Journal of Engineering Science and Technology
21st EURECA 2024 Special Issue August (2025) 132 - 147


# 3D SPATIAL FINGER RECOGNITION FOR TEXT INPUT IN XR USING MMWAVE RADAR SENSING

WEI EAN TEOH[1], HAFISOH AHMAD[1,*],
CHEW WEI JEN[1], HAFIZUL AZIZI BIN ISMAIL@AZIZ[2]

[1]School of Engineering, Taylor's University, Taylor's Lakeside Campus,
No. 1 Jalan Taylor's, 47500, Subang Jaya, Selangor DE, Malaysia
[2]Asia Pacific University of Technology and Innovation
Asia Pacific University of Technology & Innovation (APU), Jalan Teknologi 5, Kuala Lumpur, Wilayah Persekutuan Kuala Lumpur,
*Corresponding Author: Hafisoh.Ahmad@taylors.edu.my

## Abstract

This paper investigates the integration of mmWave radar sensor technology into Extended Reality (XR) environments for gesture-based text input, focusing on fine-grained finger movement detection for accurate key typing. The study begins with a comprehensive literature review and the selection of Seeed Studio's MR24HPC1 mmWave radar sensor based on such criteria e.g. sensitivity, range, and affordability. Empirical data is then collected from various finger gestures before performing identification studies on significant motion characteristics. These findings are translated into a lightweight algorithm for real-time gesture recognition, implemented in Unity. The system's performance is evaluated in diverse XR environments, highlighting its accuracy and responsiveness. Results indicate that motion energy and speed values are crucial for distinguishing key presses and directional movements, although challenges such as background noise and limited sensor protocol granularity are noticeable. Implementation on Unity XR with the sensor and Arduino UNO is successful and an accuracy study is done and found to have an average accuracy of 62.78%. This study demonstrates a possibility of using mmWave radar for XR text input, offering a promising avenue for enhancing user interaction in virtual environments. Future work will refine the gesture recognition algorithm and further integrate the system within the XR interface.

Keywords: Gesture recognition, mmWave, Radar sensing, XR text input.

## 1. Introduction

As Extended Reality (XR) has started converging in our daily life due to its advancement of technology and applications. From being a visual immersive experience, into entertainment and games, and now the potential of XR has been seen to increase the productivity of works. However, with all these desire for better application of XR, there is a need for advancements in human-computer interaction (HCI) with the XR equipment. Text input plays a vital role in enabling many possibilities in XR; with current text input systems in XR environments have several limitations, including cumbersome interfaces, lack of tactile feedback, and inefficient gesture recognition.

These systems often rely on laser pointing or computer vision on virtual keyboards, which can be slow, inaccurate, and unsuitable for certain contexts. Virtual keyboards can be particularly challenging due to the lack of physical reference points, leading to frequent errors and reduced typing speed. These issues necessitate the exploration of alternative input methods that can provide a more natural and efficient user experiences.

Research has explored various implementations, ranging from wearables and sensors to computer vision systems. Initially relying on devices like data gloves [1] and motion sensor rings, equipped with accelerometers and gyroscopes, the field witnessed a shift towards visual gesture recognition technology, eliminating the need for wearables but raising concerns related to lighting conditions and privacy [2]. Unlike to optical sensors, Millimetre-wave (mmWave) sensors seem like an alternative option due to its less affected by lighting conditions and occlusions, providing robust performance in various settings.

Gesture recognition using Millimetre-wave (mmWaves) has gained a considerable amount of research attention in the recent years for its ability to detect range and motion with high accuracy. This is achieved through analysing data from the frequency modulated continuous wave (FMCW) with the principle of Doppler's effect.

The mmWave technology has shown promising results in fine-grained gesture recognition, with applications in tracking range, velocity, and angle [3, 4]. The mmWave sensors operate by emitting electromagnetic waves and analysing the reflected signals to determine the position and motion of objects. This capability makes them ideal for capturing subtle finger movements required for typing in XR environments.

Despite advancements in deep learning algorithms enhancing accuracy, the computational demands pose challenges for real-time applications, such as keyboard typing in virtual reality. Current applications of mmWave technology primarily focus on capturing large gestures [5], leaving a research gap in exploring its potential for finer-grained hand motions, particularly in real-time keyboard typing interactions within the augmented and virtual reality domain.

Addressing this gap is crucial for maximizing user experiences and unlocking the full potential of these technologies. Therefore, this project aims to bridge the gap between traditional text input methods through utilising mmWaves to detect 3D spatial micro-gestures of the fingers typing for the application of text input on XR.he shape of a projectile is generally selected on the basis of combined aerodynamic, guidance, and structural considerations. The choice of seeker, at supersonic speeds, careful selection of the nose and tail shapes is mandatory to ensure performance and operation of the over-all system.

## 2. Research Background

### 2.1. Text input in extended reality (XR)

Extended Reality (XR) is an umbrella term that covers virtual reality (VR), augmented reality (AR), and mixed reality (MR) technologies and this field is experiencing rapid growth [1], and the need for an intuitive text input method is growing more and more prominent as users endeavour to access more capabilities on XR such as typing input to search the internet, chatting and interacting with people, or for productivity like doing work and typing long texts, Fig. 1 shows computer monitors being casted into AR for productivity. AR smart glasses like Spacetop or Lumus claims to be effectively a laptop without a screen, that allows users for a lightweight, vision-wide productivity screen layout [2].

**Fig. 1. Productivity uses on AR [6].**

However, while there are demands for better text input systems, the current implemented solutions have yet to mature to its expected efficiency. Here are some of the methods for text input. Users can plug in an external physical keyboard to enable typing on XR [3]; however, this method is not hands-free and there occurs a problem of not being able to see the physical keyboard when users put on the headsets, and this has reported to significantly hinder typing speeds. For simple text inputs, users can use the simple point and click with the VR laser controllers one alphabet by one alphabet, demonstrated in Fig. 2, which satisfies the need of text input, but has its limitation of being extremely tedious with low words per minute.

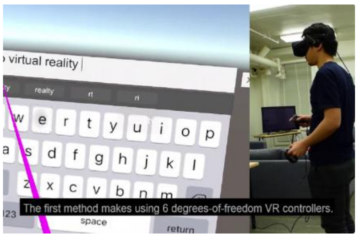



**Fig. 2. Controllers point method of typing.**

Computer vision with hand tracking has also been an increasingly popular technology that is being extensively improved and optimized among research as well as in the industry, such as Quest by Meta [4] and Leap Motion by Ultraleap [5]. Computer vision is an attractive solution as XR headsets more than often already come with cameras so this will mean that no extra hardware is needed to realize this solution.

Nevertheless, this innovation comes with its own set of challenges, such challenges as to it is sensitive to lighting conditions, interference, and occlusions [7], it is also limited to the line-of-sight [1], 3D depth, and its frame size [6] of the cameras and screens, moreover, it is also subjected to scaling and distance approximation issues, all these leading to challenges in accuracy and precision [4]. Another noble in market solution by TapXR [8] utilizes an embedded neural processor on the wrist to determine which finger has been tapping on a hard surface [9]. However, as they are limited to only knowing which finger is tapping, they have redesigned the layout of a keyboard however this poses a learning curve for users [10].

### 2.2. Wireless radar sensing for gesture recognition

In contrast to the previous methods described above, wireless radar sensing methods have an advantage in terms of overcoming problems of occlusion, obstacles, line-of-sight, that computer vision methods face by establishing a contactless and 3-dimensional recognition paradigm [11].

The earlier research in radar sensing gesture recognition focuses the use of infrared cameras arranged in arrays to implement as a motion tracking system [12], in the more recent works, higher frequency ranged electromagnetic waves are explored, such as Wi-Fi, ultrasound, radio waves, and mmWaves. Wi-Fi gesture recognition takes advantage of the Channel State Information (CSI) available from the household infrastructure to detect unique patterns in the detailed CSI [13], hence eliminating the need for extra devices.

However, this method is highly susceptible to background interference [14] and has due to relatively higher wave lengths and shorter range resolutions of Wi-Fi signals [15], it can only work well with coarse-grained-gestures such as body activities and hand movements [16]. Extracting data from such can also pose security threats when important data is being conveyed in VR such as passwords and security keys [17]. Another wavelength residing in the radio wave band, the mmWaves is found to be more suited for fine grained gesture [18] detection as it has a higher range and angle resolution [15].

### 2.3. Millimeter wave (mmWave) for gesture recognition in XR text input

Compared to Wi-Fi, mmWave-based solution has a more superior performance [15] in terms of detecting fine grained gestures, attributed to it being at a higher frequency band and short wavelength [19], at around 60GHz [20], which has wavelength at the millimetre which is capable of detecting find grained movements [21]. It utilizes Doppler’s effect to capture spatial and temporal information of objects through transmitting Frequency-Modulated Continuous-Wave (FMCW) [20]. It’s high range and angle resolution makes it a remarkably precise sensor and gives it the ability of achieving sub-millimetre gesture recognition [22].

mmWave also stands an edge over computer vision and hand tracking approaches in these ways. It is more lightweight in computation [20], making it a more attractive solution for real-time tracking. It is also independent of being subjected to environment interference such as occlusions, background clutter, lighting, and line-of-sight limitations. In contrast of Wi-Fi and CSI based solutions, it is privacy preserving that it is not easily tapped, unlike CSI [7]. There is also an availability of component for commercialisation and research, such as Google Soli and Texas Instruments' (TI) AWR/IWR sensors [15]. All of these advantages make mmWave a promising approach for it to be further researched.

Transmitted FMCW is received as it bounces back when it detects an obstacle. This received signal is processed and it can give information about the range (distance), phase (velocity), and angle of the obstacle. This processed raw radar signals can then be packaged into matrices to form frequency domain graphs of Range Doppler Image (RDI), Range Angle Image (RAI), Doppler Angle Image (DAI) or Micro-Doppler Spectogram [23] using 2D Fast Fourier Transform FFT to identify gestures.

In response to the growing demand for improved text input methods within extended reality (XR) environments, millimetre waves (mmWaves) emerge as a compelling avenue for further research. Their unique properties of being able to detect minute, fine-grained gestures, it is not affected by occlusions, it is compact in size, and computationally lightweight [24], make mmWaves an attractive technology for this purpose. In this project, it is aimed for the implementation of input from mmWave radar sensor to identify a characteristic movement that can indicate a distinctive gesture. This implementation is aimed to be visualized on XR (Meta Quest 2 or Microsoft Hololens 2) developed in Unity using C# scripting [25].

## 3. Methods of Implementation

The first stage of project consists of collecting empirical data samples to be analysed graphically for characteristic identifiers to be set as the threshold. Next, a lightweight algorithm is developed upon the identified characteristics of each action. This is then implemented onto the XR interface of Unity.

### 3.1. Hardware infrastructure

The hardware consists of, MR24HPC1 Sensor, Arduino UNO, Laptop that facilitates serial communication between Arduino UNO and Unity XR simulation. MR24HPC1 collects and sends an array of data to Arduino UNO. With the received data, Arduino UNO filters faulty entries, parses the array, and converts it into clean data. With the cleaned data, a simple algorithm is implemented to decide for which key is pressed. This hardware infrastructure is shown in Fig. 3.

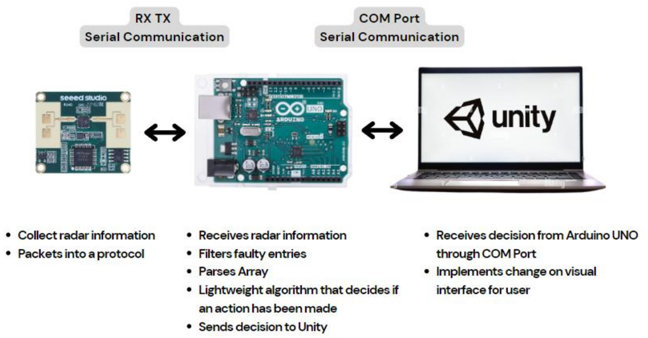



**Fig. 3. Hardware infrastructure of system.**

## 3.2. Data collection and analysis

Empirical research and observation are done to determine for the characteristic identifier of each action. The data is also processed and displayed graphically to identify any distinctive characteristics of each gesture.

The MR24HPC1 sensor by Seeed Studio is used to collect different set of data from various finger gestures. The 24 GHz mmWave sensor is a self-contained sensor which consists of the RF antenna, sensor chip and high speed MCU, with several groups of GPIOs for customization and development when equipped with a host computer, to display output data as well as to input commands.

The sensor is attached to the arm as shown in Fig. 4, as different gestures are repeated in a timeframe of 2 minutes each and the data is collected through the Arduino Uno. Total of 9 different gestures are performed, which are, no obstacles (straight hands), all fingers in resting position, index finger in resting position only, all fingers moving in typing motion, index finger moving in typing motion, all fingers moving front and back, index finger moving front and back, all fingers moving left and right, and index finger moving left and right.

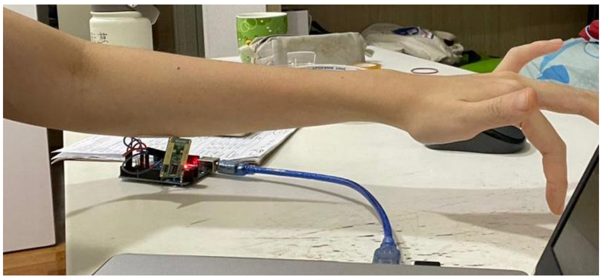


**Fig. 4. Attached sensor to detect gesture.**

A frame of data is collected every 200 ms. Each data frame contains information about the Existence Energy Value, Static Distance Range, Motion Energy Value, Motion Distance Range, and Motion Speed Range. As it is a self-contained system, the data is already pre-processed and packaged into 8-bits in the format of hex codes for the case of Existence Energy Value and Motion Energy Value, or in their predefined range in the case of the Statis Distance Range, Motion Distance Range, and Motion Speed Range. These are defined in their datasheet. The observed data frame structure is explained in Table 1.

**Table 1. Definition of received data frame structure.**

| Data structure | Definition | Example | Byte |
|---|---|---|---|
| **Frame Header** | Initiates the start of a frame | 0×53<br>0×59 | 2 |
| **Control Word** | To identify the type of content | 08 | 1 |
| **Command Word** | To identify current data content | 01 | 1 |
| **Length Identifier** | Equal to specific byte length of data | 00<br>05 | 2 |
| **Data: Existence Energy Value** | Electromagnetic wave frequencies around. | 8D | 1 |

| | | | |
|---|---|---|---|
| **Data: Static Distance Range** | Straight line distance | 01 | 1 |
| **Data: Motion Energy Value** | Amplitude of motion causing electromagnetic wave frequency changes. | 3E | 1 |
| **Data: Motion Distance Range** | Distance of moving target. | 01 | 1 |
| **Data: Motion Speed Range** | Speed of moving target. 0×01 – 0×09 for approaching, 0×0A for no motion speed, 0×0b – 0×14 for moving away. | 0A | 1 |
| **Checksum** | Checks accuracy of sent data. Sum of frame header, control word, command word, length identifier, and data summed to the lower eight bits | 91 | 1 |
| **End of frame** | Signified the end of the frame | 0×54 0×43 | 2 |

A sample of the received response is shown in Fig. 5. The data hygiene is maintained by filtering out incorrect data. Incorrect data includes data that does not return the correct control word, command word, length, and checksum codes are removed in excel functions. The data is then put into excel tables to be analysed through plotting comparison graphs.

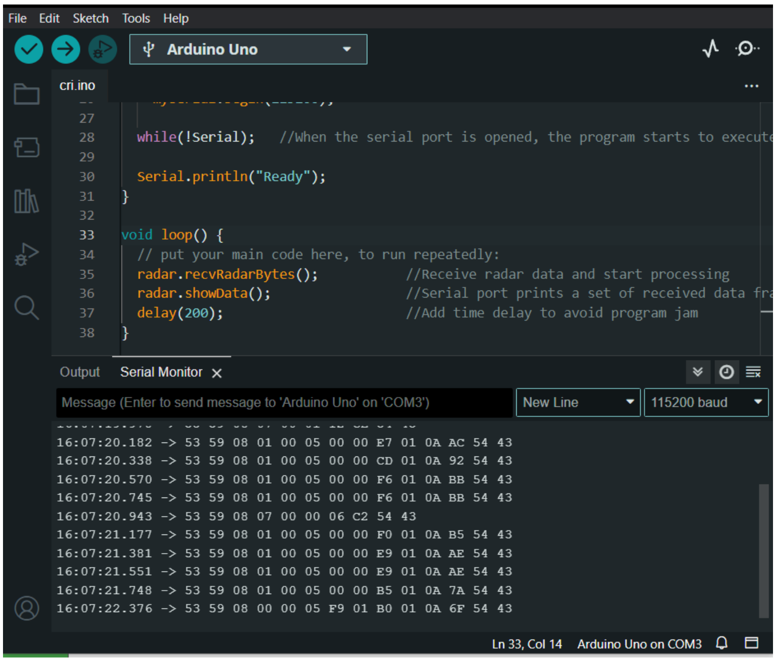



**Fig. 5. Sample of received response.**

## 3.3. Programmed logic for lightweight algorithm and implementation on unity XR

In this implementation stage, a lightweight algorithm to establish a logic to determine for a key press is designed in Arduino UNO and passed onto Unity to reflect the changes on the UI. The overview of the flowchart is shown in Fig. 6.

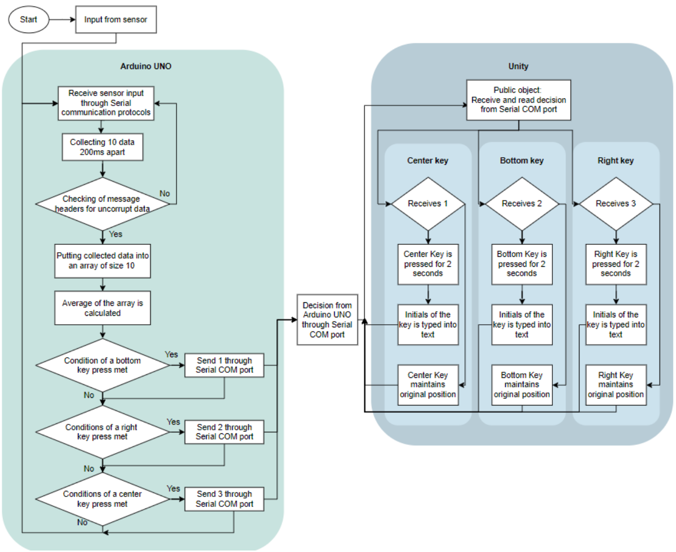



**Fig. 6. Flowchart of implementation.**

The observations from graphical analysis yielded characteristic patterns that could be used to determine for the key presses. These observations are used as thresholds to design conditions for their respective key presses, as such. A centre key press holds the characteristic of having an average static distance of 0.2 to 0.5. A bottom key press is characterized by having an approaching moving speed of more than 0. While a right key is characterized with static distance of more than 2 and dynamic value of more than 150.

The received data is also filtered to check for data hygiene with their header messages that can determine if the set of protocol is corrupted. To have better accuracy, the protocol of data is parsed, and each value is put into an array of size 10 to obtain the average over 1s with an interval of 200 ms per entry.

The decision of a key press is sent to Unity for XR implementation through the serial COM port from the Arduino to the laptop. Unity receives the serial data from a public object, and this is picked up by individual centre, bottom, and right keys to implement for any visual changes, if any. The implementation of a key press on XR shows the key being pressed for 2s, and the symbol of the key is also typed into the text, this is shown in Fig. 7.

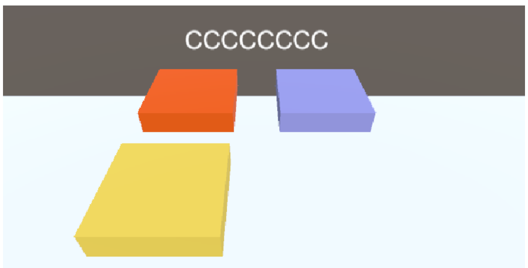



**Fig. 7. Unity interface.**

## 3.4. Measuring accuracy and effectiveness

The accuracy of this system is studied through designing a set of tests with different objectives to measure the number of keys pressed correctly.

The tests comprise of testing each key (centre C, bottom B, and right R), a mixture of centre and bottom, centre and right, and centre to bottom to centre to right. Each test is set to run with 10 keys. The tests are conducted in a few repetitions and the results are taken to compare for the rate of accuracy.

# 4. Results and Discussion

## 4.1. Observation from graphical analysis

Total of 9 different gestures are performed, which consists of 3 without movement, no obstacles (straight hands), all fingers in resting position, index finger in resting position only. As well as 3 movements of up down, front back, and left right, with all fingers moving and only index finger moving, namely, all fingers moving in typing motion, index finger moving in typing motion, all fingers moving front and back, index finger moving front and back, all fingers moving left and right, and index finger moving left and right.

The available data from the sensor gives information about the Existence Energy Value, Static Distance Range, Motion Energy Value, Motion Distance Range, and Motion Speed Range. As it is a self-contained system, the data is pre-processed by the sensor's MCU and returned in their data protocol codes. The data protocol is specified in Table 2.

**Table 2. Description of returned data protocols.**

| Data | Value | Description |
|---|---|---|
| **Existence Energy Value** | 0 – 250 | Feedback of micro-motion noise value in the environment at all times. |
| **Static Distance Range** | 0×01 – 0×06 | The straight-line distance between the micro-motion area in the environment and the sensor. |
| **Motion Energy Value** | 0 – 250 | Provide feedback on the constant motion noise in the environment. |
| **Motion Distance Range** | 0×01 – 0×08 | The straight-line distance between the motion location in the environment and the sensor. |
| **Motion Speed Range** | 0×01 – 0×14 | Real-time judgment of the speed of the moving target.<br>The speed is positive (0×01-0×09) when approaching the radar and negative (0×0b-0×14) when moving away. When there is no motion speed, the value is 0×0a (0m/s). The speed level progresses in 0.5m/s increments, such as 0×0b is 0+0.5m/s; 0×09 is 0-0.5m/s. |

## 4.2. Motion energy value is high when movement is present

Figure 8 shows the Motion Energy Value of all sets of data. It can be observed that those sets that are stationary, no obstacles, resting position, and index finger pressing down, has lower motion energy level, while those moving sets have higher motion energy levels. For better contrast, Figs. 9(a) and 9(b) separates these two graphs.

In Fig. 9(a), the motion energy value saturates at the bottom, while in Fig. 9(b), the motion energy value is high. From this observation, the characteristic of motion energy value can be used to determine whether a key is pressed. However, the anomaly result is significant, hence, background noise should be filtered, and a number of samples should be taken before determining a key press to minimize the rate of errors.

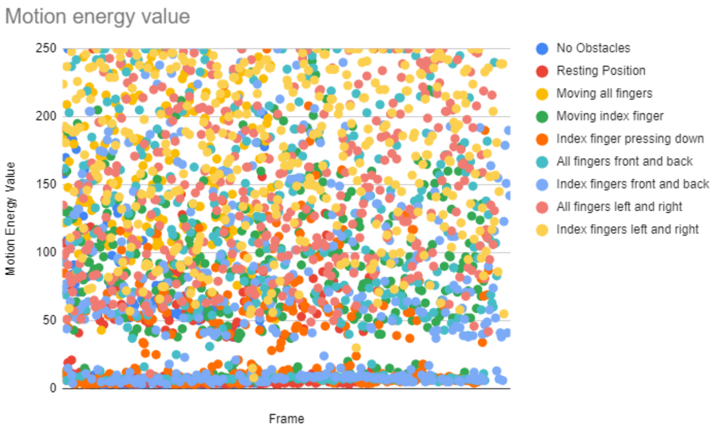
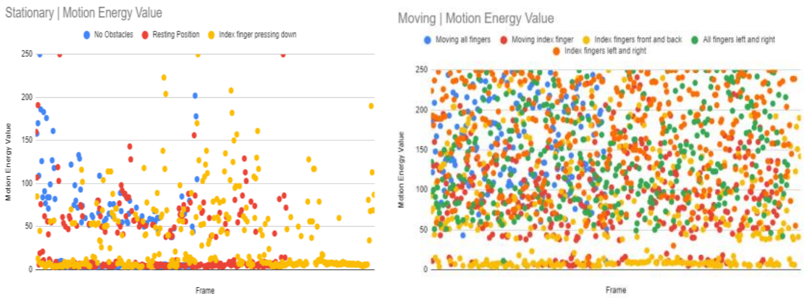



**Fig. 8. Motion energy level of all sets of data.**




**(a)** **(b)**

**Fig. 9. (a) Stationary motion energy value with lower values;**
**(b) Moving motion energy value with higher values.**

## 4.3. Motion speed range indicates a forward or backward key press

Figure 10 shows the Motion Speed Range graph for all datasets. The protocol states that at 10, it indicates for the hex code 0×0 A that indicates there is no change in speed, there is no approaching or departing objects in range. A saturation of data is focused on 0×0 A, with a few scattered anomalies outside of it, and a significant amount of data contributed by Moving all fingers front and back and Moving index finger front and back. For better contrast, Fig. 11 shows the Motion Speed Range graph for All fingers moving front and back while Fig. 12 shows the graph for tapping movement of all fingers. This pattern is consistent with the other graphs where only the front and back movement trigger the readings.

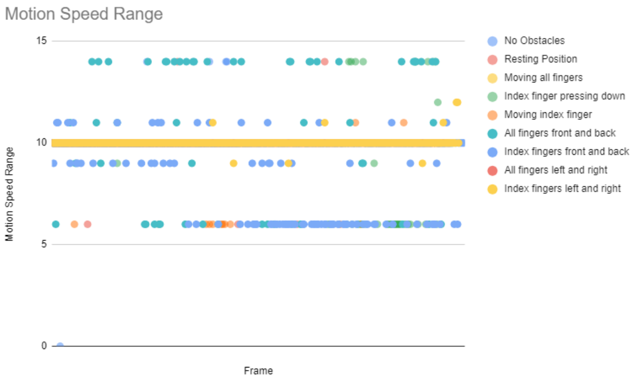



**Fig. 10. Motion speed range for all datasets showing significance for front and back movements, while other data saturates at no speed.**

This observation allows for setting a criterion to determine if a key press in the forward or backward direction is indicated, through comparing the motion speed range of the returned data by the sensor. Similarly, anomalies are observed as background noise are picked up. These should be filtered away, and more data should be sampled to ensure a robust criterion in determining a key press in the forward or backward direction.

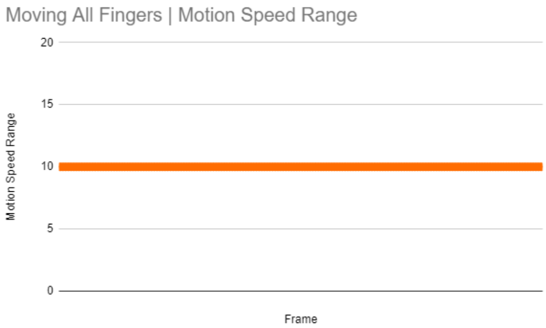



**Fig. 11. Moving all fingers in tapping motion (up and down), showing stationary data.**

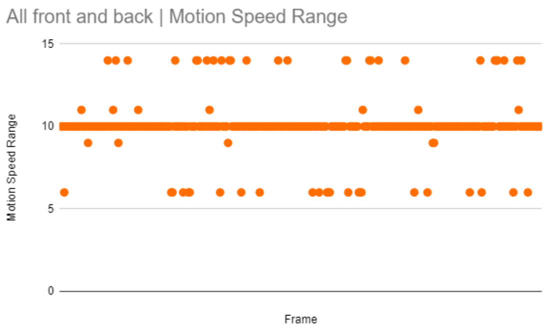
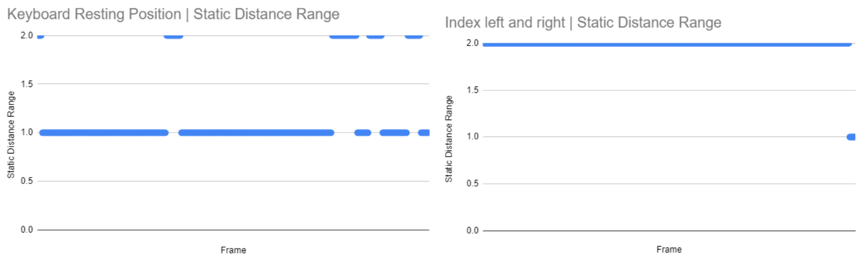



**Fig. 12. Moving all fingers forward and backward, showing a range of data indicating speed.**

### 4.4. Left and right characteristics through static distance range and motion energy value

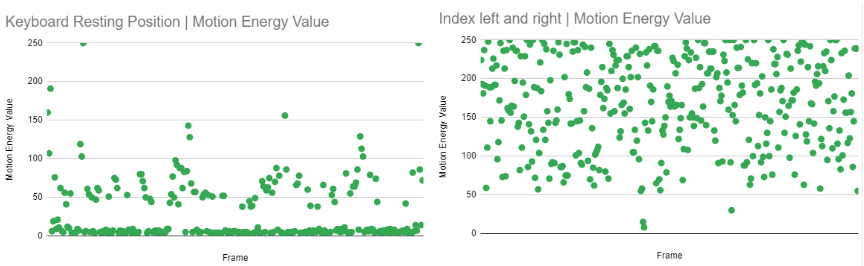
Determining left and right motions require a combination of identification, which come from static distance range and motion energy value. Figures 13(a) and (b) compares the contrast of the Static Distance Range between the fingers in keyboard resting position and while the index finger moves left to right.




**(a)** **(b)**

**Fig. 13. (a) Static distance range of keyboard resting position; (b) Static distance range of moving index finger left to right.**

Figures 14(a) and (b) shows the contrast between the motion energy value of the keyboard resting position against the movement of the index finger tapping left and right. It is observed that the left and right movement graph in Fig. 13(b) shows energy value at higher ranges while in resting position, it saturates at the bottom. With a combination of checking for these two criteria, the motion of left and right movement can be detected. However, more thorough observation and tests have to be done to determine if it is going to the left or to the right.




**(a)** **(b)**

**Fig. 14. (a) Motion energy value of keyboard resting position; (b) Motion energy value of moving index finger left and right.**

### 4.5. Accuracy and effectiveness

The accuracy of the mmWave radar-based gesture recognition system for text input in XR environments was assessed through a series of structured tests designed to measure the correctness of key presses. Each test targeted different key positions and combinations, evaluating the system's ability to accurately detect and interpret finger movements.

The tests involved pressing individual keys-centre (C), bottom (B), and right (R)-as well as combinations of keys, such as centre to bottom (C-B), centre to right (C-R), and a sequence of centre to bottom to centre to right (C-B-C-R). Each scenario was tested with 10 key presses, and the tests were repeated multiple times to gather sufficient data for accuracy analysis, this is shown in Table 3.

According to accuracy results in Table 3, the average accuracy across all tests was calculated to be 62.78%, highlighting the system's varying performance based on the complexity and type of gesture. These findings suggest that while the system shows promise for recognizing basic gestures and key presses, further refinement and optimization are needed to improve accuracy, particularly for more complex gesture sequences as discussed in next section.

**Table 3. Accuracy test.**

| Objective | Sample | Test 1 | Test 2 | Test 3 | Accuracy |
|---|---|---|---|---|---|
| **Centre Key** | CCCCCCCCCC | 70% | 90% | 80% | 80.00% |
| **Bottom Key** | BBBBBBBBBB | 70% | 40% | 60% | 56.67% |
| **Right Key** | RRRRRRRRR | 40% | 70% | 80% | 63.33% |
| **Centre and Bottom** | CBCBCBCBCB | 40% | 50% | 70% | 53.33% |
| **Centre and right** | CRCRCRCRCR | 80% | 60% | 60% | 66.67% |
| **Centre to bottom to centre right** | CBCRCBCRCBCR | 60% | 60% | 50% | 56.67% |

## 5. Conclusions

This research aimed to explore the application of mmWave radar technology for gesture recognition in Extended Reality (XR) typing applications. Through the analysis of various signal parameters provided by the Seeed Studio's MR24HCP1 sensor, significant inferences were made regarding the potential of this technology for XR text input.

The results demonstrated that the motion energy value and motion speed range are reliable indicators for detecting key presses and directional movements. However, the study also identified several challenges, such as the presence of background noise, data anomalies, and limitations imposed by the sensor's predefined data ranges. These challenges highlighted the need for advanced filtering techniques and more sophisticated programming logic to enhance accuracy and robustness.

The accuracy tests conducted revealed varying levels of success, with individual key detection accuracy ranging from 56.67% to 80%. The combined key tests showed an average accuracy of 62.78%, indicating room for improvement. Future work will focus on addressing these challenges by enhancing the filtering mechanisms, refining the gesture recognition algorithms, and potentially exploring sensors with access to raw data for more detailed analysis.

In conclusion, this study successfully demonstrated the feasibility of using mmWave radar sensors for gesture-based typing in XR environments. The insights gained from this research contribute to the development of more intuitive and immersive text input methods in XR, paving the way for further advancements in human-computer interaction through the use of radar-based gesture recognition technology.

## 6. Future Improvements

While a proof-of-concept has been developed, the accuracy of key presses is not ideal due to some limitations and has room for improvement.

Limitations of the sensor include background noise being picked up, which highlighted the need for more sophisticated filtering mechanisms to mitigate background noise and anomalies. Additionally, the predefined ranges of the MR24HCP1 sensor's data protocols limited the granularity of gesture recognition, suggesting that future research might benefit from sensors that allow access to raw data for more detailed analysis.

Faulty data, background interference, and overlapping threshold conditions also posted significant challenges. These issues underscore the importance of not just hardware upgrading, but also carefully curated programming logic and extensive data sampling to ensure accurate gesture detection.

In summary, this project demonstrated the feasibility of using mmWave radar sensors for XR typing applications, paving the way for future advancements in gesture recognition technology. The findings contribute to the ongoing efforts to revolutionize text input methods in XR, aiming to provide a more immersive and intuitive user experience.

**Abbreviations**

| | |
|---|---|
| ADC | Analog-to-Digital |
| AR | Augmented Reality |
| CDIO | Conceive Design Implement Operate |
| CSI | Channel State Information |
| DAI | Doppler Angle Image |
| FFT | Fast Fourier Transform |
| FMCW | Frequency Modulated Continuous Wave |
| FYP | Final Year Project |
| HCI | Human-Computer Interaction |
| IF | Intermediate Frequency |
| mmWaves | millimetre waves |
| MR | Mixed Reality |
| RAI | Range Angle Image |
| RDI | Range Doppler Image |
| RFID | Radio-frequency identification |
| VR | Virtual Reality |
| XR | Extended Reality |

## References

1. Arafat, A.A.; Guo, Z.; and Awad, A. (2021). VR-Spy: A side-channel attack on virtual key-logging in VR headsets. *Proceedings of the* 2021 *IEEE Virtual Reality and* 3D *User Interfaces* (*VR*), Lisboa, Portugal.

2. Collins, B. (2023). Can these AR glasses replace screens? Retrieved October 4, 2023, from https://www.techfinitive.com/ar-glasses-replace-computer-screens/#:~:text=The%20Spacetop%20is%20effectively%20a,your%20web%20browser%20on%20another.
3. Magnopus. (2021). The logitech K830 keyboard and typing in VR. Retrieved October 3, 2023, from https://medium.com/xrlo-extended-reality-lowdown/the-logitech-k830-keyboard-and-typing-in-vr-556e2740c48d.
4. Heaney, D. (2023). Meta is researching turning any flat surface into a virtual keyboard. Retrieved October 4, 2023, from https://www.uploadvr.com/meta-research-any-flat-surface-virtual-keyboard/.
5. Ultraleap. (2025). We're on a mission to make digital worlds feel more human. Retrieved October 4, 2023, from https://www.ultraleap.com/
6. Thariq Ahmed, H.F.; Ahmed, H.; and C.V., A. (2020). Device free human gesture recognition using Wi-Fi CSI: A survey. *Engineering Applications of Artificial Intelligence*, 87, 103281.
7. Liu, Y.; Zhang, S.; Gowda, M.; and Nelakuditi, S. (2022). Leveraging the properties of mmWave signals for 3D finger motion tracking for interactive IoT applications. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 6(3), 52.
8. TapXR. (2025). Your keyboard, mouse & handheld controller. Retrieved October 4, 2023, from https://www.tapwithus.com/
9. Wöbbeking, J. (2022). VR/AR wristband TapXR turns any surface into a keyboard. Retrieved October 4, 2023, from https://mixed-news.com/en/vr-ar-wristband-tapxr-turns-any-surface-into-a-keyboard/
10. Fink, C. (2021). New TapXR bracelet enables typing without a keyboard. Retrieved October 4, 2023, from https://www.forbes.com/sites/charliefink/2021/10/12/tapxr-bracelet-enables-typing-without-a-keyboard/?sh=324485002d7f.
11. Thariq Ahmed, H.F.; Ahmad, H.; and C.V., A. (2020). Device free human gesture recognition using Wi-Fi CSI: A survey. *Engineering Applications of Artificial Intelligence*, 87, 103281.
12. Reifinger, S.; Wallhoff, F.; Ablassmeier, M.; Poitschke, T.; and Rigoll, G. (2007). Static and dynamic hand-gesture recognition for augmented reality applications. *Proceedings of the* 12th *International Conference on Human-computer Interaction*: *Intelligent Multimodal Interaction Environments*, *Beijing China*.
13. Tan, S.; and Yang, J. (2016). WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition. *Proceedings of the* 17th *ACM International Symposium on Mobile Ad Hoc Networking and Computing*, Paderborn Germany.
14. Pu, Q.; Gupta, S.; Gollakota, S.; and Patel, S. (2013). Whole-home gesture recognition using wireless signals. *Proceedings of the* 19th *annual international conference on Mobile computing & networking-MobiCom'*13, Miami, Florida, USA.
15. Yan, B.; Wang, P.; Du, L.; Chen, X.; Fang, Z.; and Wu, Y. (2023). mmGesture: Semi-supervised gesture recognition system using mmWave radar. *Expert Systems with Applications*, 213(Part B), 119042.

16. Sun, L.; Sen, S.; Koutsonikolas, D.; and Kim, K.-H. (2015). Widraw: Enabling hands-free drawing in the air on commodity WiFi devices. *Proceedings of the* 21st *Annual International Conference on Mobile Computing and Networking*, Paris France.
17. Li, M.; Meng, Y.; Liu, J.; Zhu, H.; Liang, X.; Liu, Y.; and Ruan, N. (2016). When CSI meets public WiFi: Inferring your mobile phone. *Proceedings of the* 2016 *ACM SIGSAC Conference on Computer and Communications Security*, Vienna Austria.
18. Saha, S.K. et al. (2017). X60: A programmable testbed for wideband 60 GHz Wlans with phased arrays. *Proceedings of the* 11th *Workshop on Wireless Network Testbeds*, *Experimental evaluation & CHaracterization*, Snowbird Utah USA, 75-82.
19. Bjorklund, S.; Johansson, T.; and Petersson, H. (2012). Evaluation of a micro-Doppler classification method on mm-wave data. *Proceedings of the* 2012 *IEEE Radar Conference*, Atlanta, GA, USA.
20. Li, Z.; Lei, Z.; Yan, A.; Solovey, E.; and Pahlavan, K. (2020). ThuMouse: A micro-gesture cursor input through mmWave radar-based interaction. *Proceedings of the* 2020 *IEEE International Conference on Consumer Electronics* (*ICCE*), Las Vegas, NV, USA.
21. Sakran, F.; Neve-Oz, Y.; Ron, A.; Golosovsky, M.; Davidov, D.; and Frenkel, A. (2008). Absorbing frequency-selective-surface for the mm-wave range. *IEEE Transactions on Antennas and Propagation*, 56(8), 2649-2655.
22. Olick-Gibson, J.; Cai, B.; Zhou, S.; Mutic, S.; Carter, P.; Hugo, G.; and Zhang, T. (2020). Feasibility study of surface motion tracking with millimeter wave technology during radiotherapy. *Medical Physics*, 47(3), 1229-1237.
23. Jhaung, Y.-C.; Lin, Y.-M.; Zha, C.; Leu, J.S.; and Köppen, M. (2022). Implementing a hand gesture recognition system based on range-doppler map. *Sensors*, 22(11), 4260.
24. Hajika, R.; Gunasekaran, T.S.; Nassani, A.; Pai, Y.S.; and Billinghurst, M. (2021).VRTwitch: Enabling micro-motions in VR with radar sensing. *Proceedings of the SIGGRAPH Asia* 2021 *Posters*, Tokyo Japan.
25. Unity Documentation. (2025). Retrieved October 4, 2023, from https://docs.unity3d.com/Manual/ScriptingSection.html