

Towards Hand Gesture Recognition Prototype Using the IWR6843ISK Radar Sensor and Leap Motion

Nuwan T. Attygalle^{1,*†}, Una Vuletic¹, Matjaž Kljun^{1,3} and Klen Čopič Pucihar^{1,2,3}

¹University of Primorska, Faculty of Mathematics, Natural Sciences and Information Technologies, Koper, Slovenia

²Faculty of Information Studies, Novo Mesto, Slovenia

³Stellenbosch University, Department of Information Science, Stellenbosch, South Africa

Abstract

Gesture recognition with millimetre wave radar has been extensively researched and many datasets are publicly available. However, datasets with raw data (including voltage levels) are almost not available. These are essential as they allow for recreation of all signal representations, thus allowing evaluation of gesture detection pipelines on the same datasets. This paper presents the design, implementation, and testing of a data recording prototype for hand gesture recognition coupling the Texas Instruments IWR6843ISK radar sensor and magic leap motion sensor. The study emphasizes on the system architecture, system apparatus and implementation to optimize recorded data quality and format flexibility (including voltage levels).

Preliminary results demonstrate the prototype's ability to accurately capture stream of data from both radar sensor and Ultraleap LeapMotion sensor. This report assists researchers and other practitioners in building such a system, thus allowing them to capture high quality datasets on their own.

1. Introduction and related work

In recent years, gesture recognition with millimetre-wave (mmWave) radars has garnered significant interest among researchers [1, 2, 3, 4, 5, 6, 7, 8, 9] and practitioners due to its numerous advantages, including resilience to environmental factors such as lighting and weather conditions, low power consumption, compact form factor, and the ability to detect gestures through materials [8, 10]. This capability is particularly valuable for real-world applications such as controlling infotainment systems in vehicles or sensing gestures even when the sensor is embedded within an object, such as a phone in a pocket.

Current gesture recognition datasets with mmWave radars often provide processed data in the form of Range-Doppler, Range-Angle, or point-cloud representations. While these formats offer certain benefits, they lack the flexibility and granularity of raw Voltage data, which are essential for understanding and comparing different gesture recognition pipelines. Further still these datasets predominantly only offer one sensing modality, thus failing to provide reference data

HCI SI 2023: Human-Computer Interaction Slovenia 2023, January 26, 2024, Maribor, Slovenia

*Corresponding author.

†These authors contributed equally.

✉ nuwan.attygalle@famnit.upr.si (N. T. Attygalle); una.vuletic@famnit.upr.si (U. Vuletic);

matjaz.kljun@famnit.upr.si (M. Kljun); klen.copic@famnit.upr.si (K. Čopič Pucihar)

🆔 0000-0001-8498-1923 (N. T. Attygalle); 0000-0002-6988-3046 (M. Kljun); 0000-0002-7784-1356 (K. Čopič Pucihar)

© 2024 Copyright for this paper by its authors. Use permitted under Creative Commons License Attribution 4.0 International (CC BY 4.0).

CEUR Workshop Proceedings (CEUR-WS.org)

streams. For example, radar sensor can be used to reconstruct point-cloud data, however to the best of our knowledge no dataset exists where reference point-cloud data (ground truth) would be available besides the radar data. To address this gap, we present the design, implementation, and testing of a data recording prototype for hand gesture recognition coupling the Texas Instruments (TI) IWR6843ISK radar sensor and magic leap motion sensor.

Even though Texas Instruments provides extensive documentation and user guides for configuring the IWR6843ISK radar sensor with the DCA1000EVM data capture card, the process is not always straightforward. This is especially true when users need to develop their own applications and customize the sensor to suit specific use cases. Moreover, integrating multiple sensors together can be challenging. In this paper, we delve into the process of configuring the radar sensor and integrating it with the LeapMotion sensor to achieve robust data capture. We also explore the difficulties we encountered during the prototype setup and how we addressed them, providing insights into our decision-making process.

2. System architecture

This data capture tool incorporates two sensors: IWR6843ISK radar sensor¹ and an Ultraleap LeapMotion sensor². The tool captures mid-air gesture trajectories in real-time from both sensors. The radar sensor operates at a frame rate of 100 frames per second (fps), while the LeapMotion sensor operates at a frame rate of 115 fps. Both sensors are triggered simultaneously via two sockets, and the timestamps of each frame are logged. This allows the tool to align the corresponding radar frames with the LeapMotion hand captures.

2.1. Apparatus

Figure 1 shows a Radome like protective enclosure for the radar sensor. The enclosure is constructed of Plexiglass covered with thin aluminum foil to prevent radiation leakage. An open square on top of the antenna array allows the material under investigation to be placed on the enclosure. The size of the square is determined based on the radar's field of view. The thickness and proximity of the material to the radar sensor are important for optimal performance, but we used the materials in their existing state to focus on analyzing their impact on gesture recognition. The distance between the material's internal surface and the radar antenna is also critical and was set up 12mm.

2.2. User interface

The user interface (UI) of the application was implemented using the React³ JavaScript library and pre-built semantic components from the Semantic UI⁴ front-end development framework. React's component-based approach and Semantic UI's intuitive and reusable components made it

¹<https://www.ti.com/tool/IWR6843ISK>

²<https://leap2.ultraleap.com/leap-motion-controller-2/>

³<https://react.dev/>

⁴<https://react.semantic-ui.com/>

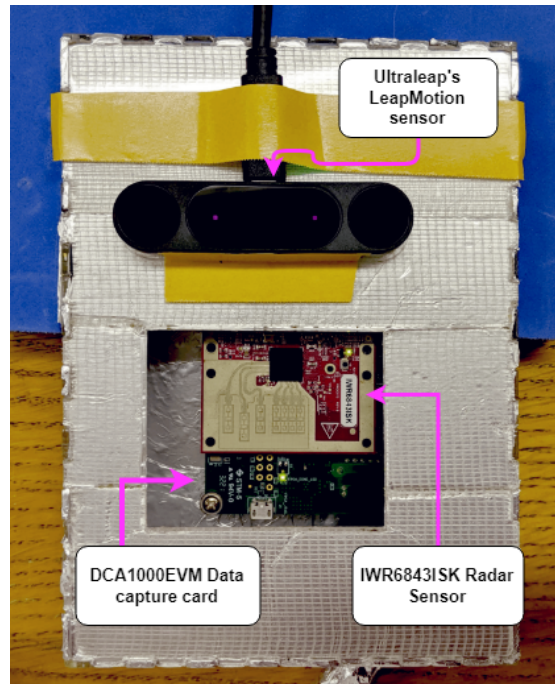


Figure 1: Apparatus: Materials are placed on the open space configured to uphold the radar's Field of View at 140 degrees azimuth and 120 degrees elevation. The Radar sensor and Data capture card are housed within the Radome, with the Leap Motion sensor positioned on its top.

an ideal choice for building a user-friendly and visually appealing interface. React's component-based structure allowed for modular development, ensuring that UI elements could be easily reused across the application. This approach simplified code maintenance and contributed to a maintainable and scalable codebase. Semantic UI's pre-built semantic components provided a consistent and modern design language, ensuring that the UI elements were well-organized, accessible, and visually appealing. This library helped to streamline the design process and ensured that the application's UI conformed to industry standards. The results is an interface shown in Figure 2

2.3. Backend

The backend was developed using Node.js⁵, Express.js⁶, and Atlas MongoDB cloud database⁷. The database stores gesture metadata and metadata about recordings such as number of frames captured, number of zero filled frames, number of Bytes recieved, and recorded temperatures of the antennas. The Express.js routing framework was utilized to efficiently handle incoming user requests and data from the sensors. Locally running backend was responsible for communicating with both the radar sensor and the LeapMotion sensor via sockets to acquire and store gesture

⁵<https://nodejs.org/en>

⁶<https://expressjs.com/>

⁷<https://www.mongodb.com/atlas/database>

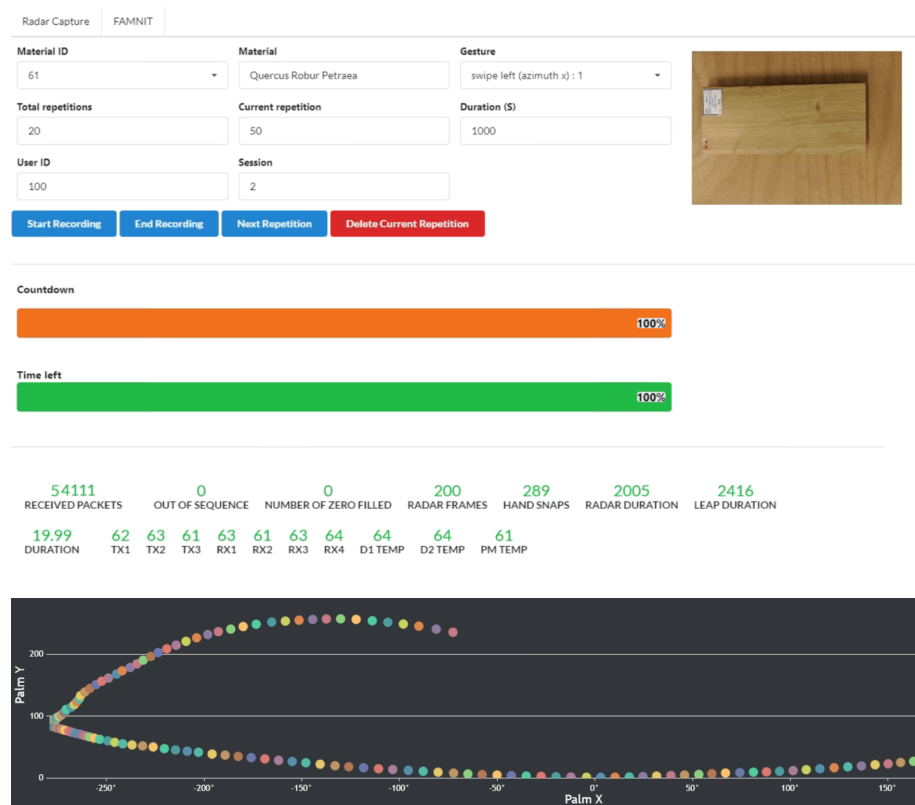


Figure 2: User Interface: Once the study coordinator presses start-gesture then countdown progress starts and the participant waits at the start position to start the gesture. Once the Time-Left progress bar start, the participant start performing the gesture

data in the Atlas MongoDB database and local disk drive. The asynchronous and non-blocking I/O capabilities of Node.js, coupled with the efficient routing capabilities of Express.js, enabled the backend to handle a high volume of data streams from the sensors with minimal latency. The NoSQL structure of Atlas MongoDB provided flexibility in storing and retrieving gesture metadata, making it suitable for the dynamic nature of mid-air gesture data.

2.4. Radar sensor controller

In our research, we employed the Texas Instruments (TI) IWR6873ISK, an advanced on-antenna radar module, as our primary radar sensor. TI provides the mmWave Studio application, a comprehensive software tool designed for communication and configuration of the radar sensor. We utilized this application for both configuring and triggering the sensor. A notable feature of mmWave Studio is its command-line accessibility, facilitated through built-in Lua shell. This feature allows for the configuration of the radar sensor via a custom Lua script.

The process of sensor activation is initiated through a Lua script within mmWave Studio. This script serves two primary functions: firstly, it enables the detailed configuration of the sensor through a series of specific commands. Once the sensor is configured, the Lua script then

establishes a socket server, actively listening for trigger commands issued from the backend of our application. Upon receiving a trigger command from the backend, the Lua script proceeds to activate the radar sensor.

The operational duration of the radar sensor is set to a precise two-second window upon activation. During this period, the sensor transmits a total of 200 frames. Following the transmission of each frame, the radar sensor relays the raw voltage data of the radar receiving antennas (R_x) to the DCA1000EVM data capture card⁸. This transmission occurs through an LVDS (Low-Voltage Differential Signaling) interface, ensuring high-speed data transfer and integrity. Upon reception, the DCA1000EVM data capture card sends the captured radar data to the computer via an Ethernet port using User Datagram protocol (UDP), ensuring efficient and timely transfer.

2.5. DCA1000EVM data capture card controller

In our setup, the R_x antennas of the radar sensor capture data at a bandwidth exceeding the capabilities of standard UART COM port communication. To address this, Texas Instruments provides the DCA1000EVM data capture card, designed specifically for streaming radar sensor data. This card connects to the radar sensor's LVDS port and transmits the data to a computer via an Ethernet port. While the DCA1000EVM receives operational commands through UART COM ports, the data transfer is executed through Ethernet for higher bandwidth efficiency.

Texas Instruments also offers a Command-Line Interface (CLI) control program for the DCA1000EVM⁹. This program manages the data flow from the capture card, filling in any missing frames with zeros and reordering the data before storing it as binary files. Additionally, it compiles the statistics of the received data into a separate CSV file.

2.6. Ultraleap's LeapMotion controller

In our research, we employed the Ultraleap Leap Motion sensor for tracking hand movements. Ultraleap provides the Gemini software¹⁰, a C#-based SDK, to facilitate the sensor's integration with computers for data capture. To integrate the Leap Motion sensor with our system backend, we developed a custom Leap Motion controller script, leveraging the LeapC Python wrapper¹¹.

The Leap Motion controller initiates a socket server to listen for interruption signals from the backend. Upon receiving an interruption signal, the controller starts the recording of hand events. The Leap Motion sensor transmits the detected hand coordinates as events via USB-C to the computer. These events are then captured by our Leap Motion controller using the LeapC Python wrapper, ensuring seamless and accurate acquisition of hand movement data.

⁸<https://www.ti.com/tool/DCA1000EVM>

⁹https://e2e.ti.com/cfs-file/__key/communityserver-discussions-components-files/1023/TI_DCA1000EVM_CLI_Software_DeveloperGuide.pdf

¹⁰<https://leap2.ultraleap.com/gemini-downloads/>

¹¹<https://github.com/ultraleap/leapc-python-bindings>

3. Evaluation of raw data

Upon acquiring the raw data, our initial step is to evaluate its utility and interpretability. This assessment is conducted by visualising the data using several established representation methods in radar signal processing. Figure 3 shows Range-Doppler[1, 8, 2, 3], Point-Cloud[4, 5], Range-Angle[2, 9], and IQ signal representation[7] deduced from the raw data gathered. These visualisation techniques provide critical insights into the data's characteristics such as detected target range, angle of arrival or velocity of the targets and are instrumental in determining its applicability for further analysis.

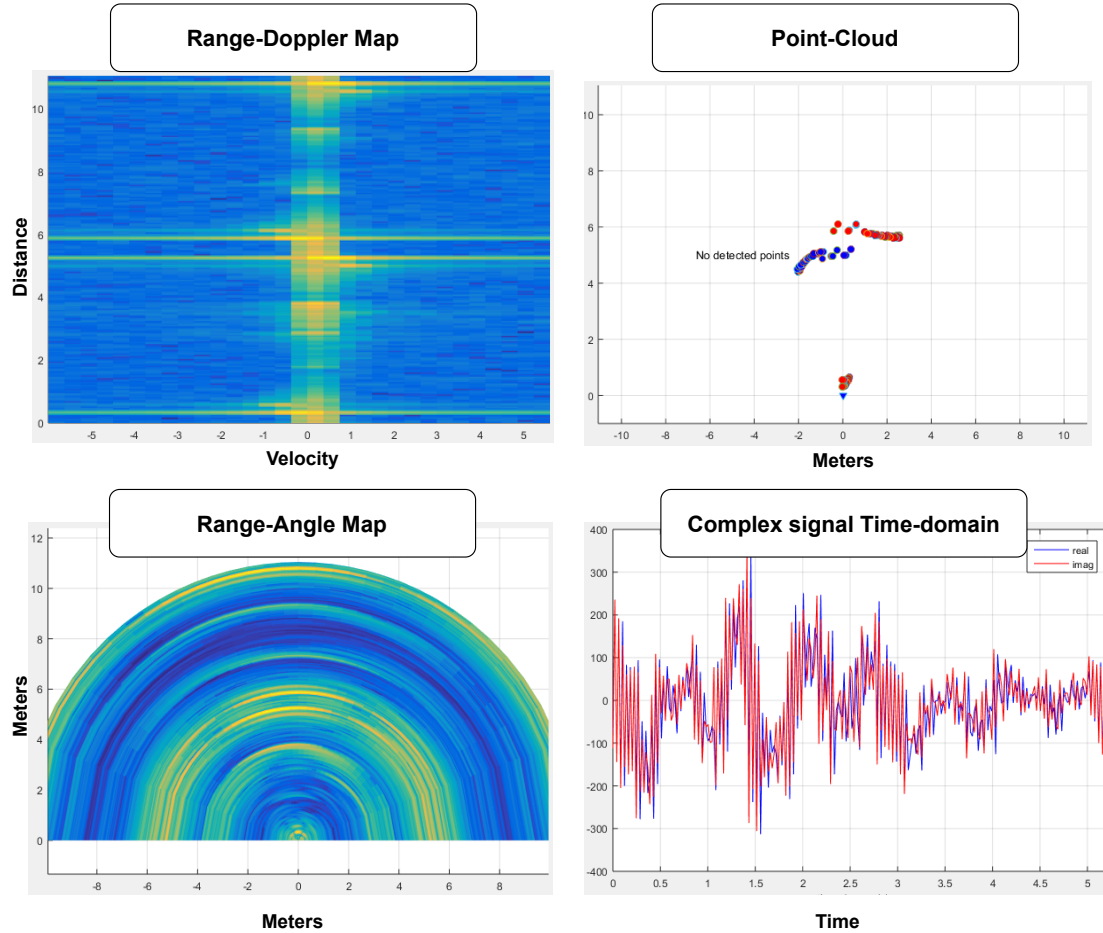


Figure 3: The image presents four signal representation plots derived from raw data for the Swipe-Left gesture's initial repetition: the Range-Doppler Map shows velocity and distance, the Point-Cloud visualises detected points in spatial coordinates, the Range-Angle Map illustrates the distribution of ranges over angles, and the Complex Signal Time-Domain plot displays the real and imaginary components of the signal over time.

4. Triggering the radar Sensor

In our experimental setup, two distinct approaches were employed for triggering the radar sensor. The first approach involves configuring the radar sensor to emit an indefinite number of frames, wherein data recording occurs exclusively during the user's gesture performance. This method eliminates the need to activate the radar sensor for each gesture repetition.

The alternative approach necessitates the activation of the radar sensor concurrently with the commencement of the user's gesture, thereby initiating data recording. This method requires the radar sensor to be invoked for every gesture repetition. However, the comparative advantages of these two approaches – finite frame versus infinite frame configurations – are not immediately apparent. To investigate this, we conducted a focused within-subject user study.

The study's objective was to assess the sensor's efficacy across these two operational modes by monitoring the incidence of data packet loss and variations in sensor temperature.

For a robust evaluation, we conducted a controlled within-subject experiment, maintaining consistent external variables while comparing the frequency of missing data frames and temperature fluctuations between the two sensor operation modes.

4.1. Data capture and Analysis

We selected a set of five commonly used gestures that are relatively simple to execute: Swipe-Left, Swipe-Right, Swipe-Up, Swipe-Down, and Hold[8, 5, 11, 12, 10, 11, 13, 14]. The participant repeated each gesture 10 times in both Infinite and Finite modes. We recorded the antenna temperatures and the number of missing radar frames after each repetition to compare sensor performance between the two modes.

Initially, we set the sensor to Infinite mode, during which the participant completed 50 gestures repetition in total. After each repetition, we recorded the antenna temperatures and the count of any missing data frames.

Subsequently, we powered down the sensor for one hour to allow it to return to baseline temperature conditions. Following this cool-down period, we reconfigured the sensor to Finite mode, ensuring that startup temperatures were consistent with the initial trials.

4.2. Results

We analysed the mean antenna temperatures and number of missing radar frames in two sensor configurations. Since the mean antenna temperatures ($W = 0.958, p < 0.01$) and number of missing radar frames ($W = 0.53375, p < 0.00001$) were not normally distributed, we opted for Wilcoxon matched pairs [15] test with false discovery rate (FDR)[16] correction.

The mean temperatures and number of missing frames across study conditions are shown in Figure 4. Significant effect could be detected between *Infinite* and *Finite* sensor configurations when analysing the distribution of the mean temperatures. ($V = 0, p < 0.0001$) Similarly, significant effect could be detected between the same same sensor configurations when analysing the distributions of the number of missing radar frames. ($V = 229, p < 0.05$)

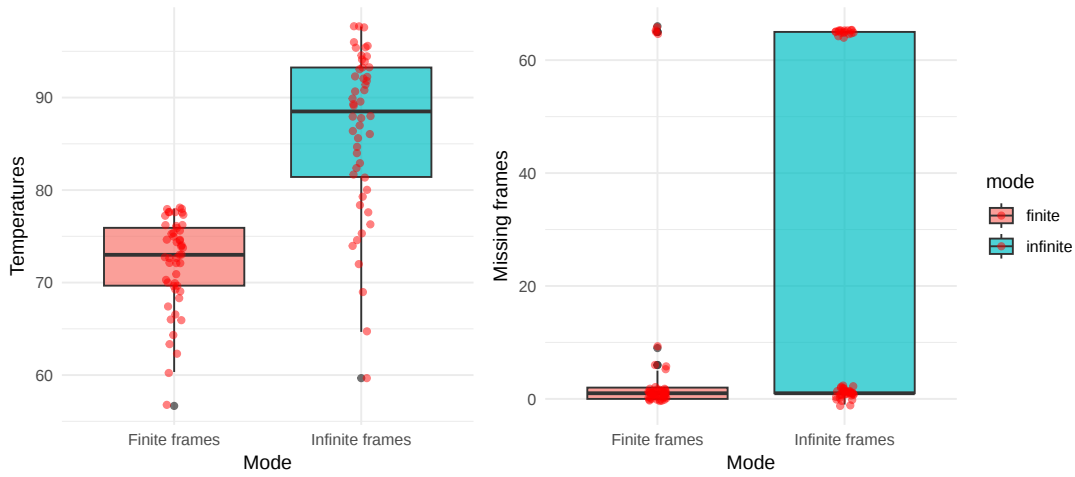


Figure 4: Comparison of Average Temperatures and Missing frames between Finite and Infinite Radar Configurations

5. Discussion

The evaluation demonstrates that the sensor effectively captures the raw voltage signal, and the acquired data can be successfully interpreted using standard radar signal processing techniques.

5.1. Controlling Temperature

Our initial setup involved setting the radar frame frequency to 100 fps, with the radar continuously radiating frames until manually stopped. This approach, however, led to a significant rise in sensor temperature, reaching approximately 100°C within 30 minutes of activation. To mitigate this issue and maintain temperature control, we modified the radar operation to activate only during gesture execution. The radar now radiates frames for a precise two-second duration aligned with gesture performance, followed by deactivation. This intermittent activation effectively regulates the sensor temperature. Moreover, we observed that at temperatures exceeding 90°C, there was a notable increase in data frame loss. This error was substantially reduced with our revised method of operation. [section 4](#) shows that the number of missing radar frames and the antenna temperatures are significantly different in the two sensor configurations. However, we should acknowledge that the findings are based on a single-participant study.

5.2. Signal processing on inbuilt DSP chip of IWR6843ISK Radar

The TI Radar Toolbox provides various applications such as Out-of-Box-Demo¹², Area-Scanner¹³, and Gesture-Recognition¹⁴, designed for deployment within the IWR6843ISK radar sensor. These applications leverage the radar's inbuilt DSP chip for signal processing, outputting data formats such as point cloud, Range-Doppler, Angle-Doppler, or recognized gestures and objects via UART COM ports. Additionally, by setting 'lvdsStreamCfg -1 1 1 1' in the configuration file, the sensor can be configured to output raw Voltage data through the LVDS interface.

However, this approach presents limitations when operating at higher frame rates, such as 100 fps, rendering it unsuitable for our specific needs. Therefore, in our study, we opt for processing the signal externally on a computer, post capturing the raw Voltage data. This method not only circumvents the frame rate limitation but also offers greater flexibility in data analysis and processing.

6. Conclusion

This study presents the development of a prototype system designed for capturing raw voltage data from the IWR6843ISK radar sensor during hand gesture performance. In our innovative approach, we integrated two distinct sensors: the IWR6843ISK radar sensor and Ultraleap's Leap Motion sensor. We developed a specialised application capable of simultaneously recording both the radar's raw voltage data and the Leap Motion sensor's hand snapshots during gesture execution.

Furthermore, we delved into the operational dynamics of the sensor, examining how different configurations affect antenna temperatures and the occurrence of missing radar frames. This exploration is crucial for optimising sensor performance and ensuring reliable data capture.

However, our research also encountered challenges, particularly in achieving synchronisation when capturing data from multiple sensors. Addressing these challenges is critical for enhancing the accuracy and efficiency of the data collection process. This study not only contributes valuable knowledge to the field of gesture recognition using radar technology but also opens avenues for future research in improving multi-sensor integration and synchronisation.

Acknowledgments

We acknowledge support from Slovenian Research Agency, grant number P1-0383, P5-0433 and J5-1796 and support from research program CogniCom (0013103) at the University of Primorska.

¹²https://dev.ti.com/tirex/explore/node?node=A__AJwREK3NVquf175i533giA__radar_toolbox__1AslXXD__LATEST

¹³https://dev.ti.com/tirex/explore/node?node=A__AOxzy5SR.-h2AygLnSknw__radar_toolbox__1AslXXD__LATEST

¹⁴https://dev.ti.com/tirex/explore/node?node=A__ABBJTJ7Y0H4LgRlsbV2S7g__radar_toolbox__1AslXXD__LATEST

References

- [1] N. T. Attygalle, L. A. Leiva, M. Kljun, C. Sandor, A. Plopski, H. Kato, K. Čopič Pucihar, No interface, no problem: gesture recognition on physical objects using radar sensing, *Sensors* 21 (2021) 5771.
- [2] S. Hazra, A. Santra, Robust gesture recognition using millimetric-wave radar system, *IEEE sensors letters* 2 (2018) 1–4.
- [3] H. R. Lee, J. Park, Y.-J. Suh, Improving classification accuracy of hand gesture recognition based on 60 ghz fmcw radar with deep learning domain adaptation, *Electronics* 9 (2020) 2140.
- [4] H. Liu, Y. Wang, A. Zhou, H. He, W. Wang, K. Wang, P. Pan, Y. Lu, L. Liu, H. Ma, Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing, *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 4 (2020) 1–28.
- [5] S. Palipana, D. Salami, L. A. Leiva, S. Sigg, Pantomime: Mid-air gesture recognition with sparse millimeter-wave radar point clouds, *Proceedings of the ACM on interactive, mobile, wearable and ubiquitous technologies* 5 (2021) 1–27.
- [6] Z. Qamar, N. Aboserwal, J. L. Salazar-Cerreno, An accurate method for designing, characterizing, and testing a multi-layer radome for mm-wave applications, *IEEE Access* 8 (2020) 23041–23053.
- [7] T. Sakamoto, X. Gao, E. Yavari, A. Rahman, O. Boric-Lubecke, V. M. Lubecke, Hand gesture recognition using a radar echo i–q plot and a convolutional neural network, *IEEE sensors letters* 2 (2018) 1–4.
- [8] K. Čopič Pucihar, N. T. Attygalle, M. Kljun, C. Sandor, L. A. Leiva, Solids on soli: Millimetre-wave radar sensing through materials, *Proceedings of the ACM on Human-Computer Interaction* 6 (2022) 1–19.
- [9] L. Zheng, J. Bai, X. Zhu, L. Huang, C. Shan, Q. Wu, L. Zhang, Dynamic hand gesture recognition in in-vehicle environment based on fmcw radar and transformer, *Sensors* 21 (2021) 6368.
- [10] A. Sluÿters, S. Lambot, J. Vanderdonckt, S. Villarreal-Narvaez, Analysis of user-defined radar-based hand gestures sensed through multiple materials, *IEEE Access* 11 (2023) 1–24. doi:10.1109/ACCESS.2023.0322000.
- [11] A.-I. Şiean, C. Pamparău, A. Sluÿters, R.-D. Vatavu, J. Vanderdonckt, Flexible gesture input with radars: systematic literature review and taxonomy of radar sensing integration in ambient intelligence environments, *Journal of Ambient Intelligence and Humanized Computing* (2023) 1–15.
- [12] H.-S. Choi, W.-S. Shin, D.-H. Bang, Mirror therapy using gesture recognition for upper limb function, neck discomfort, and quality of life after chronic stroke: a single-blind randomized controlled trial, *Medical science monitor: international medical journal of experimental and clinical research* 25 (2019) 3271.
- [13] A. Sluÿters, S. Lambot, J. Vanderdonckt, R.-D. Vatavu, Radarsense: Accurate recognition of mid-air hand gestures with radar sensing and few training examples, *ACM Transactions on Interactive Intelligent Systems* (2023).
- [14] S. Chioccarello, A. Sluÿters, A. Testolin, J. Vanderdonckt, S. Lambot, Forte: Few samples for recognizing hand gestures with a smartphone-attached radar, *Proceedings of the ACM*

on Human-Computer Interaction 7 (2023) 1–25.

- [15] M. Marusteri, V. Bacarea, Comparing groups for statistical differences: how to choose the right statistical test?, *Biochemia medica* 20 (2010) 15–32.
- [16] J. D. Storey, R. Tibshirani, Statistical significance for genomewide studies, *Proceedings of the National Academy of Sciences* 100 (2003) 9440–9445.