the global hand position, such as the hand direction. We checked whether the hand direction was different depending on input keys using the same analysis used for analyzing finger angles. The hand direction was estimated as follows.

$$\textit{hand direction} = \frac{\overrightarrow{WM}_{index} + \overrightarrow{WM}_{little}}{\left\| \overrightarrow{WM}_{index} + \overrightarrow{WM}_{little} \right\|} \tag{3}$$

With respect to the position of the wrist, $\overrightarrow{WM}_{index}$ and $\overrightarrow{WM}_{little}$ are the vectors pointing to the positions of the MCP joints of the index and little fingers, respectively. To eliminate the effect of different hand sizes between the participants, the vector of hand direction was converted into a unit vector.

As a result, the hand direction also varied for each input key (left hand: $F = 14.661$, $p < 0.001$, right hand: $F = 16.637$, $p < 0.001$). Especially, in the case of the Y and U keys, where there were no differences between the finger angles, the hand direction when entering the Y key was different from the direction when entering the U key ($p < 0.001$). When entering the Y key, the right hand turned left by approximately 8 degrees compared to when entering the U key (average hand direction: Y key = (0.269, -0.124, 0.949), U key = (0.200, -0.007, 0.971)).

All the results of the investigation implied that typists type with unique hand poses (including finger angles, etc.) and hand directions for each key. Therefore, we were highly confident that identifying the unique hand pose for each key improves the virtual keyboards.

V. A NOVEL VIRTUAL KEYBOARD BASED ON HAND POSES

The investigation of typing patterns encouraged us to implement a new hand pose-aware virtual keyboard. For implementation, we constructed key inference system that can infer a user's target key using our hand joint vectors. Our key inference system contains data preprocessing and key inference models. To construct the key inference models, we experimentally found machine learning models with optimal key inference accuracy for the hand joint vector.

A. DATA PREPROCESSING

The input vector to our key inference system represents a hand pose with ten hand joint vectors, as shown in Figure 3(B). Each hand joint vector consists of 4 dimensional (4D) features including the positional values on three-dimensional (3D) axes and a constant which is initially set to 1.0.

However, the original form of the input vector had a high variance problem that represented the differences in hand size between typists. For example, even when two hands are in the same pose, the larger hand generates higher 3D positional values. As data with low variance are generally easy to classify [7], we attempted to reduce the impact of the hand size differences by refining all the input vectors through two stages of simple preprocessing, normalizing, and scaling. After two stages of the preprocessing, the constant value within each joint vector contains information about the original vector, including his/her finger length.

During the first stage of the preprocessing, the *Normalizer* transformed each hand joint vector to be a unit vector. As the Normalizer extracted the unit vectors independently from each hand size, it reduced the variance of our typing data. Then, in the second stage, the *StandardScaler* made each feature of the unit vector follow the standard normal distribution. As the preprocessing resulted in better classification performance in various machine-learning applications [16], we expected the same effect in our key inference model.

B. MODEL SELECTION

After the two stages of data preprocessing, our key inference models infer a user's target key from the refined unit vector. Our key inference models contain principal component analysis (PCA) and Multi-layer perceptron (MLP). As the combinations between the PCA's output and MLP's parameters affect the performance of the key inference system, we compared several combinations and finally, chose the optimal parameter set in terms of key inference accuracy.

1) PRINCIPAL COMPONENT ANALYSIS (PCA)

In our case, the hand pose was represented as the hand joint vector with 40 dimensions. However, as the shape of the human hand is slightly similar, we thought that the intrinsic dimensionality of the hand joint vector during typing would be lower. This was why we used PCA [38]; it allows the original vector to be decomposed without loss of information and helps a machine learning model to converge more quickly its highest accuracy [42]. To do this, the PCA extracts a set of successive orthogonal components, called principal components, which explain the maximum amount of the variance of the original data. Therefore, we tried to determine the optimal number of components along with various parameters of the machine learning model.

2) MULTI-LAYER PERCEPTRON (MLP)

MLP [29] is one of most famous machine learning algorithms, and it has the capability to learn both linear and non-linear models. The basic structure consists of an input, hidden, and output layers where each layer has a plurality of neurons, and each neuron has its weighted connections to all neurons at the next layer. Each element of an input vector, i.e., the hand joint vector in our case, is inserted into the corresponding neuron at the input layer. Each of the elements is multiplied by the weight on its connection to the next layer and then, all the multiplied values are summed up into one accumulated value. The accumulated value is eventually passed through a specific activation function to the corresponding neuron of the next layer. When an input vector reaches the output layer, the neuron values of the output layer become the predicted result.

In this work, we used a MLP with one hidden layer. To determine the optimal MLP model, we then compared the combinations of the hyper parameters: activation function = {tanh, relu, sigmoid} [43], the number of neurons in hidden layer = {all multiples of 10 between 10 and 100} [44].

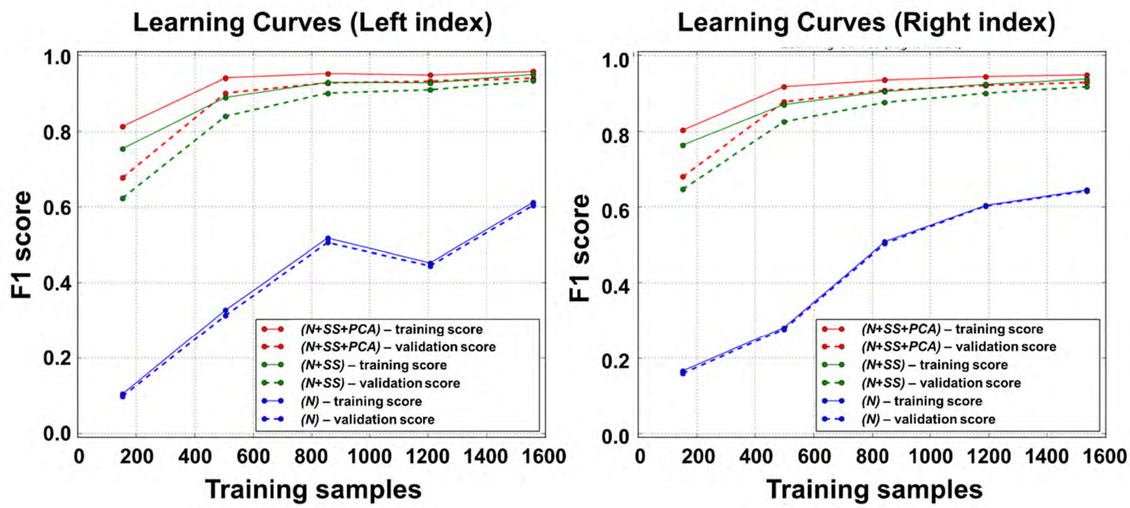

FIGURE 7. For the two index fingers, learning curves of the MLP models whenever the original hand joint vector was transformed by each stage of the Normalizer (*N*), the StandardScaler (*SS*), and the PCA.

3) RESULTS

For comparison, we measured the key inference performance for all combinations of the parameters to be considered. We used the typing data on the KP keyboard collected from session 1 to session 4. To consider recall and precision of the key inference performance, 10-fold cross-validation was carried out in terms of f1 score.

As a result, we decided that the PCA outputted 25 components and the MLP was trained with the sigmoid function as its activation function and 60 neurons in its hidden layer. For the two index fingers, whose target keys are the most difficult to infer compared to those of the other fingers, Figure 7 shows the learning curves of the MLP models whenever the original hand joint vector was transformed by each step, including the Normalizer, the StandardScaler, and the PCA. As expected, our MLPs could learn faster and converge on higher accuracies as long as the original vector went through each step. Furthermore, as the number of samples increases, the difference between the training and the validation scores decreases, indicating that the MLP was less sensitive to variance errors.

C. KEYBOARD IMPLEMENTATION

With the selected parameter set, we finally constructed a key inference model for each finger, except the thumbs. For learning, the typing data collected from session 1 to session 4 were used. Then, the data of session 5 was used to measure the performance in terms of the f1 score. To show how much our key inference model improves the virtual keyboards compared with the Choi *et al.*'s models [4], we also applied their models to our typing data and compared our models and theirs.

For all the fingers, our key inference models ($M = 97.64\%$, $SD = 1.288$) showed higher accuracies than Choi *et al.*'s models ($M = 88.21\%$, $SD = 7.586$) as shown in Table 1. Though Choi *et al.*'s models inferred target keys using the correlations between all finger movements, their design was vulnerable to unintended hand drift because they calculated the finger movements based on the initialization points. Therefore, such performance differences indicated that our models correctly inferred a target key regardless of the hand drift.

TABLE 1. Key inference accuracies of our models and Choi et al.'s models for all fingers.

Hand index	Finger index	Ours	Choi *et al.* [4]
Left	Index	96.7%	81.3%
	Middle	97.8%	91.9%
	Ring	98.6%	92.4%
	Little	96.0%	85.7%
Right	Index	95.8%	71.9%
	Middle	99.8%	95.4%
	Ring	98.0%	94.0%
	Little	98.4%	93.1%

To sum up, our hand pose aware virtual keyboard includes the high f1-scored key inference model for each finger, where each model has its own Normalizer, StandardScaler, PCA, and MLP. When a finger touch occurs on the keyboard, the touching finger's key inference model infers the target key among the keys assigned to the touching finger.

VI. TYPING PERFORMANCE EVALUATION

To evaluate the typing performance of our keyboard, we ran an additional typing experiment. Specifically, the purpose of this experiment was to determine how effective our techniques are in terms of improving typing speed, errors, and accuracy. For comparison, we implemented three different virtual keyboards including our proposed virtual keyboard.

A. THREE DIFFERENT KEYBOARD DESIGNS

For the virtual keyboards, the initialization step and backspace gesture were provided as described before. The initialization step was used to generate a keyboard layout whenever the participants wanted, such as resting all fingers on the surface or changing their initialization points. The three virtual keyboards varied according to whether each of our techniques, i.e., Key Pre-allocation and Key Inference based on Hand Poses, are applied.

First, *Normal keyboard* (or *N-key*) is the keyboard that includes none of the techniques that we proposed. On the normal keyboard, when a touch event occurs, the closest key from touch point is entered regardless of the identity of the touching finger. In other words, this keyboard does not detect which finger is used for typing, and it simply infers a target key using a touch point only. Therefore, the horizontal and vertical typing errors may occur frequently. The distance between keys is the same as 18 x 18 mm keys of QWERTY physical keyboard.

Second, *Pre-allocation keyboard* (or *P-key*) is designed based on Key Pre-allocation. When a touch event occurs, it brings the keys allocated to that touching finger and selects the one closest to the touch point. As the identity of touching finger limits the candidates of inputtable keys, the P-key performs better than the Normal keyboard. Specifically, the P-key was effective in reducing horizontal typing errors.

Lastly, *Pre-allocation and Hand Pose aware keyboard* (or *HP-key*) is designed based on both Key Pre-allocation and Key Inference based on Hand Poses. When a touch event occurs, a target key is inferred through the touching finger's key inference model. Therefore, we expected that the HP-key reduced the horizontal and vertical typing errors at the same time.

B. PARTICIPANTS & PROCEDURE

Eight of the well-adapted typists who participated in the previous experiment were re-invited. Their average typing speed on the KP keyboard was 61.85 WPM ($SD = 13.066$). Each participant was required to type on the N-key, P-key, and HP-key in random order. Because the participants have not used the three keyboards before, they entered 20 sentences per keyboard for practice. Then, we conducted a main typing experiment. During the experiment, the participants were requested to type 100 Mackenzie phrases [20] over 5 sessions on each keyboard. To measure actual typing performance, we permitted typing errors. Then, we asked the participants to type as quickly as possible while correcting all the typing errors. The experiment was conducted for three successive days for considering their tiredness; we made the participants type on only one keyboard per day. After the participants finished the three typing experiments, they were compensated with $50.

C. RESULTS

We collected 24,764, 23,010, and 22,506 typing data from the N-key, the P-key, and the HP-key, respectively. We first compared the three keyboards in terms of typing speed and several kinds of error rates. For analysis, we used two-way repeated measures ANOVA followed by the Tukey post-hoc analysis. Keyboard and session are the within subject factors. All significant findings are reported in the 95% confidence interval.

TABLE 2. **Two-way repeated measures ANOVA results. The highlighted cells represent a significant main effect ($p < 0.05$).**

Metric	*Keyboard* ($F_{2,14}$/p-value)	*Session* ($F_{4,28}$/p-value)	Interaction ($F_{8,56}$/ p-value)
WPM	**69.320 / < 0.001**	**3.597 / < 0.05**	0.423 / 0.902
TER	**35.714 / < 0.001**	2.586 / 0.059	0.507 / 0.846
CER	**35.975 / < 0.001**	2.716 / 0.050	0.549 / 0.815
NCER	1.556 / 0.245	**4.391 / < 0.05**	0.676 / 0.711

Table 2 summarizes all results of the two-way repeated measures ANOVA. As interactions between *Keyboard* and *Session* do not have a significant effect on all performance metrics, the following interpretations refer only to the main effects of *Keyboard* and *Session*.

1) TYPING SPEED

The HP-key had the fastest typing speed. The three keyboards differed significantly in terms of typing speed ($F = 69.320$, $p < 0.001$) and each keyboard performed better over five typing sessions ($F = 3.597$, $p < 0.05$). For each keyboard, Figure 8 (A) and Figure 8 (B) show the average typing speed (WPM) and the typing speed on each session, respectively. On average, the N-key had a typing speed of 33.648 WPM ($SD = 6.99$) and the P-key had 41.796 WPM ($SD = 9.45$). The HP-key ($M = 55.649$, $SD = 10.25$) was faster than both keyboards ($p < 0.05$). It is worth noting that our HP-key was approximately 24 WPM faster than the finger correlations-aware keyboard proposed by Choi *et al*. [4] ($M = 31.618$).

2) ERROR RATES

To measure the typing errors, we used four different metrics proposed by Soukoreff and Mackenzie [34]. First, the correct error rate (CER) is the ratio of errors that are subsequently fixed by users during text entry. Second, the not correct error rate (NCER) is the ratio of errors that are left in the transcribed text at the end of each phrase. Finally, the total error rate (TER) is the sum of the CER and NCER, indicating error frequencies of substitution, insertion, and omission.

As a result, the HP-key had the lowest error rates. For each keyboard, Figure 8 (C-H) shows the results of the error rates. In the case of TER, Figure 8(C) and Figure 8(D) show the mean for each keyboard and TER on each session. The three keyboards differed significantly in terms of TER ($F = 35.714$, $p < 0.001$). On an average, the TER of the N-key was 18.536 % ($SD = 4.064$), and the TER of the P-key was 13.103% ($SD = 5.055$). The HP-key ($M = 9.799\%$, $SD = 3.354$) was more accurate than both keyboards ($p < 0.05$). In the same vein, as shown in Figure 8(E) and Figure 8(F),

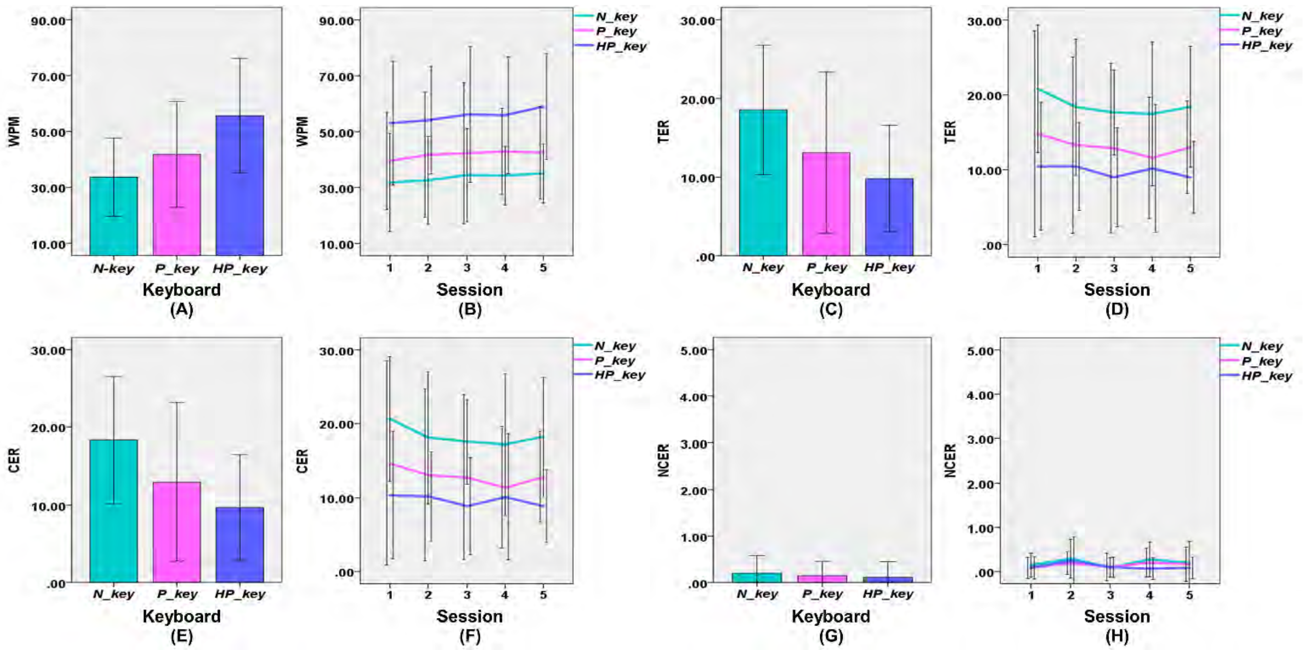

FIGURE 8. Typing performance measurements for the N-key, P-key, and HP-key. For the three keyboards, (A) an average and standard deviation of typing speed (WPM), (B) typing speed per session, (C) an average and standard deviation of TER, (D) TER per session.

the HP-key ($M = 9.688\%$, $SD = 3.377$) was statistically the most accurate in terms of CER (N-key: $M = 18.340\%$, $SD = 4.039$, P-key: $M = 12.957\%$, $SD = 5.041$).

Exceptionally, the keyboards had no difference in NCER ($F = 1.556, p < 0.001$). As the participants were requested to type as quickly as possible, occasionally they failed to correct the error in the last word of the current sentence and moved on to the next one. Nevertheless, the three keyboards showed an NCER of less than 1% (N-key: $M = 0.198\%$, $SD = 0.193$, P-key: $M = 0.147\%$, $SD = 0.154$, and HP-key: $M = 0.112\%$, $SD = 0.168$).

3) KEY INPUT ACCURACY

To know more about how our techniques improve the virtual keyboards, we checked input accuracy per key. For analysis, we used one-way repeated measures ANOVAs, followed by the Tukey post-hoc analysis. *Keyboard* is the within-subject factor.

As a result, the HP-key most accurately entered all the input keys. The three keyboards differed significantly in terms of the f1 score ($F = 29.115$, $p < 0.001$). The N-key had an average of 84.51% ($SD = 0.026$), and the P-key had the average of 88.99% ($SD = 0.049$). The HP-key had an average of 91.94% ($SD = 0.033$). The participants could most accurately type on the HP-key as compared to other keyboards ($p < 0.05$).

For the horizontal typing errors, the three keyboards had different error frequencies ($F = 61.284, p < 0.001$). The horizontal typing errors occurred 162.125 times ($SD = 49.461$) on the N-key, 45.50 times ($SD = 27.034$) on the P-key, and 27.50 times ($SD = 23.403$) on the HP-key. Although the horizontal typing errors on the HP-key occurred less than those on the N-key ($p < 0.001$), there was no difference between the HP-key and P-key ($p = 0.194$). As Key Pre-allocation was not applied to the N-key unlike the other keyboards, we concluded that our Key Pre-allocation helped reduce the horizontal typing errors.

Furthermore, the experiment showed that our Key Inference based on Hand Poses helped reduced the vertical typing errors. For the vertical typing errors, the three keyboards had different error frequencies ($F = 10.155$, $p < 0.001$). The vertical typing errors occurred 76.38 times ($SD = 18.493$) on the N-key, 63.88 times ($SD = 19.453$) on the P-key, and 38.50 times ($SD = 11.123$) on the HP-key. The vertical typing errors on the HP-key occurred less than those on other keyboards ($p < 0.05$). However, there was no difference between the N-key and P-key ($p = 0.166$).

VII. DISCUSSION

Interestingly, the HP-key (55.6 WPM) has almost reached the ideal typing speed of ten-finger typing on a large touchscreen (59.5 WPM) that was suggested in [8]. It was because several features of the HP key allowed users to type accurately without looking down at the keyboard layout or hand position. After typing on all the three keyboards, most participants commented: ''The other two keyboards require a certain level of concentration so as not to lose the initialization points, but HP-key does not require that much concentration. I could focus on the sentence I am typing''.

Figure 9 shows the distribution of all touch points by the (A) P-key and (B) HP-key, where different colors express different keys. As the P-key infers a target key by the distance between the touch point and initialization points set by users, boundaries between the keys were identified. On the other

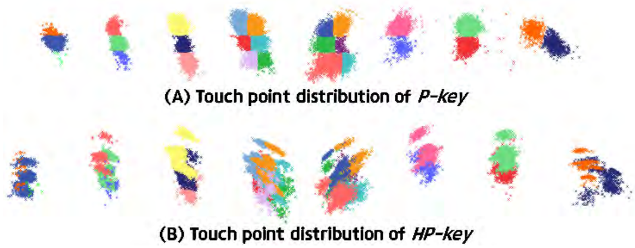

FIGURE 9. Touch point distribution of (A) the P-key and (B) HP-key. The distribution of all touch points was adjusted relative to the initialization points for the (A) P-key and wrist positions for the (B) HP-key.

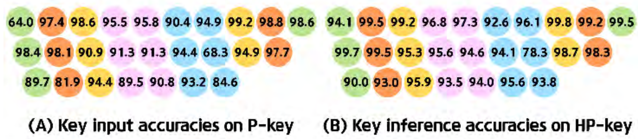

FIGURE 10. Key input accuracies (f1 score) within an assigned finger on the (A) P-key and (B) HP-key.

hand, as the HP-key determines a target key by estimating a hand pose, the inference was irrelevant to the touch point. Accordingly, the distribution of the HP-key was adjusted relative to the wrist positions. As shown in Figure 9(B), on the HP-key, because of differences in hand size between participants, the boundaries between keys were unclear.

Figure 10 shows key input accuracies (f1 score) within an assigned finger on the (A) P-key and (B) HP-key. For all input keys, the HP-key had an accuracy of 95.69% ($SD = 4.332$), and it had approximately 3.78% more accuracy than the P-key ($M = 91.91\%$, $SD = 8.508$). These results further implied that our HP-key was not affected by the difference in hand size.

However, despite the several advantages of the HP-key, the experimental environment and keyboard design of this work have concerns about feasibility and applicability. First, our experimental setup relies on a high-fidelity tracking system. Second, Key Pre-allocation may be a strong restriction for some QWERTY-familiar typists. For better usage of our keyboards, we present solutions to those problems based on recent technological developments.

A. FEASIBILITY OF HAND POSE TRACKING SYSTEM

For the feasibility of our tracking system, we note that vision-based hand pose estimation is continuously improving [35], [46]. Especially, the latest RGB-D camera-based hand pose estimation algorithm [46] supports real-time processing (~91 fps) and simultaneously accurate hand pose estimation with errors less than 10 mm. In addition, depth sensor-equipped devices, such as Google Tango, etc. [27] are also commercially available. In accordance with this trend, we believe that such advances in technologies enable our virtual keyboard to be applied to any flat surface.

Meanwhile, Virtual Reality (VR) and Mixed Reality (MR) dream of providing a huge interaction space using peripheral devices including VR data gloves. Several recent VR gloves [45], such as HaptX etc., feature accurate hand pose tracking with an error in sub-millimeters. Though, it is surprising to see that the current text entry in VR and MR has been performed either in mid-air at very low speeds ($<$ 25WPM) or by interacting with physical keyboards [47], [48], we think that VR with gloves is one practical application for our proposed keyboard, where users can type on a flat surface with the VR gloves much faster than the existing VR virtual keyboards.

B. APPLICABILITY OF HP-KEY

As the non-adapted typists may have several different finger-to-key mapping from Key Pre-allocation, their typing speed was limited on the KP keyboard. We wanted to check whether the Key Inference based on Hand Poses with individualized Key Pre-allocation was also valid for the non-adapted typists. Therefore, when individual finger-to-key mapping was applied to the virtual keyboard, we compared the accuracies between key classifications based on touch points and hand poses. To this end, we measured key classification accuracies in two ways with the typing data on the NKP keyboard of the first experiment.

First, the accuracies for all fingers were measured by classifying target keys based on touch points using a simple distance-based classification. Here, the touch points were calculated with respect to the marker positions on the wrists. Second, the accuracies were measured by classifying target keys based on hand poses. Similar to our key inference system, the hand joint vectors were classified as target keys by passing through the two stages of data preprocessing, PCA, and MLP. As individual finger-to-key mapping varied from each other, we constructed a key inference system per typist. When an alphabetic key was entered with multiple fingers, the key was considered to be assigned to the most frequently entered finger.

As a result, even the non-adapted typists seemed to be posing differently for input keys. Table 3 shows the accuracies of the two classification methods per finger. For all fingers, the classification based on hand poses ($M = 90.97\%$, $SD = 3.981$) displayed higher accuracies than the touch point based classification ($M = 82.10\%$, $SD = 6.918$). These results implied that our Key Inference based on Hand Poses is also applicable to the non-adapted typists.

Based on the above implication, we think our keyboard design can accommodate more typists by loosening the restrictions on Key Pre-allocation. Except in extreme cases, where all keys are entered with just one or two fingers in one hand, most of the mismatching problem with Key Pre-allocation occurred on the keys directly next to the assigned column, e.g., entering the C key with the left index finger and X key with the left middle finger. As the index fingers of the current HP-key showed already high accuracies ($M = 96.25\%$) despite being assigned six keys to those fingers, we believe that the HP-key can be typed with high performance even though other fingers are assigned one or two more keys.

TABLE 3. Key classification accuracies based on touch points and hand poses for the non-adapted typists.

Hand index	Finger index	Touch points	Hand poses
Left	Index	79.3%	88.1%
	Middle	89.1%	93.7%
	Ring	81.7%	89.1%
	Little	79.9%	89.1%
Right	Index	69.5%	85.8%
	Middle	82.5%	91.9%
	Ring	82.0%	91.5%
	Little	92.7%	98.7%

VIII. CONCLUSION AND FUTURE WORK

Owing to hand drift, the existing virtual keyboards are vulnerable to typing errors frequently occurring between neighboring keys in the same row or column. To reduce these typing errors on virtual keyboards, we proposed two techniques based on the typing patterns of experienced typists. First, a set of alphabetic keys in the same column are allocated to its corresponding finger. Second, as skilled typists enter the keys with different hand poses, we used the unique hand poses to infer the users' intended keys. This key inference process is tolerant of hand drift, and therefore it reduces the typing errors that occur around a target key. Based on these techniques, we finally implemented a novel hand pose-aware virtual keyboard. Our experimental studies demonstrated that the proposed keyboard design outperformed the existing virtual keyboards in terms of typing performance and conclusively reached a typing speed of 55.6 WPM.

Despite the significant improvement, we observed that the keys assigned to the index fingers still raised the most frequent typing errors in the current HP-key. It leads us to believe that the current HP-key still has room for improvement. For its further improvement in the future, it would be worthwhile to examine more joints of typists' hands to infer the target keys. A more precise hand model may contain more information for key inference. In addition, using semantic information could be another possible direction. The language modeling [10], [12] is to predict the next word or character based on the previous sequence of words. Thus, combining the HP-key with language modeling could reduce the frequent errors related to the index finger.

In addition, we need to expand the size of the experimental design. Loosening the restrictions on Key Pre-allocation allows the HP-key to be used to more typists, but the number of input data in the additional key column to each finger is significantly less than the number of data in the previously allocated column. Since the deviations between input keys for each finger adversely affect the HP-key's key inference, we will need more input data to compensate for the deviations. To do this, inviting more typists or the synthetic data augmentation based on hand kinematics model may be considered.

REFERENCES

[1] J. Barrett and H. Krueger, "Performance effects of reduced proprioceptive feedback on touch typists and casual users in a typing task," *Behav. Inf. Technol.*, vol. 13, no. 6, pp. 373–381, Mar. 1994.
[2] C. M. Bishop, *Pattern Recognition and Machine Learning*. Springer, 2006, ch. 4.3.4.
[3] B. Chaparro, B. Nguyen, M. Phan, A. Smith, and J. Teves, "Keyboard performance: Ipad versus netbook," *Usability News*, vol. 12, no. 2, pp. 1–9, Nov. 2010.
[4] D. Choi, H. Cho, and J. Cheong, "Improving virtual keyboards when all finger positions are known," in *Proc. 28th Annu. Symp. User Interface Softw. Technol.*, Nov. 2015, pp. 529–538.
[5] W. E. Cooper, *Cognitive Aspects of Skilled Typewriting*. Springer, 2012.
[6] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 20, no. 3, pp. 273–297, Sep. 1995.
[7] P. M. Domingos, "A few useful things to know about machine learning," *Commun. ACM*, vol. 55, no. 10, pp. 78–87, Oct. 2012.
[8] L. Findlater, J. O. Wobbrock, and D. Wigdor, "Typing on flat glass: Examining ten-finger expert typing patterns on touch surfaces," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, May 2011, pp. 2453–2462.
[9] L. Findlater and J. Wobbrock, "Personalized input: Improving ten-finger touchscreen typing through automatic adaptation," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, May 2012, pp. 815–824.
[10] A. Fowler, K. Partridge, C. Chelba, X. Bi, T. Ouyang, and S. Zhai, "Effects of language modeling and its personalization on touchscreen typing performance," in *Proc. 33rd Annu. Conf. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 649–658.
[11] K. Go and Y. Endo, "CATKey: Customizable and adaptable touchscreen keyboard with bubble cursor-like visual feedback," in *Proc. IFIP Conf. Hum.-Comput. Interact.* Berlin, Germany: Springer, 2007.
[12] J. Goodman, G. Venolia, K. Steury, and C. Parker, "Language modeling for soft keyboards," in *Proc. 7th Int. Conf. Intell. User Interfaces*, Jan. 2002, pp. 194–195.
[13] J. T. Grudin, "Error patterns in novice and skilled transcription typing," in *Cognitive Aspects Skilled Typewriting*. New York, NY, USA: Springer, 1983, pp. 121–143.
[14] A. Gupta and R. Balakrishnan, "DualKey: Miniature screen text entry via finger identification," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, May 2016, pp. 59–70.
[15] S. Kim, J. Son, G. Lee, H. Kim, and W. Lee, "TapBoard: Making a touch screen keyboard more touchable," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.* Apr./May 2013, pp. 553–562.
[16] S. B. Kotsiantis, D. Kanellopoulos, and P. E. Pintelas, "Data preprocessing for supervised leaning," *Int. J. Comput. Sci.*, vol. 1, no. 2, pp. 111–117, Jun. 2006.
[17] F. C. Y. Li, R. T. Guy, K. Yatani, and K. N. Truong, "The 1line keyboard: A QWERTY layout in a single line," in *Proc. 24th Annu. Symp. User Interface Softw. Technol.*, Oct. 2011, pp. 461–470.
[18] F. C. Y. Li, L. Findlater, and K. N. Truong, "Effects of hand drift while typing on touchscreens," in *Proc. Graph. Interface*, May 2013, pp. 95–98.
[19] Z. Ma, D. Edge, L. Findlater, and H. Z. Tan, "Haptic keyclick feedback improves typing speed and reduces typing errors on a flat keyboard," in *Proc. IEEE World Haptics Conf. (WHC)*, Jun. 2015, pp. 220–227.
[20] I. S. MacKenzie and R. W. Soukoreff, "Phrase sets for evaluating text entry techniques," in *Proc. Extended Abstr. Hum. Factors Comput. Syst. (CHI)*, Apr. 2003, pp. 754–755.
[21] I. S. MacKenzie. (2002). *A Note on Calculating Text Entry Speed Unpublished Work*. [Online]. Available: http://www.yorku.ca/mack/RN-TextEntrySpeed.html
[22] D. Masson, A. Goguey, S. Malacria, and G. Casiez, "Which fingers: Identifying fingers on touch surfaces and keyboards using vibration sensors," in *Proc. 30th Annu. Symp. User Interface Softw. Technol.*, Oct. 2017, pp. 41–48.
[23] C. McAdam and S. Brewster, "Distal tactile feedback for text entry on tabletop computers," in *Proc. 23rd Brit. Group Annu. Conf. People Comput., Celebrating People Technol. (HCI)*, Sep. 2009, pp. 504–511.
[24] D. F. Morrison, "Multivariate analysis of variance," *Encyclopedia Biostat.*, vol. 5, pp. 245–276, 2005.
[25] T. Murase, A. Moteki, N. Ozawa, N. Hara, T. Nakai, and K. Fujimoto, "Gesture keyboard requiring only one camera," in *Proc. 24th Annu. Symp. Adjunct User Interface Softw. Technol.*, Oct. 2011, pp. 9–10.
[26] L. E. Peterson, "K-nearest neighbor," *Scholarpedia* vol. 4, no. 2, p. 1883, 2009.

[27] H. Plank, G. Holweg, T. Herndl, and N. Druml, "High performance time-of-flight and color sensor fusion with image-guided depth super resolution," in *Proc. Conf. Design, Automat. Test Eur.*, Mar. 2016, pp. 1213–1218.

[28] H. Roeber, J. Bacus, and C. Tomasi, "Typing in thin air: The canesta projection keyboard - a new method of interaction with electronic devices," in *Proc. Extended Abstr. Hum. Factors Comput. Syst. (CHI)*, Apr. 2003, pp. 712–713.

[29] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Cognit. Model.*, vol. 323, no. 6088, pp. 533–536, 1986.

[30] T. A. Salthouse, "Effects of age and skill in typing," *J. Experim. Psychol., Gen.*, vol. 113, no. 3, pp. 345–371, 1984.

[31] T. A. Salthouse, "Anticipatory processing in transcription typing," *J. Appl. Psychol.*, vol. 70, no. 2, pp. 264–271, 1985.

[32] C. Sax, H. Lau, and E. M. Lawrence, "Liquid keyboard: An ergonomic, adaptive QWERTY keyboard for touchscreens and surfaces," in *Proc. 5th Int. Conf. Digit. Soc. Conf. (IARIA)*, 2011, pp. 117–122.

[33] A. Sears, "Improving touchscreen keyboards: Design issues and a comparison with other devices," *Interacting Comput.*, vol. 3, no. 3, pp. 253–269, Dec. 1991.

[34] R. W. Soukoreff and I. S. MacKenzie, "Metrics for text entry research: An evaluation of MSD and KSPC, and a new unified error metric," in *Proc. SIGCHI Conf. Human Factors Comput. Syst.*, Apr. 2003, pp. 113–120.

[35] T. Sharp, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, A. Vinnokv, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, and S. Izadi, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd ACM Annu. Hum. Factors Comput. Syst.*, Apr. 2015, pp. 3633–3642.

[36] M. Weiss, R. Jennings, R. Khoshabeh, J. Borchers, J. Wagner, Y. Jansen, and J. D. Hollan, "SLAP widgets: Bridging the gap between virtual and physical controls on tabletops," in *Proc. Extended Abstr. Hum. Factors Comput. Syst. (CHI)*, Apr. 2009, pp. 3229–3234.

[37] D. Wigdor, G. Penn, K. Ryall, A. Esenther, and C. Shen, "Living with a tabletop: Analysis and observations of long term office use of a multi-touch table," in *Proc. 2nd IEEE Annu. Int. Workshop Horizontal Interact. Hum.-Comput. Syst. (TABLETOP)*, Oct. 2007, pp. 60–67.

[38] S. Wold, K. Esbensen, and P. Geladi, "Principal component analysis," *Chemometrics Intell. Lab. Syst.*, vol. 2, nos. 1–3, pp. 37–52, Aug. 1987.

[39] Y. Yin, Q. Li, L. Xie, S. Yi, E. Novak, and S. Lu, "CamK: A camera-based keyboard for small mobile devices," in *Proc. 35th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM)*, Apr. 2016, pp. 1–9.

[40] W. Shi, C. Yu, X. Yi, Z. Li, and Y. Shi, "TOAST: Ten-finger eyes-free typing on touchable surfaces," *Interact., Mobile, Wearable Ubiquitous Technol.*, vol. 2, no. 1, Mar. 2018, Art. no. 33.

[41] A. M. Feit, D. Weir, and A. Oulasvirta, "How we type: Movement strategies and performance in everyday typing," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, May 2016, pp. 4262–4273.

[42] N. Halko, P. G. Martinsson, and J. A. Tropp, "Finding structure with randomness: Stochastic algorithms for constructing approximate matrix decompositions," *SIAM Rev.*, vol. 53, no. 2, pp. 217–228, 2011.

[43] B. Karlik and A. V. Olgac, "Performance analysis of various activation functions in generalized MLP architectures of neural networks," *Int. J. Artif. Intell. Expert Syst.*, vol. 1, no. 4, pp. 111–122, Feb. 2011.

[44] N. Murata, S. Yoshizawa, and S. Amari, "Network information criterion-determining the number of hidden units for an artificial neural network model," *IEEE Trans. Neural Netw.*, vol. 5, no. 6, pp. 865–872, Nov. 1994.

[45] J. Perret and E. V. Poorten, "Touching virtual reality: A review of haptic gloves," in *Proc. 16th Int. Conf. New Actuat. (ACTUATOR)*, Jun. 2018, pp. 1–5.

[46] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.

[47] P. Knierim, V. Schwind, A. M. Feit, F. Nieuwenhuizen, and N. Henze, "Physical keyboards in virtual reality: Analysis of typing performance and effects of avatar hands," in *Proc. Conf. Hum. Factors Comput. Syst. (CHI)*, Apr. 2018, p. 345.

[48] K. Wang, D. Iwai, and K. Sato, "Supporting trembling hand typing using optical see-through mixed reality," *IEEE Access*, vol. 5, pp. 10700–10708, 2017.

DAEWOONG CHOI received the B.S. and M.S. degrees in computer and information science from Korea University, South Korea, in 2016, where he is currently pursuing the Ph.D. degree in computer and information science. His research interests include text entry, ten-finger typing on virtual keyboards, human–computer interaction, and machine learning.

HYEONJOONG CHO received the Ph.D. degree in computer engineering from the Virginia Polytechnic Institute and State University (Virginia Tech), in 2006. He is currently a Professor with the Department of Computer and Information Science, Korea University, South Korea. His research interests include real-time systems on various platforms, including single-/multi-processors and sensor networks. He is also interested in solving problems based on machine learning, such as text entry, hand pose estimation, and natural language processing.

KYEONGEUN SEO received the B.S. and M.S. degrees in computer and information science from Korea University, South Korea, in 2015, where she is currently pursuing the Ph.D. degree in computer and information science. Her research interests include deep learning-based hand pose estimation, gesture-based user interface, and human–computer interaction.

SANGYUB LEE received the B.S. and M.S. degrees in electronic and electrical engineering from Yonsei University, South Korea, in 2005. He is currently pursuing the Ph.D. degree in computer and information science with Korea University, South Korea. He is also a Senior Researcher with the Korea Electronics Technology Institute. His research interests include automotive network systems and various embedded platform systems.

JAEKYU LEE received the B.S. and M.S. degrees in electronic and electrical engineering from the Chonbuk National University, South Korea, in 2012. He is currently pursuing the Ph.D. degree in computer and information science with Korea University. He is also a Researcher with the Korea Electronics Technology Institute. His research interests include automotive software and various embedded platform systems.

JAEJIN KO received the B.S., M.S., and Ph.D. degrees in computer engineering from Kwangwoon University, South Korea, in 2005 and 2013. He is currently the Head Researcher with the Korea Electronics Technology Institute. His research interests include wearable devices and various embedded platform systems.