Multimodal Technologies and Interaction

MDPI

Article

Development and Evaluation of a Low-Jitter Hand Tracking System for Improving Typing Efficiency in a Virtual Reality Workspace

Tianshu Xu [1],*, Wen Gu [2], Koichi Ota [2] and Shinobu Hasegawa [2],*

[1] Division of Advanced Science and Technology, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan
[2] Center for Innovative Distance Education and Research, Japan Advanced Institute of Science and Technology, Ishikawa 923-1292, Japan; wgu@jaist.ac.jp (W.G.); ota@jaist.ac.jp (K.O.)
* Correspondence: xutianshu@jaist.ac.jp (T.X.); hasegawa@jaist.ac.jp (S.H.)

Abstract: Virtual reality technology promises to transform immersive experiences across various applications, particularly within office environments. Despite its potential, the challenge of achieving efficient text entry in virtual reality persists. This study addresses this obstacle by introducing a novel machine learning-based solution, namely, the two-stream long short-term memory typing method, to enhance text entry performance in virtual reality. The two-stream long short-term memory method utilizes the back-of-the-hand image, employing a long short-term memory network and a Kalman filter to enhance hand position tracking accuracy and minimize jitter. Through statistical analysis of the data collected in the experiment and questionnaire results, we confirmed the effectiveness of the proposed method. In addition, we conducted an extra experiment to explore the differences in users' typing behavior between regular typing and virtual reality-based typing. This additional experiment provides valuable insights into how users adapt their typing behavior in different environments. These findings represent a significant step in advancing text entry within virtual reality, setting the stage for immersive work experiences in office environments and beyond.

Keywords: virtual reality; typing efficiency; low jitter; hand tracking

check for updates

Academic Editor: Keith Cheverst

Received: 17 September 2024
Revised: 3 November 2024
Accepted: 12 December 2024
Published: 8 January 2025

Citation: Xu, T.; Gu, W.; Ota, K.; Hasegawa, S. Development and Evaluation of a Low-Jitter Hand Tracking System for Improving Typing Efficiency in a Virtual Reality Workspace. *Multimodal Technol. Interact.* **2025**, *9*, 4. https://doi.org/10.3390/mti9010004

1. Introduction

In the post-pandemic era, the increasing acceptance of remote work and online education has become an undeniable reality. Against this backdrop, virtual reality (VR) technology is gradually gaining prominence, hailed as an innovative approach to remote work and education. Compared to conventional modes of remote work and learning, VR offers users a fresh experience of interacting with data in a visualized environment, liberating them from the physical constraints of traditional screens. This technological innovation endows remote work and education with a more appealing and immersive quality. For instance, Hodgson et al. [1] discuss how immersive VR is being integrated into higher education environments to enhance student engagement. Christopoulos et al. [2] explore the benefits of virtual interactions in education to increase student motivation. Tunk and Kumar [3] further highlight the potential of VR to redefine "work from home" by making remote work more engaging and collaborative.

However, realizing the full potential of these trends faces a significant obstacle—the lack of robust text input functionality in VR. Bowman et al. [4] compared various text input methods in VR and identified several challenges. Meanwhile, Grubert et al. [5] examined the usability of physical keyboards in VR and emphasized the need for more precise hand tracking. Additionally, another study by Grubert et al. [6] investigated how different hand representations influence typing accuracy, which underlines the necessity of optimizing VR text input systems to match real-world typing efficiency.

Although existing solutions, such as wearable devices, controllers, and motion sensors, provide text input support, they are often inconvenient and incur additional costs. For instance, Boletsis and Kongsvik [7] propose a VR keyboard solution using a drum-like design, while Otte et al. [8] explore text input using a touch-sensitive physical keyboard, and Meier et al. [9] introduce the TapID wristband for text input based on finger taps. Recently, machine learning methods have gained attention, with some solutions using additional cameras to capture users' hand movements and display them in real time in VR. For example, Hwang et al. [10] developed a 3D pose estimation approach using a monocular fisheye camera, while Wu et al. [11] proposed a wrist-mounted camera for estimating finger positions. Although these approaches are innovative, they require extra equipment, which adds complexity and cost to the user experience.

Furthermore, many existing solutions do not support using a physical keyboard, which can disrupt users' typing habits and cause inconvenience. Studies like those by Fourrier et al. [12] and Kim et al. [13] analyze gesture-based VR typing systems, highlighting the limitations of using virtual keyboards that deviate from traditional physical keyboard experiences. To preserve a familiar and comfortable experience, our research focuses on using a physical keyboard with 3D tactile feedback, aligning with the principle of "easy adaptation for users with keyboard input experience".

Therefore, we propose a solution that utilizes the built-in cameras of HMDs to capture users' typing actions on a physical keyboard. This approach avoids inconvenience and additional costs while respecting users' typing habits. However, using the built-in cameras of HMDs presents unique challenges. From the perspective of the HMDs' cameras, the hand's fingers are difficult to capture accurately due to the palm obstructing the view, making it challenging to obtain a complete hand outline and precise finger positions.

Another significant barrier to effective text input in VR is the presence of jitter. Jitter refers to image rendering issues that cause virtual hands to shake, resulting in inconsistency between the movements of virtual hands and the responses in the virtual environment. This inconsistency further prevents users from interacting correctly with the virtual keyboard, causing severe typing errors. Stauffert et al. [14] emphasize that even small amounts of jitter can negatively impact VR performance, particularly in tasks requiring precision.

This study aims to address the text entry challenge in VR, especially within immersive office experiences. As mentioned above, we consider the following research questions:

1. How can the built-in cameras of HMDs accurately detect typing actions, even when the line of sight is obstructed, considering the unique challenges posed by these cameras?
2. How can jitter be reduced to enhance the accuracy of virtual hand movements in VR, thereby improving the user experience?
3. How does using a physical keyboard with 3D tactile feedback impact users' typing efficiency and habits in VR?

To address these research questions, we propose utilizing the back of the hand image. By extracting information from the back of the hand image, we can accurately predict the finger's position even when it is obstructed. To achieve this, we first establish a database of back of the hand images. Subsequently, we input the back of the hand images and corresponding motion history images (MHI) into a two-stream long short-term memory (LSTM) network. This network processes the information and applies Kalman filtering (KF) to reduce jitter, thereby enhancing the precision of hand position tracking. The reason for using LSTM instead of other models is based on Section 4, which evaluates multiple models using key evaluation criteria such as latency, accuracy, and jitter. Advanced models such as TSSequencer [15], PatchTST [16], BNNActionNet [17], and LSTM are applied to comprehensively compare each model's suitability for VR typing tasks. Based on this thorough evaluation, including factors such as latency, accuracy, jitter, and ease of deployment on head-mounted displays (HMDs), we ultimately decided to adopt 2S-LSTM-KF as the optimal model for our VR typing system.

Subsequently, comparative experiments were conducted, and statistical analyses were performed on typing and questionnaire results, confirming the efficacy of our proposed method in maintaining typing efficiency. Finally, through additional experiments, we analyzed changes in users' typing habits between regular and virtual reality typing, confirming that our method minimally impacts users' typing habits.

Following this introduction, Section 2 reviews related work on VR typing systems and text entry challenges, providing a background for our approach. Section 3 explains the proposed method, detailing the 2S-LSTM network and KF techniques for hand tracking and jitter reduction. Section 4 presents the performance comparison, which evaluates our model alongside other state-of-the-art models. Section 5 describes the experimental studies, including a typing efficiency comparison and a typing behavior analysis examining how different VR typing solutions affect finger usage and typing habits, providing insights into typing performance and user interaction in VR. Finally, Section 6 concludes the study and suggests directions for future research.

2. Related Work

2.1. Typing in VR

Recent studies have explored various methods for text input in VR environments. For instance, Boletsis and Kongsvik [7] investigated a VR typing interface using Leap Motion to track users' hand movements on a circular virtual keyboard with 26 keys arranged in concentric rings. This design aims to simplify VR typing, although the interface may not replicate the familiarity of a physical keyboard. Another study by Otte et al. [8] compared different typing methods, including standard physical keyboards, touch-sensitive keyboards, and virtual keyboards with mid-air gestures, revealing key insights into how physical feedback influences typing efficiency. Additionally, Fourrier et al. [12] examined handwriting input in VR with an optical motion capture system, where a haptic glove provided tactile feedback to simulate handwriting, highlighting an alternative to keyboard-based input.

Motion sensors and cameras present another approach, such as Meier et al.'s TapID wristband [9], which detects bone vibrations from finger taps to facilitate VR typing without a traditional keyboard. This technique illustrates how sensor-based wristbands can streamline VR text input, though they still require additional wearable devices. Gesture-based systems have also gained traction; for example, Kim et al. [13] and Gil et al. [18] developed STAR and Thumb Air, allowing users to perform virtual key presses or simulate smartphone typing through hand gestures.

However, these solutions require additional devices, which can inconvenience typing (due to cumbersome device-wearing) and increase costs. Similarly, a physical keyboard is more user-friendly, as users are more familiar with physical keyboards than specially designed virtual ones. We naturally shifted our focus to machine learning approaches to circumvent the use of specially designed and potentially costly devices.

2.2. Hand Tracking

Hand tracking, a technology that facilitates the detection and monitoring of a user's hands' position, depth, speed, and orientation, utilizes various methods such as LiDAR arrays or external sensor stations. These tracking data undergo analysis and processing, generating a virtual, real-time representation of the user's hands and movements within the virtual environment. This representation is then transmitted to the relevant application or video game, enabling users to interact organically with the virtual environment using their hands.

Regrettably, wearable hand tracking solutions like lidar arrays or external sensor stations often impede typing efficiency due to the necessity of wearing additional devices. In contrast, deep learning solutions offer cost advantages as they can rely solely on the camera, eliminating the need for extra special hardware. This also means that the deep learning solution impacts typing efficiency less because it does not require wearing extra devices. For example, Zhang et al. [19] proposed a hand-tracking solution using only a standard camera, reducing the need for external hardware. Similarly, Johnson and Everingham [20] introduced an efficient clustered pose model for human pose estimation, and Mueller et al. [21] demonstrated a GAN-based approach that estimates 3D hand positions in real time using RGB cameras. These studies highlight the potential of camera-based solutions for VR typing without additional wearable devices. We plan to utilize the cameras on HMDs to capture the movements of the typing hand. However, capturing typing movements using the cameras on HMDs presents unique challenges. From the perspective of HMDs, typing fingers are often obscured by the back of the hand. This makes it difficult for the HMDs' cameras to capture a complete view of the typing hand. Consequently, accurately tracking the position of the typing hand becomes challenging.

One study proposes a methodology for estimating 3D human pose using a monocular fisheye camera mounted on a VR headset [10]. Another study has been conducted to estimate finger positions during typing by utilizing subtle variations on the back of the hand, using a wrist-mounted camera [11]. Additionally, a study presents a metaphoric gesture interface tailored for manipulating virtual objects, offering an egocentric viewpoint [22]. Inspired by their work, our approach focuses on visual features on the back of the hand, extending it to support richer, total typing hand position estimation.

2.3. Jitter in VR Systems

In VR systems, jitter, characterized by subtle signal fluctuations, is a crucial factor influencing motor performance and user experience. Despite continuous technological advancements, effectively mitigating or eliminating jitter remains challenging, especially in tracking systems integrated into various HMDs. Numerous researchers have extensively studied the impact of jitter on VR systems. An analysis conducted in one study indicated that even minor spatial jitter (0.3 mm) in input devices significantly reduces user performance [23]. Moreover, more pronounced jitter levels exhibit a more noticeable negative impact on user performance, particularly when dealing with smaller targets [24]. Another observation in a separate study revealed that as jitter levels increase, users experience a significant decline in performance metrics such as time, error rate, and throughput [25]. Additionally, a recent experiment introduced artificial jitters of 0.5°, 1°, and 1.5° in a VR system, resulting in a substantial increase in error rates with each incremental level of jitter [26].

In summary, considering the detrimental effects of jitter on user performance in virtual reality systems highlighted by the above studies, we firmly believe that an efficient VR Typing system must possess low jitter characteristics. Given the diverse and complex causes of jitter, various research directions propose methods to reduce jitter, with the use of filters catching our attention. We have incorporated Kalman filtering into the proposed network architecture to reduce jitter.

3. Proposed and Method

3.1. Data Collection Experiment

As discussed, capturing the typing hand's position using HMD cameras presents unique challenges because the fingers are frequently obscured by the palm, limiting the camera's ability to capture a complete hand profile. Existing hand image databases, such as those developed by Wang et al. [27] and Afifi [28], predominantly contain fully visible hand images, which are insufficient for our needs. Additionally, Qian et al. [29] and Roth et al. [30] created datasets focused on hand segmentation and user authentication, respectively, but these do not account for the occlusion that frequently occurs in VR typing tasks.

Given this gap, we identified the need for an "obscured typing hand" dataset specifically tailored to VR typing, where subtle and precise finger movements are critical for accurate tracking, as shown in Figure 1. Training a model on non-targeted datasets that lack occlusion features would limit its reliability in real-world VR applications. Consequently, we conducted an independent data collection process to capture images of typing hands from multiple angles with varying levels of finger occlusion, creating a specialized dataset that accurately reflects the challenges faced in VR typing scenarios.

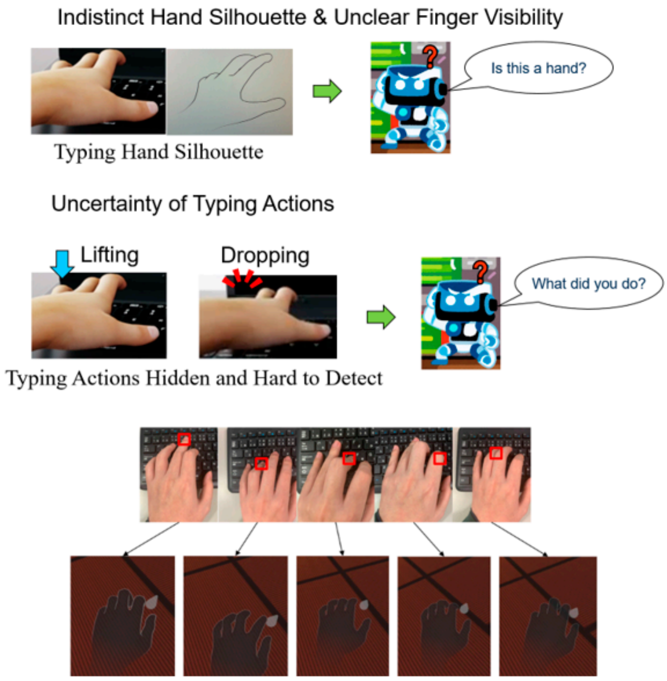

Indistinct Hand Silhouette & Unclear Finger Visibility
Is this a hand?
Typing Hand Silhouette
Uncertainty of Typing Actions
Lifting
Dropping
What did you do?
Typing Actions Hidden and Hard to Detect

Figure 1. Challenges in typing hand tracking: obscured fingers and subtle (or delicate) movements.

As shown in Figure 1, the typing actions are very subtle, making them challenging to detect. It is also difficult to predict the position of the typing hand through subtle changes in the contour. In the examples shown in the lower part of the figure, it is evident that even with different typing positions, there is no significant difference in the position of the VR hands.

The comparison shows the difference between "obscured typing hand" and other complete hand images in Figure 2. The left one is collected from the perspective of HMDs, and the right one is from the KBH dataset [27]. The bottom one in the middle is the MSU dataset [30].

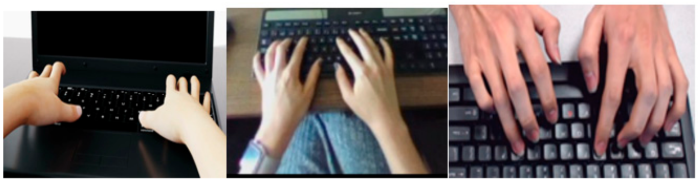


Figure 2. The difference between "obscured typing hand" and other complete hand.

3.2. Participants in Data Collection Experiment

A total of eleven students from our graduate university participated in the data collection phase, including four females aged between 25 and 31 (seven males and four females in total, with an average age of M = 28), and they all possessed fluent typing skills. The participants were instructed to use a wearable camera while typing on a computer. The 4K high-definition camera worn on the ear captured images of the "obscured typing hands", as shown in Figure 3.

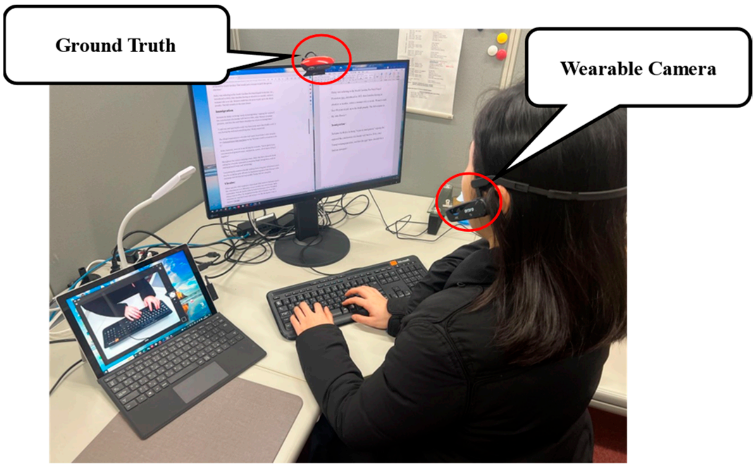
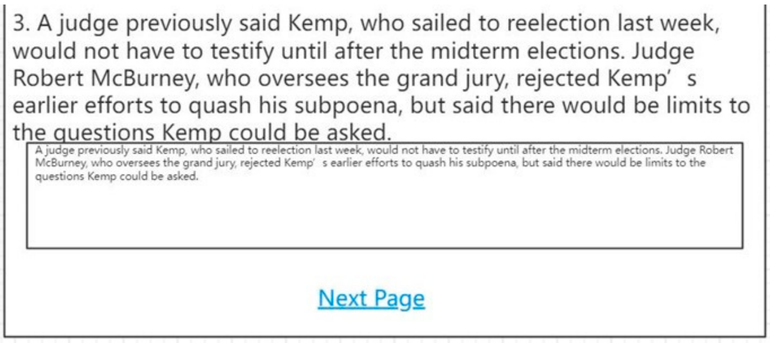


Figure 3. Typing scene with a wearing camera.

We downloaded CNN news from CNN/Daily Mail (https://github.com/abisee/cnn-dailymail, accessed on 17 December 2024) and split the news into sentences of varying lengths. Participants were required to input paragraphs of varying lengths using the QWERTY keyboard based on prompts. The UI is shown in Figure 4. We developed a small program to monitor the participants' keypress states, recording the time of keypress events. After the experiment, participants uploaded video footage from a wearable camera. We automatically extracted images before and after each keypress event using the recorded keypress times. This approach helps avoid entering invalid content into the database, such as distraction, rest, or contemplation moments.



3. A judge previously said Kemp, who sailed to reelection last week, would not have to testify until after the midterm elections. Judge Robert McBurney, who oversees the grand jury, rejected Kemp' s earlier efforts to quash his subpoena, but said there would be limits to the questions Kemp could be asked.

A judge previously said Kemp, who sailed to reelection last week, would not have to testify until after the midterm elections. Judge Robert McBurney, who oversees the grand jury, rejected Kemp' s earlier efforts to quash his subpoena, but said there would be limits to the questions Kemp could be asked.

Figure 4. The UI that the participants used for typing.

Additionally, to ensure that each key on the keyboard has a minimum number of keystrokes, we manually selected certain sentences to control the occurrence frequency of specific letters.

In the experiment, each participant engaged in a one-hour typing session, resulting in a total of 21,900 images collected. Subsequently, following the steps outlined in related research [11], we employed OpenCV to apply image processing techniques for data augmentation. Specifically, we adjusted the hand color and brightness of these images to create variations. By employing the HSV model, we randomly varied the values of H (Hue) and V (Value brightness). Consequently, we generated a dataset comprising 438,000 images, approximately 20 times larger than the original dataset.

After the experiment, human annotators manually annotated the bounding boxes using Media Pipe [19] as an assistive tool. Through this process, we extracted applicable portions of images featuring the typing hand from the wide-angle wearable camera, as shown in Figure 5.

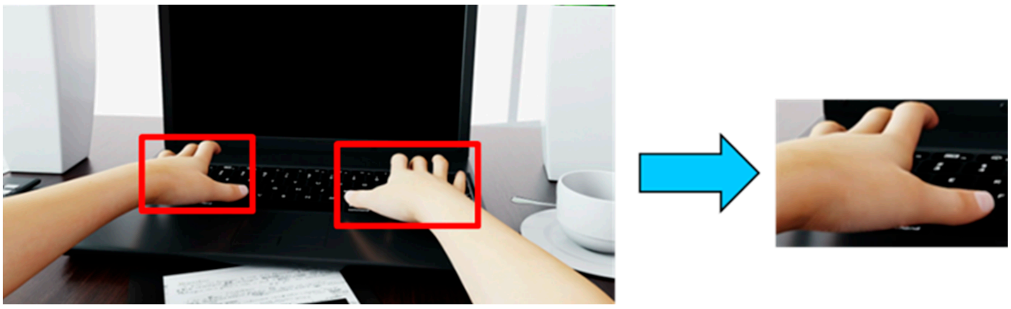


Figure 5. Bounding box annotation and cropping of the data were conducted.

3.3. Motion History Image

Motion history image (MHI) is a valuable concept in computer vision, specifically designed for capturing and representing temporal information in video sequences [31]. It plays a crucial role in motion analysis, allowing for extracting meaningful patterns related to object movements over time. MHI is a chronological representation of motion in a sequence of images, emphasizing the recency of pixel changes. It assigns higher pixel values to regions where motion has occurred more recently, creating a visual representation of the temporal evolution of movement within a video. The formula is as follows [31]:

$$H_\tau\left(x, \quad y, \quad t\right) = \begin{cases} \tau & if\ \Psi\ (x,y,t) = 1 \\ \max(0,\ H_\tau(x,y,t-1) - \delta) & otherwise \end{cases}. \tag{1}$$

In the formula, $(x,\ y)$ and t represent the pixel's position and time, respectively; τ represents the duration, determining the temporal scope of the motion from the frame perspective; δ is the decay parameter; and $\Psi\ (x,\ y,\ t)$ is the updating function, which can be defined by frame difference:

$$\psi(x,y,t) = \begin{cases} 1 & if\ D(x,y,t) \geq \xi \\ 0 & otherwise \end{cases}, \tag{2}$$

where

$$D(x,y,t) = |I(x,y,t) - I(x,y,t \pm \Delta)|. \tag{3}$$

Here, $I(x,y,t)$ is the intensity value of the pixel at coordinates (x,y) in the video image sequence at frame t, delta is the frame interval, and ξ is a manually set difference threshold adjusted with changes in the video scene.

Building upon this foundation, a more advanced approach involves using optical flow to define $\psi(x,y,t)$ [32]:

$$E(x,y,t) = s\,(x,\ y,\ t) + E\,(x,\ y,\ t-1)\cdot\alpha, \tag{4}$$

where $s\,(x,\ y,\ t)$ denotes the optical flow length corresponding to pixel $(x,\ y)$ at time frame t. The data processing is shown in Figure 6.

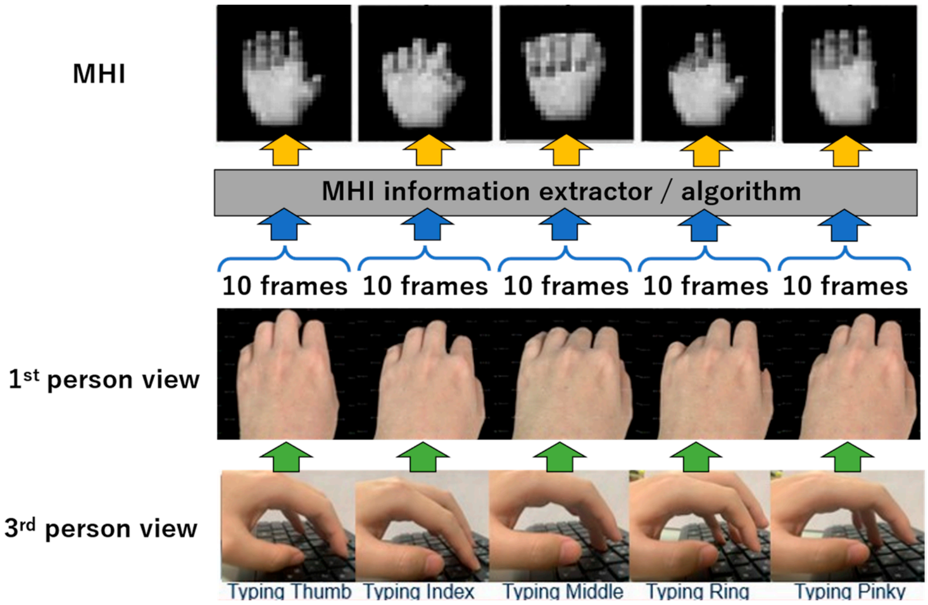


Figure 6. From normal image to MHI.

3.4. Network Architecture

This chapter explains the network architecture and the purpose of each component. Figure 7 illustrates a network that predicts the hand posture for typing using features extracted from back of hand images.

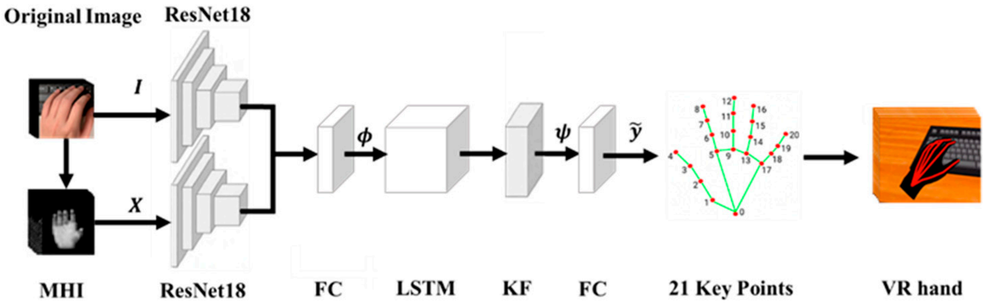


Figure 7. Overview of 2S-LSTM network.

3.4.1. ResNet 18

ResNet is a convolutional neural network (CNN) architecture that builds upon the foundation laid by VGG while introducing innovative residual connection structures [33]. As a variant of ResNet, ResNet 18 stands out for its smaller size compared to its counterparts. ResNet 18 is particularly well-suited for deployment in environments with resource constraints, such as HMDs. The advantages of ResNet18 include its relatively compact architecture while retaining the benefits of the residual connections. This smaller depth ensures that deploying ResNet18 on HMDs does not introduce significant latency, making it an optimal choice for real time applications.

In this research, the training sequence of length τ is 10. For each τ, we use the hand position labels $y_{1:\,\tau}$ and two input streams: original image $I_{1:\,\tau}$ and MHI $X_{1:\,\tau}$, are separately processed through a ResNet18 network to extract visual features. Subsequently, a fully connected layer is used to combine two visual features into a unified visual feature ϕ. Following this, the visual feature sequence $\phi_{1:\,\tau}$ is fed into an LSTM layer to extract the temporal feature sequence $\psi_{1:\,\tau}$.

3.4.2. LSTM

Long short-term memory (LSTM) is a specialized recurrent neural network (RNN) architecture designed to address challenges in capturing long-term dependencies within sequential data [34]. Unlike traditional RNNs, LSTM introduces a memory cell equipped with gating mechanisms, allowing it to selectively store, forget, and update information over extended sequences. This design overcomes issues like vanishing and exploding gradients, making LSTM particularly effective for sequential data analysis tasks. With advantages such as maintaining context over extended periods and selective information retention, LSTM has become a cornerstone in diverse applications, including natural language processing and time series prediction. The architecture's key features include memory cells, gating mechanisms, and hidden states, governed by mathematical formulations involving input gates, forget gates, cell states, output gates, and hidden states. These equations, characterized by weight matrices, biases, and activation functions, enable LSTM to excel in capturing intricate temporal patterns, making it a pivotal technology in the realm of deep learning. Given the distinctive characteristics of long short-term memory (LSTM), we employ LSTM to establish a connection with the two-stream ResNet18, aiming to extract temporal feature sequence $\psi_{1:\,\tau}$.

3.4.3. Kalman Filter

Kalman filtering (KF) is a recursive algorithm designed to estimate the state of a system [35]. This filtering method deals with dynamic systems characterized by uncertainties and measurement noise. One of its notable advantages is the ability to provide accurate estimates of the system state by fusing information from both the system model and actual measurements. Kalman filtering is a mathematical technique that can estimate the state of a dynamic system from noisy measurements. Kalman filtering has two steps: prediction and update. In the prediction step, the filter uses a motion model to predict the next state based on the previous state and the control input. In the update step, the filter uses a measurement model to correct the prediction based on the observation and the measurement noise.

The combination of LSTM and Kalman filtering [36] can be used for position regularization and state estimation. LSTM-KF integration capitalizes on the strengths of Kalman filtering, which excels in handling uncertainties and noise, and LSTM, renowned for capturing temporal dependencies in sequential data. In conclusion, the combination of LSTM and Kalman filtering holds significant potential to reduce jitter in virtual reality systems. Given that typing behavior is a continuous and linear process, the introduction of Kalman filtering is expected not only to minimize jitter but also to enhance the accuracy of recognizing the position of the typing hands.

The KF stabilizes the sequence of features extracted by the network, enhancing the accuracy and robustness of hand position estimation, especially in the presence of occlusions and complex backgrounds. Then, the output is passed through another fully connected layer. This step maps the temporal feature to the estimated position of the typing hands $\widetilde{y}_{1:\,T} = f(I_{1:\,\tau}, X_{1:\,\tau})$.

3.4.4. Key Point

We referred to the design of BlazePalm [19], each hand position label includes 42 key points (21 key points in one hand). Figures 7 and 8 show the key points.

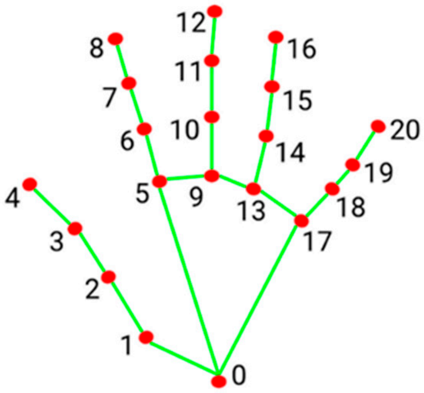


Figure 8. 21 key points for one hand.

To visualize $\widetilde{y}_{1:\tau}$, we implemented a hand simulator using Unity3D. This simulator can map $\widetilde{y}$ to a both-hand model consisting of 42 key points. By associating these key points with $\widetilde{y}$, we can dynamically reproduce and simulate the movements and positions of typing hands in real time.

4. Performance Comparison

To identify the optimal network framework for VR typing tasks, we preliminary compared multiple models, focusing on latency, accuracy, and jitter. This comparison aimed to determine which model provides the good overall performance.

4.1. Participants and Equipment

Latency, accuracy, and jitter are influenced primarily by the performance of hardware and algorithms. Therefore, we standardized the hardware across all conditions, using the HTC VIVE Pro paired with our developed VR typing interface. The only variable across conditions was the hand-tracking model employed. To gain insights into real-world user experience, we recruited three participants (two males and one female) with normal vision for the comparison.

Latency data were captured using VRScore [37], a widely-used VR performance assessment tool. Accuracy was evaluated using the test set from the internal dataset described in Section 3. Jitter was quantified by comparing the positions of real and virtual hands.

4.2. Comparison Conditions

We tested the following models in the VR typing environment, where 2S denotes a 2-stream network architecture, and KF represents Kalman filtering:

Condition 1: HTC VIVE Pro built-in gesture detection;
Condition 2: TSSequencer [15];
Condition 3: 2S-TSSequencer;
Condition 4: 2S-TSSequencer-KF;
Condition 5: PatchTST [16];
Condition 6: 2S-PatchTST;
Condition 7: BNNActionNet [17];
Condition 8: 2S-BNNActionNet;
Condition 9: 2S-BNNActionNet-KF;
Condition 10: LSTM [34];

Condition 11: 2S-LSTM;

Condition 12: 2S-LSTM-KF (Ours).

Each condition differed only in the model used, with all other factors, such as refresh rate, kept consistent to ensure that performance differences were attributed solely to the models.

4.3. Metrics and Data Collection

For an optimal VR typing experience, latency, accuracy, and jitter are all crucial evaluation metrics. We chose to collect data on all three metrics across the conditions to make a comprehensive assessment and identify the best-performing model.

4.3.1. Latency

Latency is an important evaluation metric, as high latency can induce motion sickness in users [38]. While an ideal latency is below 20 ms [39], most VR systems struggle to maintain stability within this range due to various factors like graphical rendering, signal transmission, and computational load. Individual sensitivity to latency varies, with some users perceiving delays as short as 3–4 ms [40]. We recorded the minimum, maximum, and average latency over 10 min intervals for each model. Participants provided feedback on their perceived latency and were allowed to switch between conditions for better comparison.

4.3.2. Accuracy

The accuracy for each model was measured using the test set from our internal dataset after training with the training set. This allowed us to evaluate each model's effectiveness in accurately recognizing hand movements during the typing task.

4.3.3. Jitter

Jitter was evaluated as the stability of hand positions by measuring discrepancies between real and virtual hand positions at 21 × 2 key points. Points with a discrepancy exceeding a threshold were counted as contributing to jitter, while points below this threshold were not. The threshold value was established based on criteria published in our prior work at TENCON2023 [41].

4.4. Result and Discussion

4.4.1. Result of Latency, Accuracy, and Jitter

The latency measurements for different conditions are summarized in Table 1 below. The table presents the minimum, maximum, and average latency values derived from the total 10 min of latency data collected for each condition.

Concerning latency, as shown in Table 1, most models demonstrated acceptable latency compared to the baseline (HTC VIVE Pro built-in gesture detection), with only PatchTST and 2S-PatchTST showing significantly higher latency. This increased latency may lead to user discomfort, such as dizziness, making these models less suitable for VR typing tasks. Participants reported feeling very uncomfortable and restless after using PatchTST and 2S-PatchTST for a period of time, which differed from their experiences in other conditions.

Regarding accuracy, Table 1 indicates that all models, except LSTM, achieved respectable accuracy. Excluding the high-latency PatchTST and 2S-PatchTST, the highest accuracy was observed with 2S-BNNActionNet-KF, which outperformed our proposed model by 2.27%. However, this difference was not substantial enough to noticeably affect typing performance, as participant feedback confirmed that users could not perceive a clear difference in accuracy among the top-performing models.

Table 1. The result of latency, accuracy, and jitter.

Condition	Latency (10 min)			Accuracy (%)	Jitter (Number of Point)
	Min.	Max.	Avg.		
HTC VIVE Pro built-in gesture detection	39 ms	73 ms	48 ms	65.05%	3565
TSSequencer [15]	41 ms	83 ms	62 ms	80.25%	2687
2S-TSSequencer	45 ms	107 ms	61 ms	78.80%	2606
2S-TSSequencer-KF	45 ms	111 ms	62 ms	80.45%	2049
PatchTST [16]	117 ms	250 ms	201 ms	83.73%	2389
2S-PatchTST	151 ms	297 ms	274 ms	83.19%	2710
BNNActionNet [17]	29 ms	75 ms	51 ms	77.47%	2194
2S-BNNActionNet	29 ms	91 ms	57 ms	80.81%	2124
2S-BNNActionNet-KF	30 ms	105 ms	61 ms	81.15%	1989
LSTM [35]	41 ms	77 ms	52 ms	69.75%	3134
2S-LSTM	44 ms	99 ms	58 ms	77.00%	3111
Ours	44 ms	112 ms	59 ms	78.88%	1974

In terms of jitter, as shown in Table 1, comparing 2S-TSSequencer with 2S-TSSequencer-KF, 2S-BNNActionNet with 2S-BNNActionNet-KF, and 2S-LSTM with 2S-LSTM-KF (Ours), it is evident that the models with KF exhibit smaller jitter values compared to their non-KF counterparts. Additionally, participants reported being generally satisfied with the jitter performance of the conditions which have KF.

4.4.2. Discussion on Performance Comparison

The latency results showed that while the vast majority of models (except for PatchTST and 2S-PatchTST) exhibited slightly higher latency than the baseline condition (Condition 1), this increase of a few milliseconds to over ten milliseconds remained within an acceptable range. Participant feedback indicated that the slight increase in latency brought by these models was imperceptible compared to Condition 1. Consequently, due to excessive latency, both PatchTST and 2S-PatchTST can be excluded from consideration, and we believe that the computational heaviness of PatchTST, which is based on the Transformer architecture, is a key factor contributing to its significant latency issues.

In terms of accuracy, 2S-BNNActionNet-KF emerged as the top performer, while the 2S-LSTM-KF model trailed by 2.27%. The 2S-TSSequencer-KF also performed admirably, leading 2S-LSTM-KF by just 1.57%. Given the nature of typing actions, which involve subtle movements and rapid finger lifts, the task of identifying typing fingers may not necessitate complex long-range dependency modeling, thereby limiting the advantages of the TSSequencer. Furthermore, the TSSequencer model might require larger and higher-quality datasets to fully realize its strengths. However, the dataset used in this study was self-made under limited conditions and funding, potentially constraining the performance of the TSSequencer. The results show that 2S-BNNActionNet-KF is a promising solution, especially in terms of accuracy. However, LSTM performed slightly better in terms of latency and jitter. Some previous research reported that BNNActionNet has the advantage with lower computing resources, but that LSTM achieves higher accuracy, especially in applications that require capturing subtle temporal variations [42]. As the computing resources of new HMDs improve in the future, these results may change.

Jitter analysis showed that 2S-LSTM-KF performed the best, followed by 2S-BNNActionNet-KF and 2S-TSSequencer-KF, which also demonstrated solid results. When comparing models with and without Kalman filtering (KF), the KF-enhanced versions consistently showed improved jitter performance. This suggests that incorporating KF benefits jitter reduction not only in 2S-LSTM-KF but across other models as well.

After considering latency, accuracy, and jitter performance, we believe that both 2S-BNNActionNet-KF and 2S-LSTM-KF are optimal choices. Given that 2S-BNNActionNet does not significantly outperform 2S-LSTM-KF across all metrics and considering the author's extensive experience in deploying LSTM on VR devices, we have decided to use 2S-LSTM-KF for this experiment. In our future work, we will further explore and investigate the potential applications of 2S-BNNActionNet.

5. Experiment

All experiments conducted in this study received approval from the JAIST Life Sciences Committee (H04-032).

5.1. Typing Experiment

A comparative experiment assessed the developed assistance solution (2S-LSTM) compared to two existing solutions: Oculus Quest 2 and Leap Motion. The primary objective was to validate the effectiveness of the proposed method in enhancing typing efficiency.

5.1.1. Participants

A total of 24 participants were recruited, comprising 23 right-handed individuals and 1 left-handed individual (16 males and 8 females, with an average age of M = 26), all with normal or corrected-to-normal vision. Among the participants, seven had prior VR experience. We balanced the six participant groups by gender and experimental order. All participants demonstrated a certain level of English proficiency, with some having English as their native language and using it for daily conversations. The remaining participants' English proficiency ranged from TOEIC scores of 500 to 900. Advanced touch-typing skills were not required for participation.

5.1.2. Equipment of Typing Experiment

The experiment was conducted on a desktop PC with an NVIDIA GeForce GTX 1080 Ti graphics card. The 2S-LSTM network was applied using an HTC VIVE Pro Eye headset, while Oculus Quest 2 and Leap Motion served as baseline solutions. The VR environment and other VR models utilized in the experiment were developed using Unity3D. Various USB cameras were employed to record experimental data from the participants.

5.1.3. Experimental Conditions

- Regular Typing: Participants initially completed typing tasks without wearing the HMDs for 30 min. This condition served as a baseline to assess participants' regular typing ability.
- HMDs Typing: Participants wore the HMDs and performed typing tasks using three distinct typing assistance solutions—Oculus Quest 2, Leap Motion, and the developed 2S-LSTM solution. Each task was conducted for 30 min. The order of the solutions was counterbalanced among participants to mitigate potential order effects.

5.1.4. Experiment Procedure

- Pre-Experiment Session: Participants underwent a brief training session to acquaint themselves with the HMDs and the typing assistance solutions. This session ensured participants' comprehension of task requirements and their ability to perform typing tasks comfortably.
- Typing Tasks: The above regular and HMDs Typing were performed as the Typing Task.
- Breaks and Comfort: Participants had the flexibility to take breaks at any point during the experiment to ensure their comfort and prevent symptoms such as "VR sickness".

- Typing Hands Position: The experimental setup involved recording participants' typing actions using a combination of a USB camera and a virtual camera within real and VR environments. These cameras captured the real hand position and the virtual hand positions when participants pressed keys on the keyboard. The dataset for each typing session was created by combining these recordings. High hand tracking accuracy and minimal jitter were expected to resemble typing postures of real and virtual hands. The comparison of typing postures assessed the level of fidelity and jitter in replicating hand movements in the virtual environment.

As shown in Figure 9, experiment order for each group A, B, and C are standing for Oculus Quest 2, Leap Motion, and the developed 2S-LSTM solution.

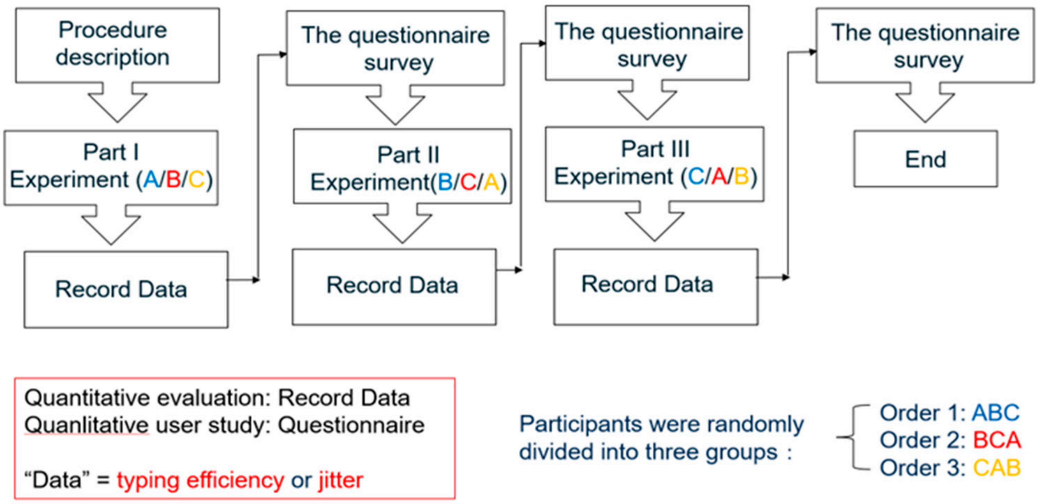

Procedure description
Part I Experiment (A/B/C)
Record Data
The questionnaire survey
Part II Experiment(B/C/A)
Record Data
The questionnaire survey
Part III Experiment (C/A/B)
Record Data
The questionnaire survey
End
Quantitative evaluation: Record Data
Quanlitative user study: Questionnaire
"Data" = typing efficiency or jitter
Participants were randomly divided into three groups :
Order 1: ABC
Order 2: BCA
Order 3: CAB

Figure 9. Process of experiment.

5.1.5. Data Collection

During typing tasks, the following data were collected:

- Total number of words (NoW) entered (including errors) in normal, Oculus Quest 2, Leap Motion, and 2S-LSTM conditions. The quantity of NoW (Number of Words) within a unit of time can also measure typing speed and fluency.
- Number of errors (E) in normal, Oculus Quest 2, Leap Motion, and 2S-LSTM conditions.
- Error rate (ER) in normal, Oculus Quest 2, Leap Motion, and 2S-LSTM conditions.
- Difference (Diff.) of hand positions in HMD typing conditions. The difference between real and virtual hand positions was quantified at 21 * 2 key points of the hand, and the differences were summed for 100 inputs.

To further analyze and evaluate our proposal, we conducted an ablation study and questionnaire survey among the participants. Detailed information and results were presented at TENCON2023 [41].

5.1.6. Result of Typing Experiment

To assess the influence of factors on user performance, we conducted statistical tests using SPSS software. Initially, tests were performed to examine the normality and homogeneity of variance for all collected data.

The average results of typing data are collected, as shown in Figure 10. Tests were conducted for normality and homogeneity of variances. Since the sample size for all collected data is less than 50, the Shapiro–Wilk (S-W) test was employed for the normality test. The results indicate that the number of errors (E) and error rates (ER) for all conditions followed the normal distribution (p-values of E: 0.421, 0.137, 0.188, 0.484, respectively;

p-values of ER: 0.082, 0.138, 0.338, 0.344, respectively). However, tests for homogeneity of variances indicated that the number of errors (E) (*p* = 0.011) and error rates (ER) (*p* = 0.000 **) did not meet the assumption of equal variances.

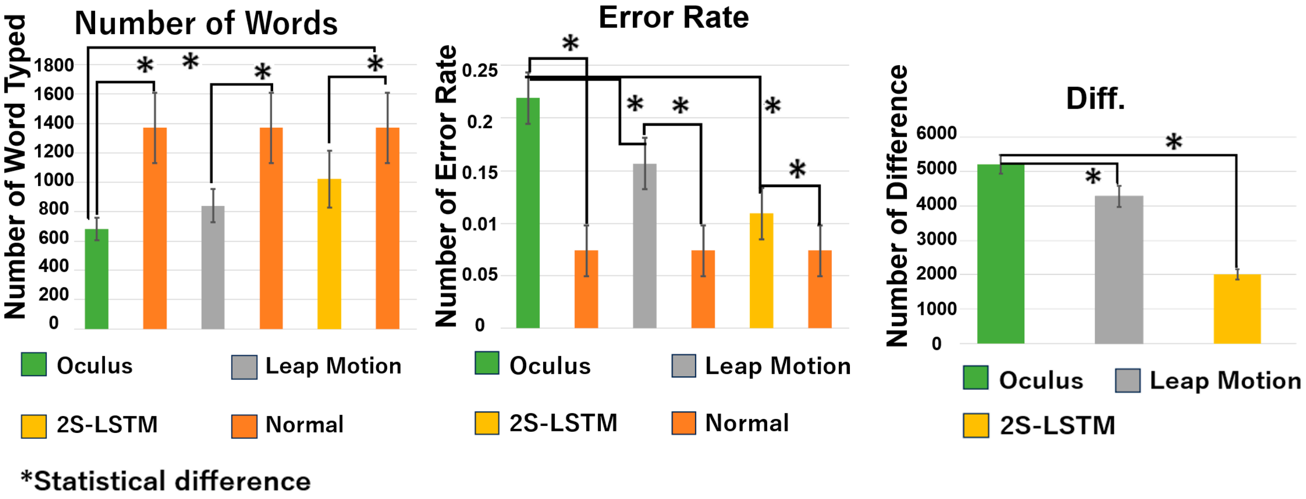


Figure 10. The result of typing data.

Moreover, none of the conditions exhibited normal distributions for the total number of words typed (NoW) and Diff. values (*p*-values of NoW: 0.001, 0.012, 0.011, 0.001, respectively; *p*-values of Diff.: 0.001, 0.013, 0.011, respectively). Therefore, non-parametric tests were employed to analyze the total number of words typed, the number of errors, error rates, and Diff. values. Since there were more than two conditions, the Kruskal–Wallis test was used to examine the differences among conditions. The results indicated significant differences among the NoW, E, ER, and Diff. conditions (*p*-values: 0.000 **, 0.001 **, 0.000 **, 0.000 **, respectively).

Multiple comparisons were conducted using the Mann–Whitney U test with Bonferroni's adjustment. For NoW, the comparison of 2S-LSTM and Leap Motion showed no significant difference (*p* = 0.357). For E, the comparison of 2S-LSTM and Leap Motion also showed no significant difference (*p* = 0.313). For other comparisons, the *p*-values are all less than 0.05. In summary, the number of NoW is Regular > 2S-LSTM = Leap Motion > Oculus, the number of E is Oculus > Leap Motion = 2S-LSTM > Regular, and the number of Diff. is Oculus > Leap Motion > 2S-LSTM, respectively.

5.1.7. Discussion

The statistical analysis demonstrates that the 2S-LSTM outperformed both the Oculus Quest 2 and Leap Motion in terms of typing efficiency, error rates, and Diff. values. These findings underscore the significance of considering the specific typing scheme when evaluating different typing assistance solutions. The Mann–Whitney U test with Bonferroni's adjustment was instrumental in drawing these conclusions.

The results of the Mann–Whitney U test with Bonferroni's adjustment indicate no significant difference between 2S-LSTM and Leap Motion in the number of inputs and errors per unit of time. Notably, our method utilizes a regular RGB camera on the HMDs, while Leap Motion employs a depth camera. Therefore, achieving comparable results to Leap Motion using a standard device is considered a positive outcome. Additionally, there is a significant difference between 2S-LSTM and the other methods in Diff. This result indicates that employing the original image and MHI, combined with implementing a Kalman filter (KF) to reduce jitter, does indeed reduce Diff. Considering the deployment cost and the overall results obtained in this research, there are compelling reasons to believe that our approach is superior to both Leap Motion and Oculus solutions.

5.2. Typing Behavior Experiment

In a previous experiment, we observed variations in finger usage among participants under different experimental conditions. Specifically, we noticed that participants' typing habits were influenced by changes in the experimental setup. Based on these observations, we hypothesize the following:

- The more effective a VR typing solution is, the less it affects the user, resulting in a smaller difference in typing habits compared to normal typing.

To further evaluate our proposed solution and verify this hypothesis, we decided to conduct a detailed analysis of the typing habit data collected under four different conditions again: regular typing, Oculus, Leap Motion, and our solution.

The overall experimental design in this experiment, including the settings for Participants, Equipment, Experimental Conditions, and Experimental Procedure, remains largely consistent with the Typing Experiment detailed in Section 5.1. To avoid redundancy, only the aspects that differ from the previous experiment will be explicitly introduced in this section. Commonalities will not be reiterated.

5.2.1. Participants

A total of 22 participants were recruited in this experiment. Unlike the previous experiment (5.1 Typing Experiment), where prior VR experience was considered, all participants in this study were VR novices, having never engaged in typing within a VR environment before. The participant group included 21 right-handed individuals and 1 left-handed individual, with 15 males and 7 females, maintaining an average age of M = 26.

All participants were proficient in English, ensuring that typing in English posed no challenges. In contrast to the previous experiment, where advanced touch-typing skills were not required, this study imposed no restrictions on participants' typing skills, allowing individuals with advanced touch-typing abilities to participate as well.

5.2.2. Equipment

The equipment setup was largely consistent with the previous experiment, with the addition of a camera and the use of Media Pipe to accurately record which keys each finger pressed during typing. This addition was specifically implemented to capture and analyze participants' typing habits more accurately.

5.2.3. Experimental Conditions and Procedure

The experimental conditions and procedures in this section were identical to those outlined in Sections 5.1.3 and 5.1.4 of the Typing Experiment. All participants underwent the same pre-experiment training session, followed the same typing tasks, and had the same flexibility to take breaks. Typing hands were recorded using the same methods, with no additional modifications to the setup.

5.2.4. Data Collection

To investigate whether the participants' typing habits changed under different VR typing conditions, we collected the following data:

- Typing habit data: We extracted the number of times each participant used each finger in four different conditions from the typing experiment.
- Typing habit difference data: We calculated the differences in typing habits by comparing the three VR typing conditions with the normal condition.

Subsequently, we performed cluster analysis and statistical analysis to determine whether the typing conditions influenced participants' typing habits and to clarify the specific nature of these changes. The typing habit data are recorded in Appendix A and show in Table A1.

It is important to note that during the actual typing tasks, participants did not use their thumbs to type on keys other than the spacebar. Therefore, we focused only on the usage of the eight fingers, excluding the thumbs. The fingers are named from the left pinky to the left index and the right index to the right pinky: L1, L2, L3, L4, R4, R3, R2, R1.

5.2.5. Use Typing Habit Data to Cluster

For all participants' typing habit data, we used k-means clustering. We used k-means clustering for two primary reasons: (1) k-means is not very sensitive to outliers in the data; and (2) k-means is well-known and easy to implement. Table 2 shows the sum of squares due to error (SSE) and average silhouette width (ASW).

Table 2. SSE and ASW values for different cluster numbers.

Cluster Number	SSE (the Sum of Squares Due to Error)	ASW (Average Silhouette Width)
2	425.782	0.380
3	379.603	0.407
4	318.158	0.495
5	301.294	0.508
6	301.862	0.509

Through practical observation, two clusters are the most suitable. One cluster consists of typists who use five fingers on each hand (referred to as "balance typists"), while the other cluster consists of typists who use only two or three fingers on each hand (referred to as "crab typists"). Although the SSE and ASW values for the four-cluster solution are better than those for the two-cluster solution, some clusters in the four-cluster solution are too small, making the two-cluster solution more practical. Details of the two-cluster solution and four-cluster solution are shown in Tables 3 and 4.

Table 3. Two-cluster solution details.

Clustering Category	Frequency	Percentage (%)
Cluster_1 (crab typist)	9	40.91%
Cluster_2(balance typist)	13	59.09%
Sum	22	100%

Table 4. Four-cluster solution details.

Clustering Category	Frequency	Percentage (%)
Cluster_1	2	9.09%
Cluster_2	13	59.09%
Cluster_3	4	18.18%
Cluster_4	3	13.64%
Sum	22	100%

Figure 11 illustrates L1 and R1 fingers usage by 22 participants under different conditions. The usage of L1 and R1 fingers in different conditions was visualized using Python, based on the typing habit data collected from participants. The data include the frequency of L1 and R1 finger usage across four typing conditions. This visualization confirms distinct differences in typing behaviors between two clusters: Cluster_1 (crab typists) and Cluster_2 (balance typists). Notably, balance typists show relatively stable usage of L1 and

R1 across different conditions, whereas crab typists exhibit an increased usage trend under the Leap and Oculus conditions compared to the normal and 2S conditions. This pattern highlights the influence of VR typing conditions on finger usage, a point further explored in the Discussion section to understand the adaptive responses of crab typists in varied VR environments.

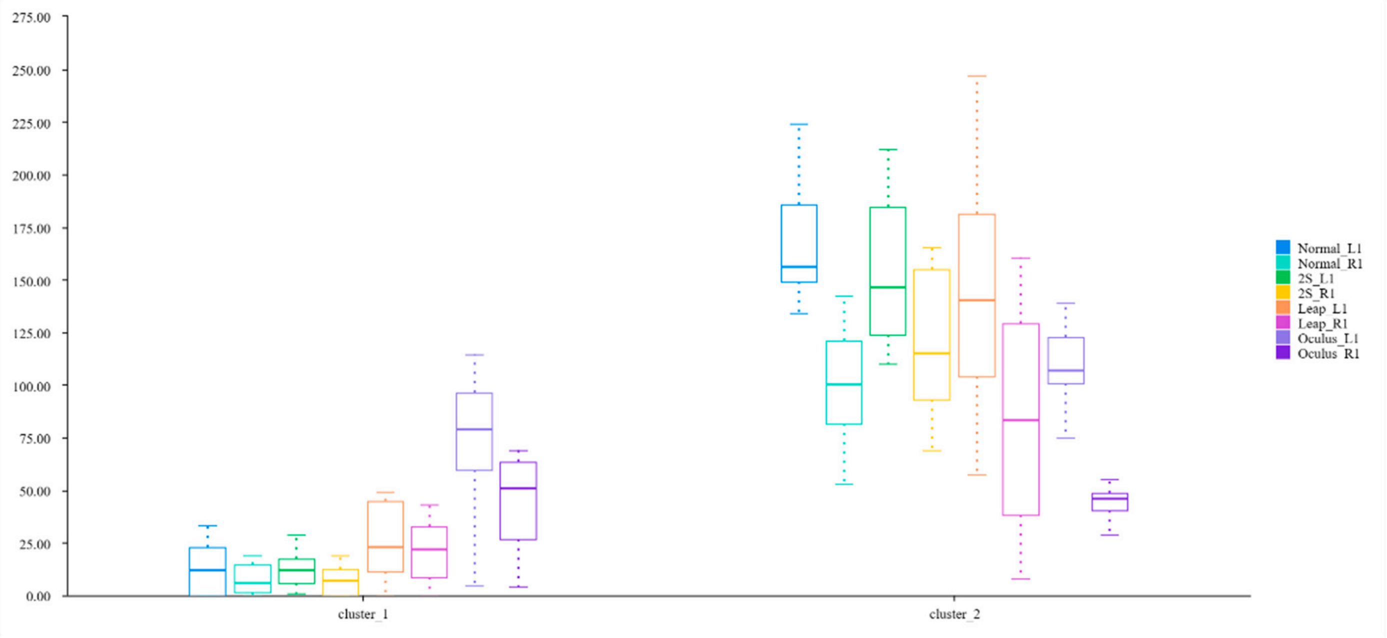


Figure 11. The usage of L1 and R1 fingers in different conditions.

5.2.6. Use Typing Habit Data to Re-Clustering

By clustering the 22 participants, we identified two clusters representing crab typists and balance typists, which aligns with our actual observations of all participants during the typing tasks. Next, we re-cluster the typing habits of these two types of typists under the four typing conditions to clarify their more detailed typing characteristics.

It is important to note that there are 9 participants in cluster 1 and 13 participants in cluster 2, which is consistent with the actual situation. Therefore, we re-clustered the typing habits of the 9 participants in cluster_1 under 4 conditions and the 13 participants in cluster_2 under 4 conditions. This results in 9 participants × 4 conditions = 36 data points for cluster 1, and 13 participants × 4 conditions = 52 data points for cluster 2.

1. Crab typists.

We used k-means to re-cluster the typing habits. The results of the re-clustering are shown in Table 5. The variance analysis results are shown in Table 6.

Table 5. Re-clustering results for crab typists.

Clustering Category	Frequency	Percentage (%)
Cluster 1_1	11	30.56%
Cluster 1_2	25	69.44%
Sum	36	100%

From Table 6, the items L1, R4, R3, and R1 exhibit highly significant differences ($p < 0.05$ or $p < 0.01$), reflecting notable changes in usage patterns. These significant results suggest that crab typists vary their usage of L1, R4, R3, and R1 across different conditions, possibly adapting these finger movements to accommodate VR-related constraints.

Table 6. Comparison results of variance analysis of clustering categories.

	Mean ± Standard Deviation		***F***	***p***
	Cluster_1 (*n* = 11)	**Cluster_2 (*n* = 25)**		
L1	72.27 ± 23.90	12.88 ± 11.02	106.175	0.000 **
L2	203.00 ± 44.58	149.08 ± 80.95	4.262	0.047 *
L3	444.18 ± 72.11	443.76 ± 90.00	0.000	0.989
L4	540.09 ± 63.17	535.12 ± 70.24	0.041	0.842
R4	557.18 ± 69.57	626.16 ± 61.57	8.865	0.005 **
R3	489.27 ± 85.11	588.92 ± 65.83	14.616	0.001 **
R2	147.82 ± 71.48	135.32 ± 67.24	0.254	0.617
R1	46.18 ± 15.78	8.76 ± 7.45	95.158	0.000 **

* $p < 0.05$; ** $p < 0.01$.

2. Balance typists.

We still used k-means to re-cluster the typing habits for balance typists. The results of the re-clustering are shown in Table 7. The variance analysis results are shown in Table 8.

Table 7. Re-clustering results for balance typists.

Clustering Category	**Frequency**	**Percentage (%)**
Cluster 2_1	22	42.31%
Cluster 2_2	30	57.69%
Sum	52	100%

Table 8. Comparison results of variance analysis of clustering categories.

	Mean ± Standard Deviation		***F***	***p***
	Cluster_1 (*n* = 22)	**Cluster_2 (*n* = 30)**		
L1	141.77 ± 51.52	144.60 ± 33.62	0.057	0.812
L2	146.55 ± 25.01	202.80 ± 30.25	50.625	0.000 **
L3	491.73 ± 73.70	486.30 ± 44.10	0.110	0.742
L4	598.36 ± 52.23	499.87 ± 56.76	40.851	0.000 **
R4	606.64 ± 34.30	525.23 ± 44.44	51.308	0.000 **
R3	279.68 ± 69.46	324.10 ± 114.05	2.617	0.112
R2	153.91 ± 44.45	226.33 ± 52.56	27.373	0.000 **
R1	81.36 ± 49.77	90.77 ± 34.92	0.642	0.427

* $p < 0.05$ ** $p < 0.01$.

Items L2, L4, R4, and R2 have p-values below 0.01, indicating significant variance across clusters. This finding implies that balance typists demonstrate notable differences in the usage of L2, L4, R4, and R2, highlighting the impact of VR environments on their typing patterns for these specific fingers.

5.2.7. Use Typing Habit Difference Data to Cluster

Typing habit difference data represent the differences in finger usage between VR conditions and normal condition. Similar to previous steps, we used k-means clustering on this data for the 22 participants. The SSE and ASW values are shown in Table 9.

Table 9. SSE and ASW values for different cluster numbers.

Cluster Number	**SSE (the Sum of Squares Due to Error)**	**ASW (Average Silhouette Width)**
2	369.933	0.292
3	290.189	0.368
4	278.493	0.387

We rely on the SSE and ASW values to determine the optimal number of clusters. As shown in Table 9, a cluster number of 3 shows an optimal inflection point for both SSE and ASW. Hence, we chose a cluster number of 3. Details of the three-cluster solution are shown in Table 10, with variance analysis results in Table 11. Here, N-2S represents the difference in typing habits between 2S-LSTM and Normal conditions, N-Le represents the difference between Leap Motion and Normal conditions, and N-Oc represents the difference between Oculus Quest 2 and Normal conditions. L1 to R1 represent different fingers.

Table 10. Three-cluster solution details.

Clustering Category	Frequency	Percentage (%)
Cluster_1	5	22.73%
Cluster_2	9	40.91%
Cluster_3	8	36.36%
Sum	22	100%

Table 11. Comparison results of variance analysis of clustering categories.

	Mean ± Standard Deviation			F	p
	Cluster_1 (n = 5)	Cluster_2 (n = 9)	Cluster_3 (n = 8)		
N-2SL1	6.20 ± 17.28	0.00 ± 14.70	19.13 ± 15.21	3.304	0.059
N-2SL2	32.00 ± 14.65	5.78 ± 10.40	21.25 ± 12.10	8.294	0.003 **
N-2SL3	36.40 ± 38.55	−28.89 ± 56.62	−38.00 ± 40.05	4.228	0.030 *
N-2SL4	−26.20 ± 38.32	18.78 ± 58.52	47.00 ± 40.70	3.491	0.051
N-2SR4	8.40 ± 17.01	−8.89 ± 32.98	−14.88 ± 16.45	1.385	0.274
N-2SR3	−27.80 ± 24.89	5.11 ± 27.71	−10.25 ± 15.67	3.259	0.061
N-2SR2	−3.80 ± 14.25	7.00 ± 10.90	−14.00 ± 12.75	6.128	0.009 **
N-2SR1	−25.20 ± 19.31	1.11 ± 4.76	−10.25 ± 15.53	6.348	0.008 **
N-LeL1	−27.80 ± 12.87	−13.89 ± 16.36	48.88 ± 34.34	20.602	0.000 **
N-LeL2	63.80 ± 10.89	−5.11 ± 32.26	10.63 ± 28.76	10.199	0.001 **
N-LeL3	34.80 ± 39.91	39.78 ± 51.98	−41.63 ± 22.61	9.739	0.001 **
N-LeL4	−67.40 ± 38.40	−41.78 ± 49.96	−3.63 ± 24.97	4.243	0.030 *
N-LeR4	−57.80 ± 18.02	2.11 ± 41.14	−28.50 ± 21.93	6.246	0.008 **
N-LeR3	11.40 ± 19.22	8.00 ± 37.68	−29.00 ± 25.60	4.074	0.034 *
N-LeR2	65.80 ± 10.89	23.22 ± 28.34	2.63 ± 28.21	9.412	0.001 **
N-LeR1	−22.80 ± 12.87	−12.33 ± 11.00	40.63 ± 36.99	14.168	0.000 **
N-OcL1	62.60 ± 42.32	−60.89 ± 34.57	55.75 ± 33.53	29.283	0.000 **
N-OcL2	71.20 ± 20.89	−46.89 ± 76.19	71.25 ± 40.37	11.830	0.000 **
N-OcL3	−78.80 ± 52.35	−48.67 ± 80.77	−75.00 ± 69.15	0.408	0.670
N-OcL4	−51.80 ± 54.61	13.78 ± 99.35	−108.00 ± 67.18	4.892	0.019 *
N-OcR4	−91.20 ± 60.13	62.00 ± 52.23	−88.63 ± 40.72	24.255	0.000 **
N-OcR3	−79.80 ± 43.34	112.11 ± 114.65	0.38 ± 127.28	5.373	0.014 *
N-OcR2	103.40 ± 74.19	5.22 ± 92.13	91.00 ± 60.74	3.618	0.047 *
N-OcR1	64.40 ± 21.31	−36.67 ± 22.01	53.25 ± 19.96	53.285	0.000 **

* $p < 0.05$ ** $p < 0.01$.

Considering the results in Table 10 and the actual types of typists, we found that the 9 crab typists were still clustered into one group, while the 13 balance typists were clustered into two groups. This indicates that the changes in typing habits among crab typists tend to be consistent, whereas the changes in typing habits among balance typists fall into two distinct categories.

From Table 11, the majority of items exhibit significant differences ($p < 0.05$ or $p < 0.01$). Under the N-Le and N-Oc conditions, nearly all items display significant differences (with only N-OcL3 showing no significance), whereas only half of the items under the N-2S condition show significant differences. This reflects variations in typing habits across different VR modes. These important findings indicate that typists adjust the usage of almost all their fingers under the Leap Motion and Oculus conditions, while only half of the finger usage patterns show changes under the 2S-LSTM condition. This further supports the idea that typists modify their finger movements to adapt to constraints specific to each VR condition.

5.2.8. Statistical Test

To identify the specific changes in finger usage for crab typists and balance typists under different conditions, we conducted statistical tests to analyze their typing habit difference data.

1. Compare crab typist's typing differences in different conditions.

Because some data lack normality and homogeneity of variance, we used Welch ANOVA, a robust alternative to standard ANOVA when assumptions of normality and homogeneity of variance are violated. This method accommodates unequal variances across groups and reduces the risk of Type I error under these conditions, making it suitable for our dataset. The results are shown in Table 12. The normality and homogeneity of variance test results are recorded in Appendix B.

Table 12. The result of welch ANOVA for crab typists.

	Condition (Standard Deviation)				Welch *F*	*p*
	Normal (n = 9)	**2S-LSTM (n = 9)**	**Leap Motion (n = 9)**	**Oculus Quest 2 (n = 9)**		
L1	12.33 ± 12.05	12.33 ± 8.28	26.22 ± 18.27	73.22 ± 32.92	10.054	0.001 **
L2	154.00 ± 76.93	148.22 ± 77.21	159.11 ± 82.56	200.89 ± 65.46	1.010	0.411
L3	434.44 ± 77.22	463.33 ± 115.12	394.67 ± 47.75	483.11 ± 65.76	3.582	0.036 *
L4	534.33 ± 64.59	515.56 ± 64.57	576.11 ± 81.64	520.56 ± 47.50	1.151	0.356
R4	618.89 ± 61.50	627.78 ± 74.65	616.78 ± 69.67	556.89 ± 64.79	2.015	0.148
R3	589.78 ± 62.61	584.67 ± 77.05	581.78 ± 67.33	477.67 ± 85.53	3.693	0.032 *
R2	148.00 ± 75.02	141.00 ± 77.18	124.78 ± 61.84	142.78 ± 65.95	0.196	0.898
R1	8.22 ± 6.89	7.11 ± 6.79	20.56 ± 14.17	44.89 ± 21.92	9.235	0.001 **

* $p < 0.05$; ** $p < 0.01$.

From Tables A2 and A3, we can see that some data do not have normality and homogeneity of variance. Therefore, we used Welch ANOVA in the next step, the results shown in Table 12.

It can be concluded that samples with different conditions do not show significant differences in terms of L2, L4, R4, and R2. However, samples with different conditions show significant differences in terms of L1, L3, R3, and R1. The analysis and comparison results of all fingers under the four conditions are shown in Figure 12.

From the above analysis, it can be concluded that crab typists exhibit different typing styles in L1, L3, R3, and R1 fingers under different conditions.

2. Compare balance typist's typing differences in different conditions.

Following the analysis of typing habit differences for crab typists, we conducted a similar analysis for balance typists. Similar to the previous step, because some data do not meet the assumptions of normality and homogeneity of variance, we used Welch ANOVA. This approach is specifically recommended for datasets with unequal variances and non-normal distributions, allowing for more accurate comparisons across the groups in question. The results are shown in Table 13, with normality and variance homogeneity test results recorded in Appendix B.

It can be concluded that samples with different conditions show significant differences in all terms except R3. The analysis and comparison results of all fingers under the four conditions are shown in Figure 13.

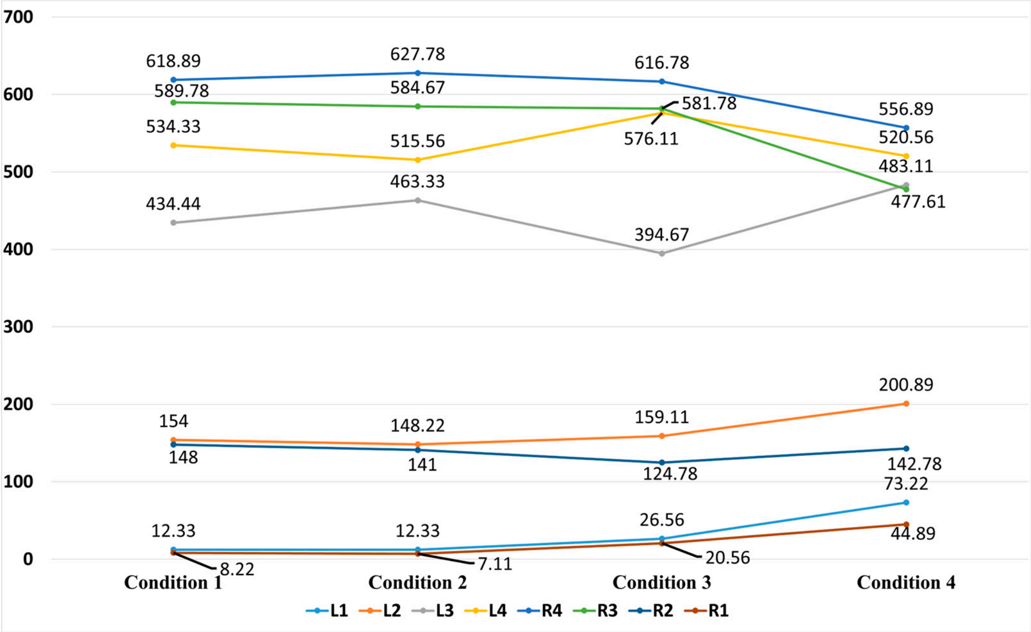


Figure 12. Comparison of crab typists' finger usage analysis under four conditions.

Table 13. The result of welch ANOVA for balance typists.

	Condition (Standard Deviation)				Welch F	p
	Normal ($n = 9$)	2S-LSTM ($n = 9$)	Leap Motion ($n = 9$)	Oculus Quest 2 ($n = 9$)		
L1	166.38 ± 25.62	152.23 ± 33.98	147.00 ± 56.55	108.00 ± 18.64	15.970	0.000 **
L2	210.92 ± 29.10	185.54 ± 33.11	179.85 ± 36.06	139.69 ± 24.17	15.560	0.000 **
L3	464.08 ± 37.30	473.46 ± 64.19	476.31 ± 59.60	540.54 ± 34.44	10.843	0.000 **
L4	517.62 ± 49.81	498.77 ± 74.43	545.77 ± 65.77	604.00 ± 60.23	6.764	0.002 **
R4	525.85 ± 49.87	531.77 ± 49.64	565.62 ± 47.87	615.46 ± 33.21	13.369	0.000 **
R3	290.08 ± 113.94	307.08 ± 106.05	303.54 ± 122.54	320.54 ± 46.98	0.308	0.819
R2	223.85 ± 48.35	233.92 ± 47.20	196.92 ± 52.32	128.08 ± 31.35	20.486	0.000 **
R1	101.23 ± 25.70	117.23 ± 32.82	85.00 ± 48.33	43.69 ± 7.51	38.024	0.000 **

* $p < 0.05$ ** $p < 0.01$.

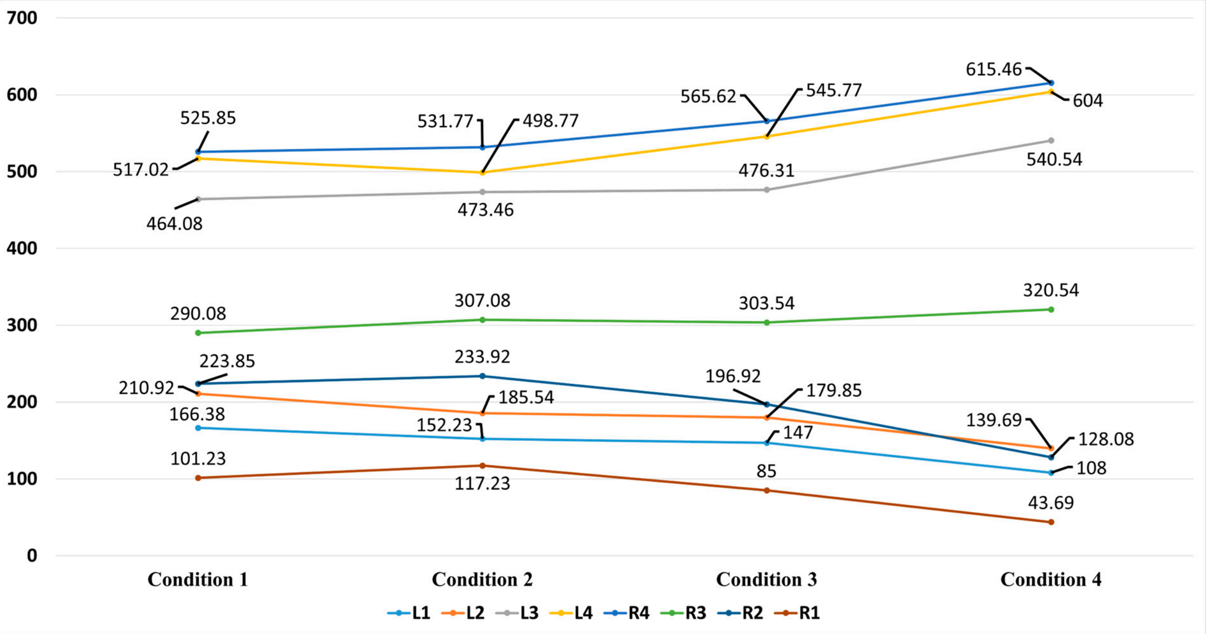


Figure 13. Comparison of balance typists' finger usage analysis under 4 conditions.

From the above analysis, we can see that in different conditions, balance typists will not change their typing habit in R3 but exhibit different typing styles in other fingers.

5.2.9. Result Summary

From the clustering of normal data combined with practical experience, it is evident that there are two types of typists: crab typists and balance typists (Sections 5.2.5–5.2.7). Both types of typists have distinct typing habits, and these habits change differently in various VR environments (Section 5.2.8). According to the actual data, compared with normal and 2S conditions, crab typists will increase the use of L1 and R1 and decrease the use of L3 and R3 in Leap and Oculus conditions (degree of change: normal <= 2S < Leap < Oculus). Conversely, balance typists will change their typing habits in a more chaotic manner (degree of change: normal < 2S < Leap < Oculus).

5.2.10. Discussion

The analysis reveals that both crab and balance typists exhibit changes in their typing habits under different VR conditions. However, the nature and extent of these changes vary between the two groups. From Section 5.2.7 we can know that crab typists show a more consistent pattern of change, while balance typists exhibit a more unpredictable alteration in their typing habits. This insight could inform the design of VR typing systems to better accommodate different typing styles and enhance user experience.

1. Behavior of balance typists.

Balance typists displayed a systematic change in their typing habits across different VR environments. Both initiative and passively changes were noted:

- Initiative changes: Balance typists consciously reduced the use of error-prone fingers (R1, R2, L1, and L2) and increased reliance on other fingers (R4, L3, and L4) to maintain typing efficiency. The reason for the change is that R1, R2, L2, and L1 are error-prone, and changes are made to maintain typing efficiency.
- Passive changes: The same shift in finger usage occurred reactively, as balance typists compensated for errors by using more reliable fingers for corrections. The reason for the change is that R1, R2, L2, and L1 are error-prone; to edit errors, use other fingers to re-type.

Interview feedback confirmed these findings, with balance typists reporting an awareness of their changing habits. They attributed these adjustments to the higher error rates and the need to maintain their overall typing speed and accuracy in VR.

2. Behavior of crab typists.

Crab typists, who typically do not use their pinkies, exhibited a unique pattern of adaptation:

- Increased pinky usage: Despite their usual reluctance, crab typists increased their use of pinkies in VR, particularly with the Oculus system. This increase, ranging from one to three times their normal usage, though still less frequent than balance typists, suggests a significant behavioral shift.
- Unawareness of changes: Unlike balance typists, crab typists often did not perceive their habits as having changed. This lack of awareness indicates an unconscious adaptation process, likely driven by the VR system's feedback mechanisms rather than a deliberate strategy.

Interviews highlighted the challenges crab typists faced, with many reporting unexpected difficulties and a heightened impact of VR hand motion accuracy. Despite these challenges, the increased pinky usage suggests that the VR environment might implicitly encourage (or force) a more balanced finger usage.

3. Common factors and additional insights.

Both groups noted the substantial impact of VR hand motion accuracy on their typing experience. This feedback aligns with the broader observations of adaptation and change in typing behavior:

- Perception of VR Tools: Many participants felt they were typing with a VR controller rather than their hands. This perception can be compared to the "fake hand experiment," where the brain is tricked into perceiving a fake hand as part of the body. In VR, if the hand models are highly realistic and closely mimic human hands, users can more easily adapt and integrate their virtual hands as part of their body. Conversely, suppose the hand models are less realistic or resemble controllers rather than hands. In that case, it becomes difficult for users to feel a natural connection, leading to disconnection and impacting their typing behavior.
- Adaptation over time: Some participants reported that the feeling of using a controller persisted throughout the experiment, while others adapted over time, suggesting that familiarity with the VR setup could reduce the sense of disconnection and lead to more stable typing habits.

6. Conclusions and Future Work

This study addresses the challenge of text entry in VR environments, particularly within immersive office settings. By leveraging machine learning techniques, the proposed 2S-LSTM typing solution, which utilizes the back of the hand image, demonstrates superior performance compared to existing solutions like the Oculus Quest 2 and Leap Motion. The 2S-LSTM solution significantly enhances typing efficiency, reduces fatigue, accurately replicates hand positions, and provides a more positive user experience. These findings underscore the potential of the developed solution to improve typing performance and user satisfaction in VR environments.

Through the performance comparison, we evaluated several advanced models, including TSSequencer, PatchTST, and BNNActionNet, across latency, accuracy, and jitter metrics. Considering latency, accuracy, and jitter performance, as well as constraints posed by our available resources, we believe that both 2S-BNNActionNet-KF and 2S-LSTM-KF are solid choices. While both 2S-LSTM-KF and 2S-BNNActionNet-KF demonstrated strong performance, we chose to proceed with 2S-LSTM-KF, as the performance gap between it and 2S-BNNActionNet-KF is minimal and imperceptible to users in the VR typing task. Additionally, the author's extensive experience with deploying LSTM on VR devices supports this decision.

Furthermore, the outcomes of this study are expected to significantly contribute to the fields of distance learning and telecommuting. Addressing the challenges of text entry in VR can facilitate the development and widespread adoption of VR technology across various applications. Future research and development efforts can focus on refining the solution and exploring its potential applications in practical settings. Additionally, expanding the sample size, incorporating additional typing metrics, and further investigating factors influencing typing performance in VR environments can provide valuable insights for developing and refining VR typing systems.

However, several avenues for future research and development could further enhance the effectiveness and user experience of VR typing systems. One key limitation of this study is the relatively small and homogeneous sample size. Future research should aim to include a larger and more diverse group of participants, helping to generalize the findings across different demographics, such as age, typing proficiency, and familiarity with VR technology.

One notable aspect of the performance comparison was the promising potential of 2S-BNNActionNet-KF. This model demonstrated strong performance in various metrics, indicating that future work could explore the replacement of LSTM-KF with 2S-BNNActionNet-KF to achieve even better results. Investigating the benefits of integrating this model may yield further improvements in typing accuracy and overall user experience.

While the 2S-LSTM typing solution has shown promise, there remains room for improvement. Future work could focus on refining the algorithm to further enhance typing accuracy and efficiency, potentially by incorporating more sophisticated machine learning techniques or adapting the algorithm to account for individual differences in typing habits.

This study primarily focused on short-term adaptation to VR typing. Future research should investigate long-term adaptation and learning effects. Understanding how typing habits evolve over extended periods of VR use could provide valuable insights into designing more intuitive and efficient typing systems.

Overall, this research has laid a strong foundation for future advancements in VR typing systems. By addressing these identified areas for future work, researchers and developers can continue to enhance VR typing solutions, making them more efficient, intuitive, and accessible to a broader range of users.

Author Contributions: Conceptualization, T.X. and S.H.; Formal analysis, T.X. and S.H.; Data curation, T.X.; Investigation, T.X.; Writing—original draft, T.X.; Methodology, T.X. and S.H.; Writing—review and editing, T.X., W.G., K.O. and S.H.; Supervision, S.H.; Validation, T.X. and S.H.; Project administration: T.X. and S.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: This study received approval from the JAIST Life Sciences Committee (H04-032).

Informed Consent Statement: Informed consent was obtained from all subjects involved in the study.

Data Availability Statement: The data are available upon request from the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

Appendix A

Table A1. The typing habit data.

Method and Participant	L1	L2	L3	L4	R4	R3	R2	R1
N and 1	185	195	503	565	513	201	256	82
N and 2	157	239	468	585	498	239	219	95
N and 3	224	186	406	571	606	162	203	142
N and 4	191	201	503	583	503	201	251	67
N and 5	152	260	422	466	496	360	244	100
N and 6	186	206	481	469	591	196	245	126
N and 7	185	186	461	526	580	216	265	81
N and 8	154	242	455	470	505	262	291	121
N and 9	141	246	463	495	428	494	139	94
N and 10	156	174	392	462	523	480	196	117
N and 11	134	174	492	562	523	240	258	117
N and 12	146	230	508	511	492	434	126	53

Table A1. *Cont.*

Method and Participant	L1	L2	L3	L4	R4	R3	R2	R1
N and 13	152	203	479	464	578	286	217	121
N and 14	14	198	446	525	644	545	109	19
N and 15	0	41	300	616	686	629	225	3
N and 16	18	190	437	503	610	532	204	6
N and 17	7	39	538	611	717	557	31	0
N and 18	33	167	476	542	616	559	91	16
N and 19	27	159	455	469	518	583	276	13
N and 20	12	113	481	566	621	579	117	11
N and 21	0	268	317	417	611	739	142	6
N and 22	0	211	460	560	547	585	137	0
2S and 1	188	156	486	569	503	230	253	115
2S and 2	124	230	482	563	466	299	245	91
2S and 3	212	160	346	621	599	191	213	158
2S and 4	200	158	565	507	520	201	244	105
2S and 5	123	247	414	469	506	374	266	101
2S and 6	181	174	559	381	594	215	244	152
2S and 7	146	182	525	459	564	256	297	71
2S and 8	153	200	422	493	496	293	285	158
2S and 9	114	230	496	452	443	511	159	95
2S and 10	132	156	478	366	560	472	214	122
2S and 11	110	156	478	566	547	245	276	122
2S and 12	132	204	511	498	521	429	136	69
2S and 13	164	159	393	540	594	276	209	165
2S and 14	18	191	511	461	689	519	102	9
2S and 15	12	18	322	601	732	603	212	0
2S and 16	29	182	404	536	631	511	196	11
2S and 17	1	41	633	516	721	553	35	0
2S and 18	10	162	572	476	630	545	86	19
2S and 19	7	172	483	441	560	541	289	7
2S and 20	5	114	548	509	591	609	110	14
2S and 21	12	258	295	469	584	761	117	4
2S and 22	17	196	402	631	512	620	122	0
Leap and 1	231	116	461	636	592	166	175	123
Leap and 2	172	186	463	625	542	243	164	105
Leap and 3	247	126	400	610	657	159	141	160
Leap and 4	166	181	563	566	534	224	229	37
Leap and 5	140	230	491	438	520	386	212	83
Leap and 6	104	232	535	468	600	253	269	39
Leap and 7	117	201	482	556	612	246	278	8
Leap and 8	190	171	354	602	580	233	218	152
Leap and 9	136	209	529	468	504	470	100	84
Leap and 10	76	199	418	489	529	538	219	32
Leap and 11	104	157	509	588	557	264	239	82
Leap and 12	57	183	528	530	508	472	157	65
Leap and 13	171	147	459	519	618	292	159	135
Leap and 14	38	175	370	600	699	494	81	43
Leap and 15	15	77	351	573	653	670	157	4
Leap and 16	23	230	387	561	571	569	144	15
Leap and 17	0	46	474	669	702	566	43	0
Leap and 18	40	160	461	555	644	533	84	23
Leap and 19	49	141	364	556	555	542	257	36
Leap and 20	14	110	429	616	659	545	114	13
Leap and 21	8	323	349	392	560	699	147	22
Leap and 22	49	170	367	663	508	618	96	29

Table A1. *Cont.*

Method and Participant	L1	L2	L3	L4	R4	R3	R2	R1
Ocu and 1	101	119	507	671	600	314	142	46
Ocu and 2	107	152	555	623	597	294	132	40
Ocu and 3	100	129	553	616	604	301	149	48
Ocu and 4	111	147	582	619	614	299	99	29
Ocu and 5	101	132	562	643	598	306	109	49
Ocu and 6	75	96	524	646	634	325	153	47
Ocu and 7	117	139	586	616	604	299	97	42
Ocu and 8	139	149	551	569	670	306	66	50
Ocu and 9	106	128	561	644	575	302	141	43
Ocu and 10	128	124	549	630	595	298	135	41
Ocu and 11	130	143	474	586	643	317	159	48
Ocu and 12	77	198	484	554	582	472	103	30
Ocu and 13	112	160	539	435	685	334	180	55
Ocu and 14	55	229	497	496	550	489	133	51
Ocu and 15	114	233	549	420	592	312	247	33
Ocu and 16	64	123	450	543	658	559	34	69
Ocu and 17	0	61	563	563	645	571	93	4
Ocu and 18	81	207	456	549	575	461	124	47
Ocu and 19	103	211	435	560	494	551	126	20
Ocu and 20	74	244	549	497	497	441	144	54
Ocu and 21	89	249	360	560	507	520	147	68
Ocu and 22	79	251	489	497	494	395	237	58

Appendix B

Table A2. Normality test for crab typists.

	Sample Size	Average	SD	Skewness	Kurtosis	Shapiro–Wilk Test	
						W	*p*
L1	36	31.028	31.881	1.148	0.335	0.850	0.000 **
L2	36	165.556	75.472	−0.320	−0.525	0.961	0.231
L3	36	443.889	83.906	0.077	−0.569	0.978	0.670
L4	36	536.639	67.295	−0.146	−0.302	0.982	0.813
R4	36	605.083	70.860	0.035	−1.019	0.959	0.195
R3	36	558.472	84.902	−0.166	2.005	0.943	0.064
R2	36	139.139	67.778	0.568	−0.230	0.941	0.054
R1	36	20.194	20.368	1.095	0.168	0.855	0.000 **

* $p < 0.05$ ** $p < 0.01$.

Table A3. Homogeneity of variance test for crab typists.

	Condition (Standard Deviation)				*F*	*p*
	Normal (*n* = 9)	2S-LSTM (*n* = 9)	Leap Motion (*n* = 9)	Oculus Quest 2 (*n* = 9)		
L1	12.05	8.28	18.27	32.92	2.904	0.050 *
L2	76.93	77.21	82.56	65.46	0.135	0.938
L3	77.22	115.12	47.75	65.76	2.916	0.049 *
L4	64.59	64.57	81.64	47.50	0.240	0.868
R4	61.50	74.65	69.67	64.79	0.457	0.714
R3	62.61	77.05	67.33	85.53	0.491	0.691
R2	75.02	77.18	61.84	65.95	0.333	0.802
R1	6.89	6.79	14.17	21.92	4.850	0.007 **

* $p < 0.05$ ** $p < 0.01$.

Table A4. Normality test for balance typists.

	Sample Size	Average	SD	Skewness	Kurtosis	Kolmogorov–Smirnov Test	
						D	***p***
L1	52	143.404	41.684	0.350	−0.136	0.072	0.720
L2	52	179.000	39.565	0.133	−0.738	0.108	0.140
L3	52	488.596	57.875	−0.489	−0.202	0.093	0.316
L4	52	541.538	73.277	−0.316	−0.674	0.110	0.121
R4	52	559.673	57.063	−0.072	−0.428	0.112	0.099
R3	52	305.308	99.366	0.776	−0.286	0.170	0.001 **
R2	52	195.692	60.744	−0.196	−1.083	0.114	0.091
R1	52	86.788	41.657	0.202	−1.025	0.123	0.046 *

* $p < 0.05$ ** $p < 0.01$.

Table A5. Homogeneity of variance test for balance typists.

	Condition (Standard Deviation)				***F***	***p***
	Normal (*n* = 9)	2S-LSTM (*n* = 9)	Leap Motion (*n* = 9)	Oculus Quest 2 (*n* = 9)		
L1	25.62	33.98	56.55	18.64	6.124	0.001 **
L2	29.10	33.11	36.06	24.17	1.263	0.298
L3	37.30	64.19	59.60	34.44	1.956	0.133
L4	49.81	74.43	65.77	60.23	0.657	0.582
R4	49.87	49.64	47.87	33.21	1.120	0.350
R3	113.94	106.05	122.54	46.98	4.880	0.005 **
R2	48.35	47.20	52.32	31.35	1.198	0.320
R1	25.70	32.82	48.33	7.51	7.764	0.000 **

* $p < 0.05$ ** $p < 0.01$.

References

1. Hodgson, P.; Lee, V.; Chan, J.; Fong, A.; Tang, C.; Chan, L.; Wong, C. Immersive virtual reality (IVR) in higher education: Development and implementation. In *Augmented Reality and Virtual Reality: The Power of AR and VR for Business*; Springer: Berlin/Heidelberg, Germany, 2019; pp. 161–173.
2. Christopoulos, A.; Conrad, M.; Shukla, M. Increasing student engagement through virtual interactions: How? *Virtual Real.* **2018**, *22*, 353–369. [CrossRef]
3. Tunk, N.; Kumar, A. Work from home—A new virtual reality. *Curr. Psychol.* **2023**, *42*, 30665–30677. [CrossRef] [PubMed]
4. Bowman, D.; Rhoton, C.; Pinho, M. Text Input Techniques for Immersive Virtual Environments: An Empirical Comparison. In *Proceedings of the Human Factors and Ergonomics Society Annual Meeting*; SAGE Publications: Los Angeles, CA, USA, 2002; Volume 46, pp. 2154–2158.
5. Grubert, J.; Witzani, L.; Ofek, E.; Pahud, M.; Kranz, M.; Kristensson, P. Text Entry in Immersive Head Mounted Display Based Virtual Reality Using Standard Keyboards. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany, 18–22 March 2018; pp. 159–166.
6. Grubert, J.; Witzani, L.; Ofek, E.; Pahud, M.; Kranz, M.; Kristensson, P. Effects of Hand Representations for Typing in Virtual Reality. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany, 18–22 March 2018; pp. 151–158.
7. Boletsis, C.; Kongsvik, S. Text Input in Virtual Reality: A Preliminary Evaluation of the Drum-Like VR Keyboard. *Technologies* **2019**, *7*, 31. [CrossRef]
8. Otte, A.; Schneider, D.; Menzner, T.; Gesslein, T.; Gagel, P.; Grubert, J. Evaluating Text Entry in Virtual Reality using a Touch-sensitive Physical Keyboard. In Proceedings of the 2019 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct), Beijing, China, 10–18 October 2019; pp. 387–392.
9. Meier, M.; Streli, P.; Fender, A.; Holz, C. TapID: Rapid Touch Interaction in Virtual Reality using Wearable Sensing. In Proceedings of the 2021 IEEE Virtual Reality and 3D User Interfaces (VR), Lisboa, Portugal, 27 March–1 April 2021; pp. 519–528.
10. Hwang, D.; Aso, K.; Koike, H. MonoEye: Monocular Fisheye Camera-based 3D Human Pose Estimation. In Proceedings of the 2019 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Osaka, Japan, 23–27 March 2019; pp. 988–989.
11. Wu, E.; Ye, Y.; Yeo, H.; Quigley, A.; Koike, H.; Kitani, M. Back-Hand-Pose: 3D Hand Pose Estimation for a Wrist-Worn Camera via Dorsum Deformation Network. In Proceedings of the 33rd Annual ACM Symposium on User Interface Software and Technology, Virtual, 20–23 October 2020; Association for Computing Machinery: New York, NY, USA, 2020; pp. 1147–1160.

12. Fourrier, N.; Moreau, G.; Benaouicha, M.; Norm, J. Handwriting for Efficient Text Entry in Industrial VR Applications: Influence of Board Orientation and Sensory Feedback on Performance. *IEEE Trans. Vis. Comput. Graph.* **2023**, *29*, 4438–4448. [CrossRef] [PubMed]
13. Kim, T.; Karlson, A.; Gupta, A.; Grossman, T.; Wu, J.; Abtahi, P.; Collins, C.; Glueck, M.; Surale, H. STAR: Smartphone-analogous Typing in Augmented Reality. In Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, San Francisco, CA, USA, 29 October–1 November 2023; pp. 1–13.
14. Stauffert, J.; Niebling, F.; Latoschik, M. Effects of Latency Jitter on Simulator Sickness in a Search Task. In Proceedings of the 2018 IEEE Conference on Virtual Reality and 3D User Interfaces (VR), Tuebingen/Reutlingen, Germany, 18–22 March 2018; pp. 121–127.
15. Tatsunami, Y.; Masato Taki, M. Sequencer: Deep LSTM for Image Classification. *arXiv* **2020**, arXiv:2205.01972.
16. Nie, Y.; Nguyen, N.; Sinthong, P.; Kalagnanam, J. A Time Series is Worth 64 Words: Long-term Forecasting with Transformers. *arXiv* **2022**, arXiv:2211.14730.
17. Fontana, F.; Matteo, A.; Cinque, L.; Placidi, G.; Marini, M. BNNAction-Net: Binary Neural Network on Hands Gesture Recognitions. In Proceedings of the ACM SIGGRAPH 2024 Posters (SIGGRAPH'24), Denver, CO, USA, 26–28 July 2024; Association for Computing Machinery: New York, NY, USA, 2024; pp. 1–2.
18. Gil, H.; Oakley, I. ThumbAir: In-Air Typing for Head Mounted Displays. In *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*; Association for Computing Machinery: New York, NY, USA, 2023; Volume 6, pp. 1–30.
19. Zhang, F.; Bazarevsky, V.; Vakunov, A.; Tkachenka, A.; Sung, G.; Chang, C.; Grundmann, M. MediaPipe Hands: On-device Real-time Hand Tracking. *arXiv* **2020**, arXiv:2006.10214.
20. Johnson, S.; Everingham, M. Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In Proceedings of the British Machine Vision Conference, Aberystwyth, UK, 31 August–3 September 2010.
21. Mueller, F.; Bernard, F.; Sotnychenko, O.; Mehta, D.; Sridhar, S.; Casas, D.; Theobalt, C. GANerated Hands for Real-Time 3D Hand Tracking from Monocular RGB. In Proceedings of the 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 49–59.
22. Jang, Y.; Jeon, I.; Kim, T.; Woo, W. Metaphoric Hand Gestures for Orientation-Aware VR Object Manipulation with an Egocentric Viewpoint. *IEEE Trans. Hum.-Mach. Syst.* **2017**, *47*, 113–127. [CrossRef]
23. Teather, R.; Pavlovych, A.; Stuerzlinger, W.; MacKenzie, I. Effects of Tracking Technology, Latency, and Spatial Jitter on Object Movement. In Proceedings of the 2009 IEEE Symposium on 3D User Interface, Lafayette, LA, USA, 14–15 March 2009; pp. 43–50.
24. Pavlovych, A.; Stuerzlinger, W. The Tradeoff between Spatial Jitter and Latency in Pointing Tasks. In Proceedings of the 1st ACM SIGCHI Symposium on Engineering Interactive Computing Systems, Pittsburgh, PA, USA, 15–17 July 2009; pp. 187–196.
25. Batmaz, A.; Seraji, M.; Kneifel, J.; Stuerzlinger, W. No Jitter Please: Effects of Rotational and Positional Jitter on 3D Mid-Air Interaction. In Proceedings of the Future Technologies Conference (FTC); Springer International Publishing: Cham, Switzerland, 2020; Volume 2, pp. 792–808.
26. Mughrabi, M.; Mutasim, A.; Stuerzlinger, W.; Batmaz, A. My Eyes Hurt: Effects of Jitter in 3D Gaze Tracking. In Proceedings of the 2022 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW), Christchurch, New Zealand, 12–16 March 2022; pp. 310–315.
27. Wang, W.; Yu, K.; Hugonot, J.; Fua, P.; Salzmann, M. Beyond One Glance: Gated Recurrent Architecture for Hand Segmentation. *arXiv* **2018**, arXiv:1811.10914.
28. Afifi, M. 11K Hands: Gender Recognition and Biometric Identification Using a Large Dataset of Hand Images. *Multimed. Tools Appl.* **2017**, *78*, 20835–20854. [CrossRef]
29. Qian, C.; Sun, X.; Wei, Y.; Tang, X.; Sun, J. Realtime and Robust Hand Tracking from Depth. In Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, USA, 23–28 June 2014; pp. 1106–1113.
30. Roth, J.; Liu, X.; Metaxas, D. On Continuous User Authentication via Typing Behavior. *IEEE Trans. Image Process.* **2014**, *23*, 4611–4621. [CrossRef] [PubMed]
31. Bobick, A.; Davis, J. The Recognition of Human Movement Using Temporal Templates. *IEEE Trans. Pattern Anal. Mach. Intell.* **2001**, *23*, 257–267. [CrossRef]
32. Tsai, D.; Chiu, W.; Lee, M. Optical Flow-Motion History Image (OF-MHI) for Action Recognition. *Signal Image Video Process.* **2015**, *9*, 1897–1906. [CrossRef]
33. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep Residual Learning for Image Recognition. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
34. Hochreiter, S.; Schmidhuber, J. Long Short-Term Memory. *Neural Comput.* **1997**, *9*, 1735–1780.
35. Welch, G.; Bishop, G. *An Introduction to the Kalman Filter*; University of North Carolina: Chapel Hill, NC, USA, 1995.
36. Coskun, H.; Achilles, F.; DiPietro, R.; Navab, N.; Tombari, F. Long Short-Term Memory Kalman Filters: Recurrent Neural Estimators for Pose Regularization. In Proceedings of the 2017 IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 22–29 October 2017; pp. 5525–5533.

37. GPU Score Legacy Products. Available online: https://www.gpuscore.com/benchmarks/legacy-products/ (accessed on 8 November 2024).
38. Simon, D.; Keith, N.; Eugene, N. A Systematic Review of Cybersickness. In Proceedings of the 2014 Conference on Interactive Entertainment, Newcastle, NSW, Australia, 2–3 December 2014; pp. 1–9.
39. Hou, X.; Lu, Y.; Dey, S. Wireless VR/AR with Edge/Cloud Computing. In Proceedings of the 2017 26th International Conference on Computer Communication and Networks (ICCCN), Vancouver, BC, Canada, 31 July–3 August 2017; pp. 1–8.
40. Jerald, J. Scene-Motion- and Latency-Perception Thresholds for Head-Mounted Displays. Ph.D. Thesis, University of North Carolina, Chapel Hill, NC, USA, 2009.
41. Xu, T.; Gu, W.; Ota, K.; Hasegawa, S. A Low-Jitter Hand Tracking System for Improving Typing Efficiency in Virtual Reality Workspace. In Proceedings of the TENCON 2023—2023 IEEE Region 10 Conference (TENCON), Chiang Mai, Thailand, 31 October–3 November 2023; pp. 1–6.
42. Tejo, C.; Aljosa, S. Simultaneous Segmentation and Recognition: Towards More Accurate Ego Gesture Recognition. In Proceedings of the 2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW), Seoul, Republic of Korea, 27–28 October 2019; pp. 4367–4375.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.