sensors

MDPI

Article

Low Complexity Radar Gesture Recognition Using Synthetic Training Data

Yanhua Zhao [1,2,*], Vladica Sark [1], Milos Krstic [1,3] and Eckhard Grass [1,2]

[1] IHP—Leibniz-Institut für Innovative Mikroelektronik, 15236 Frankfurt, Germany
[2] Institute of Computer Science, Humboldt University of Berlin, Rudower Chaussee 25, 12489 Berlin, Germany
[3] Design and Test Methodology, University of Potsdam, August-Bebel-Straße 89, 14482 Potsdam, Germany
* Correspondence: zhao@ihp-microelectronics.com

Abstract: Developments in radio detection and ranging (radar) technology have made hand gesture recognition feasible. In heat map-based gesture recognition, feature images have a large size and require complex neural networks to extract information. Machine learning methods typically require large amounts of data and collecting hand gestures with radar is time- and energy-consuming. Therefore, a low computational complexity algorithm for hand gesture recognition based on a frequency-modulated continuous-wave (FMCW) radar and a synthetic hand gesture feature generator are proposed. In the low computational complexity algorithm, two-dimensional Fast Fourier Transform is implemented on the radar raw data to generate a range-Doppler matrix. After that, background modelling is applied to separate the dynamic object and the static background. Then a bin with the highest magnitude in the range-Doppler matrix is selected to locate the target and obtain its range and velocity. The bins at this location along the dimension of the antenna can be utilised to calculate the angle of the target using Fourier beam steering. In the synthetic generator, the Blender software is used to generate different hand gestures and trajectories and then the range, velocity and angle of targets are extracted directly from the trajectory. The experimental results demonstrate that the average recognition accuracy of the model on the test set can reach 89.13% when the synthetic data are used as the training set and the real data are used as the test set. This indicates that the generation of synthetic data can make a meaningful contribution in the pre-training phase.

Keywords: FMCW radar; gesture sensing; machine learning; mmWave; synthetic features

check for updates

Citation: Zhao, Y.; Sark, V.; Krstic, M.; Grass, E. Low Complexity Radar Gesture Recognition Using Synthetic Training Data. *Sensors* **2023**, *23*, 308. https://doi.org/10.3390/s23010308

Academic Editors: Janusz Dudczyk and Piotr Samczyński

Received: 24 October 2022
Revised: 19 December 2022
Accepted: 25 December 2022
Published: 28 December 2022

1. Introduction

Low-cost, miniaturised radars have become increasingly popular in recent years. This has led to a large number of radar-based applications. For example, automotive radar can play an important role in collision avoidance systems. In addition to this, the potential of the radar in the field of medical applications is also being investigated. Applications such as weather radar and ground-penetrating radar reveal a need for such applications and research on radar technology and algorithms is highly desirable.

Traditional human–computer interaction mediums such as buttons, mice and keyboards are not always convenient in certain situations, such as operations in clean rooms. Contactless human–computer interaction requires less touching and is more hygienic. It can also further enhance the user experience. Hand gestures are an important medium for contactless human–computer interaction [1].

The outstanding privacy-protecting character of radar makes it preferable over cameras and its ability to be unaffected by light conditions is again preferable to LIDAR. The frequency-modulated continuous-wave (FMCW) radar is able to detect the distance, velocity and angle of several objects at the same time; hence, it is employed for hand gesture recognition in our work. FMCW radars can suffer from mutual interference. If there are other radars as sources of interferences, the methods in [2] for finding the range, velocity and angle of the target can be referred to. In this paper, only one radar is employed and interference from other radars is not considered.

Hand gesture recognition based on FMCW radar can be grouped into two categories. The first is based on raw data, meaning that the raw data are fed directly into a neural network to classify the hand gestures. The second is based on features, such as range, velocity and angle. These three features are obtained by certain data pre-processing methods and then combined with machine learning methods to achieve hand gesture classification. In this work, feature-based hand gesture recognition will be considered since the size of the radar raw data is large and will increase the training complexity of the neural network.

1.1. Related Work of Heat Map-Based Recognition

The features of a hand gesture can be represented in the form of a heat map. Many researchers have made various contributions to the field of heat map-based hand gesture recognition. Heat map-based gesture recognition is a relatively common approach and in [3–11] all authors use heat map-based recognition.

In [3], the authors conducted an experiment on gesture sensing using an FMCW radar with a centre frequency of 25 GHz, without considering the recognition of gestures. Later in [4], the authors collected 1714 gesture samples using FMCW radar, which contains ten types of gesture. Then they extracted time-Doppler heat maps from the radar raw data and trained a deep convolutional neural network model. An average recognition rate of 89.1% was achieved. Deep convolutional networks can be very challenging to implement at the hardware level. This is because they have a large number of weights and although it is possible to remove some of the unimportant weights, by pruning and other methods, it does require a lot of computational resources.

In [5], Lien et al. developed a small, low-power radar with a center frequency of 60 GHz. In contrast to [4], a time-Doppler heat map and a time-range heat map were extracted. With a classical random forest classifier, the average recognition accuracy of the four micro-hand gestures was up to 92.10%.

From the raw data of the FMCW radar not only time-distance and time-Doppler heat maps can be extracted, but also a time–angle heat maps. These three types of heat maps are employed as the basis for hand gesture recognition in [9]. The authors adopted background modelling to separate the static background and the moving target so that the feature heat maps were cleaned efficiently. The average recognition rate of six gestures was over 98.93%. The authors employed a pre-trained model, but a large number of weights of the model made implementation on an FPGA problematic.

Similarly, the authors used the multi-stream convolutional neural network (MS CNN) model in [11] to learn features of the dataset for on-air writing recognition. Although the accuracy achieved was very impressive, the drawback, as before, was that the MS CNN would require a lot of hardware resources.

Given the above references, we can state that heat map-based hand gesture recognition can achieve encouraging results, but its disadvantages should not be ignored. Firstly, the process of constructing a heat map is relatively complex and time-consuming. A hand gesture is made up of several frames of data, each of which will form one or more heat maps depending on the combination of features selected. A single hand gesture sample can produce a large number of heat maps, which leads to complex processing. In addition, the hand gesture features in the form of heat maps need to be further fed into a deep convolutional network to extract features, which requires even more computational efforts.

1.2. Motivation for Synthetic Data Generation and Related Work

Machine learning-based hand gesture recognition faces data scarcity issues and only a few open radar datasets are available. Many scientists spend a lot of effort and time collecting data. In Google's Soli project [5], its team collected 5000 samples with 5 participants. For [10], 7200 gesture samples were collected from 20 people. The authors of [6] collected

2750 samples, which involved 11 participants. A total of 1500 and 1200 gesture samples were collected in [9] and [8], respectively.

In [12], the authors utilised a sparse point cloud extraction method and a Message Passing Neural Network (MPNN)-based graphical convolution method for real-time gesture recognition. Despite the reduced computational complexity, all datasets were collected by other teams manually, which involved a lot of time and effort.

Collecting gesture samples with radar is very challenging. It is needed to perform specific hand gestures repetitively, which is time-consuming and labour-intensive. An efficient gesture feature synthesiser would be highly beneficial.

Human motion simulation is a good start for synthetic gestures. Some researchers have already combined human motion simulation with radar sensors. The authors have developed a human walking model in [13]. Afterwards, in [14], this model was employed to construct micro-Doppler spectrograms for gait analysis.

Reference [15] proposed a radar data synthesis process for four hand gestures. The authors used the 3D computer graphics software Blender [16] to build a simple human hand animation that captured the motion trajectories of the hand gestures. The motion trajectories were then utilised to synthesise radar data. The drawback of this work is that only micro-Doppler spectrograms were considered and the model was not tested with real data.

In [17], the authors proposed a radar data synthesis flow for macro-gestures. Seven gestures were simulated as a training set, which was used to train the Multi-Layer Perceptron model and the real data were employed to test the model with an average recognition accuracy of 84.2%. The drawback is that the ranges and angles of the gestures were not taken into account.

A human target model for the flexible simulation of various modalities of gesture was constructed in [18]. It covered the main parts of the body. In a similar way to the work in [17], only the Doppler spectrum was simulated. A CNN model was applied to classify eight macro-gestures with an accuracy of 80.4%.

The authors in [19] converted video footage of human activity into realistic, synthetic Doppler radar data by means of a cross-domain conversion approach to achieve the goal of synthesising radar training data for human activity. Other features such as range and angle were not synthesised.

In view of this, the gesture synthetic training data generator proposed by the authors in [20] can generate range–time heat maps, velocity–time heat maps and angle–time heat maps. Six gestures were synthesised and real data were also employed to test their validity. The authors used the VGG19 [21] pre-trained model to extract the features from heat maps. After that, the XGBoost [22] and Random Forest [23] classifiers were employed to recognise the hand gestures. The achieved average accuracy was 84.93% and 87.53%, respectively.

1.3. Contributions

To reduce the complexity of processing and tackle data scarcity, the main contributions of this paper are as follows.

- A simplified gesture recognition algorithm is proposed. The features of the gestures are represented as one-dimensional vectors instead of images.
- A simplified synthetic hand gesture feature generator is presented. As the hand gesture features extracted from the real data are simplified to one dimension, the synthesis processes from [15,17–20] are no longer needed. In our simplified synthetic hand gesture feature generator, the generation of radar raw data is skipped.
- The impact of range, velocity and angle features, extracted from a real data set, on the accuracy of gesture recognition is analysed. The experimental results reveal that all three features have a positive effect on gesture recognition. For the evaluation scenarios with a single feature, the average recognition rate based on the velocity feature alone achieves the highest recognition rate on the test set, with a support vector machine (SVM) classifier, which is 87.59%. For the different feature combination evaluation

scenarios, the average recognition rate based on the three features yields the best result with an average recognition rate of 98.48%.

- The impacts of the synthetic data set on the recognition accuracy of the real gesture data set are investigated. The SVM classifier trained with the synthetic data has an average recognition accuracy of 89.13% on the real data.

The remainder of the paper is organised as follows: Section 2 describes the FMCW radar system. Section 3 introduces low computational complexity algorithms for extracting features from radar raw data and Section 4 presents a synthetic hand gesture feature generator. Section 5 presents the experiments and results. The conclusions are given in Section 6.

2. FMCW Radar System

The system architecture of the FMCW radar is illustrated in Figure 1.

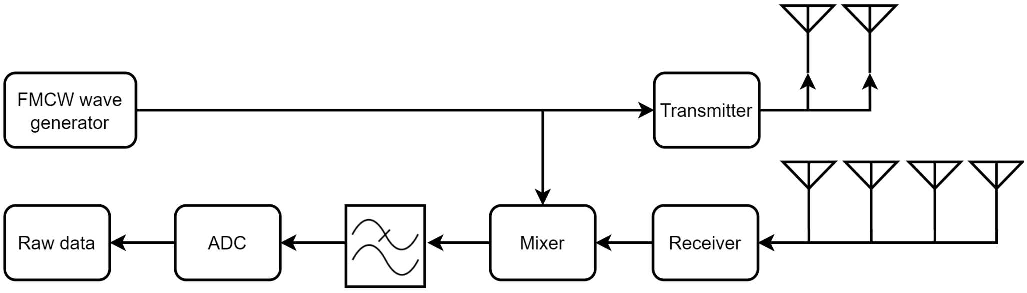


Figure 1. FMCW radar system.

The classical waveforms of FMCW radar are rectangular, upward sawtooth, triangular and staircase voltage waves. Since the hand gesture speed is not as high as an aircraft's, the upward sawtooth waveform is used in our radar. Firstly, the waveform generator generates an upward sawtooth wave as depicted in Figure 2. The bandwidth of the waveform is B. The solid blue line represents the transmitted wave and the dashed black line is the received wave. The frequency slope of the waveform is:

$$s = \frac{B}{T_c}. \tag{1}$$

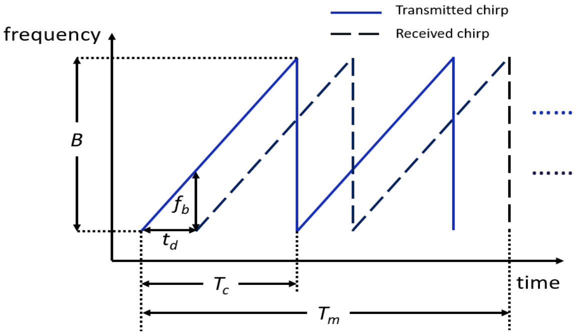


Figure 2. FMCW waveform.

The signal is emitted into space by the transmitter via the transmitting antennas. Our radar has two transmitting and four receiving antennas. The distance between the transmitting antennas is two wavelengths and the distance between the receiving antennas is half a wavelength. A transmitted wave can also be referred to as a chirp. The equation of the transmit chirp is given in (2):

$$T_x(t) = A \exp\left(j(2\pi f_c t + \pi s t^2)\right). \tag{2}$$

where the amplitude of the chirp is denoted as A and the starting frequency is represented by f_c. T_c is the duration of the chirp.

When a wave encounters any object in space, it will be reflected back to the receiver. The object could be our target or any other irrelevant object. The distance between the target and the radar is assumed as r_0, then the time delay t_d between the received and transmitted waves can be represented as:

$$t_d = \frac{2r_0}{c}. \tag{3}$$

where c is the velocity of light.

As the waveform loses energy as it travels through the air and is bounced by objects, there is an amplitude attenuation μ and phase shift in the received waveform. The received wave is defined as follows:

$$R_{\mathrm{x}}(t) = \mu A \exp\left(j(2\pi f_c(t - \frac{2r_0}{c}) + \pi s(t - \frac{2r_0}{c})^2)\right). \tag{4}$$

In the next step, the received and transmitted waves will be mixed in a mixer and passed through a low-pass filter to remove the high-frequency components and preserve the low-frequency signal. The remaining signal at this stage is known as the intermediate frequency signal or beat signal. As shown in Figure 2, the slope and time delay of the waveform are s and t_d, respectively, and the frequency of that beat signal can then be given as f_b:

$$f_b = st_d = \frac{2Br_0}{cT_c}. \tag{5}$$

The beat signal can be defined as follows:

$$\begin{aligned}
B(t) &= \mu A^2 \exp\left(j(2\pi f_c \frac{2r_0}{c} + \pi s \frac{4r_0}{c} t - \pi s (\frac{2r_0}{c})^2)\right) \\
&= \mu A^2 \exp j(\pi s \frac{4r_0}{c} t + 2\pi f_c \frac{2r_0}{c} - \pi s(\frac{2r_0}{c})^2) \\
&= \mu A^2 \exp\left(j(2\pi s t_d t + \underbrace{2\pi f_c \frac{2r_0}{c} - \pi s(\frac{2r_0}{c})^2}_{\phi(t)})\right) \\
&= \mu A^2 \exp\left(j(2\pi f_b t + \phi(t))\right) \\
&= \mu A^2 \exp\left(j(\frac{4\pi B r_0}{cT_c} t + \phi(t))\right).
\end{aligned} \tag{6}$$

In the final step, the beat signal is digitalised by an analogue-to-digital converter (ADC).

3. Feature Extraction from Radar Raw Data

In this section, the process of extracting gesture features from the radar raw data is presented.

3.1. Range and Velocity Extraction

As illustrated in Figure 3, the structure of a frame of radar raw data has three dimensions, namely range, chirps and antennas. There are eight virtual antenna channels in one frame of the data. Each antenna channel has M chirps and each chirp has N range bins.

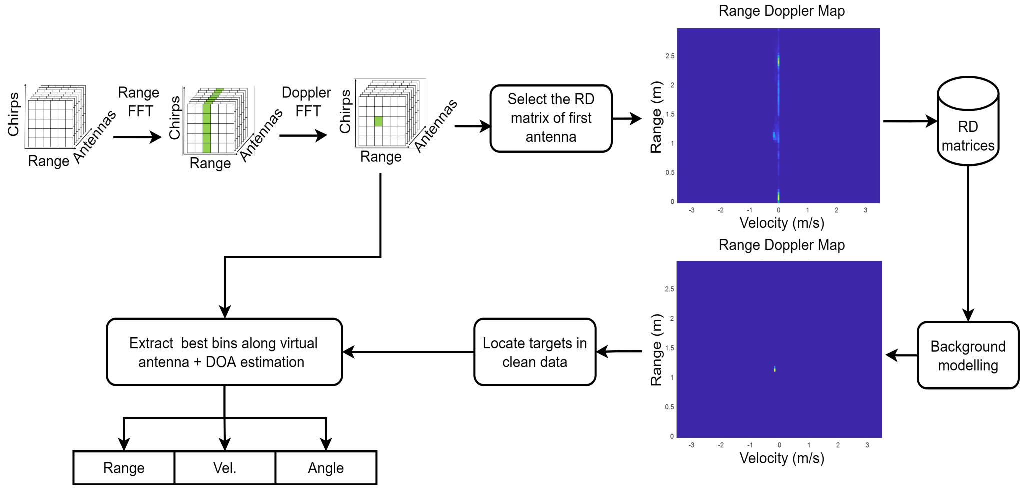


Figure 3. Processing chain for feature extraction from radar raw data.

To derive range and velocity information, a Fast Fourier Transform (FFT) is applied to the range and chirp dimensions. As depicted in Figure 3, after the range FFT, the range information of the target is highlighted. Then after the Doppler FFT, the range-Doppler (RD) matrix is formulated and the velocity information of the target is enhanced.

FMCW radar detects not only dynamic objects but also static ones. The radar data contain static clutter and static backgrounds, which can be disruptive to gesture recognition. Therefore, it is mandatory to take measures to combat irrelevant noise and static background.

Background modelling based on the Greedy bilateral smoothing (GreBsmo) [24] is employed to remove static clutter and static background objects, while the dynamic objects are retained. The data X can be decomposed to the background L, clean data S and noise G.

$$X = L + S + G. \tag{7}$$

After obtaining the RD matrices, the RD matrix for the first antenna channel is selected and saved in the data container. After executing the data for a whole hand gesture, the background modelling is carried out. The two heat maps in Figure 3 indicate the contribution of background modelling. After background modelling, the RD heat map becomes clean and only the target object remains. Thus the range and velocity of the target can be located more accurately in the RD matrix. It is assumed that the moving target in the data set has only one hand and is the main component. The index of that best bin can be derived by finding the maximum value in the RD matrix. Once the index is found, it is possible to calculate the range and velocity of the target.

3.2. Angle Extraction

The target bin is defined as the index of the RD matrix where the target is located. The best bin is searched along the antenna index. The bins being extracted are a 1 × 8 vector, denoted by $\mathbf{e}$.

By performing Fourier beam steering (FB) [25,26] on $\mathbf{e}$, angle information is derived. The virtual beam steering matrix is represented as:

$$\mathbf{V}(\Theta, q) = \exp\left(j2\pi\left(-\frac{Q-1}{2} + q\right)\frac{\Delta d}{\lambda}\sin\left(\frac{\pi}{180}\Theta\right)\right). \tag{8}$$

where Q is the total number of virtual antennas, $q \in \{1, 2, 3, \ldots, 8\}$. λ stands for the wavelength. Δd is the spacing distance between the receiving antennas and its value is $\frac{1}{2}\lambda$. The scanning scope of the angle Θ is $[-90, 90]$ and the step size is 1.

$$I = \mathbf{V} \cdot \mathbf{e}^T. \tag{9}$$

It is assumed that there is only one target and therefore the angle corresponds to the maximum value in I, as depicted in Figure 4.

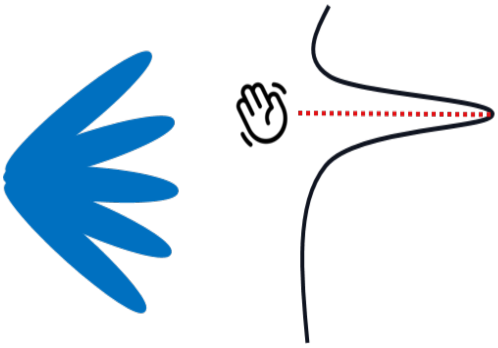


Figure 4. Fourier beam steering.

The processing sequence of our approach makes the angle estimation more stable and accurate. By selecting the best bins in the RD matrices, the angle information of irrelevant objects will be excluded. This is exactly the opposite of the processing order of unmanned aerial vehicle (UAV) swarms detection using the radar in [27,28]. In contrast to UAV swarm detection, only one target is taken into account in gesture recognition and a linear array of antennas is used in the radar.

In contrast to heat map-based gesture recognition, a gesture sample has 32 frames only, leading to a 1 × 96 feature vector. The features of a sample are shown in Figure 5. The overall process of extracting features from the radar raw data is shown in Algorithm 1.

Algorithm 1: Feature extraction from radar raw data.

```
Data: Radar raw data: Data
Result: Range, velocity and angle feature vectors
initialisation;
for $frame\_counter = 1, 2, \ldots, 32$ do
    Perform range FFT to the Data;
    Perform Doppler FFT to the Data;
    Data_container(:,:,$frame\_counter$) = Data (:,:,1);
end
Perform background modelling ;
for $frame\_counter = 1, 2, \ldots, 32$ do
    Locate the peak in the clean RD matrix, compute range and velocity;
    Extract bins along antenna dimension;
    Perform Fourier beam steering;
    Locate the peak, compute the angle;
end
```

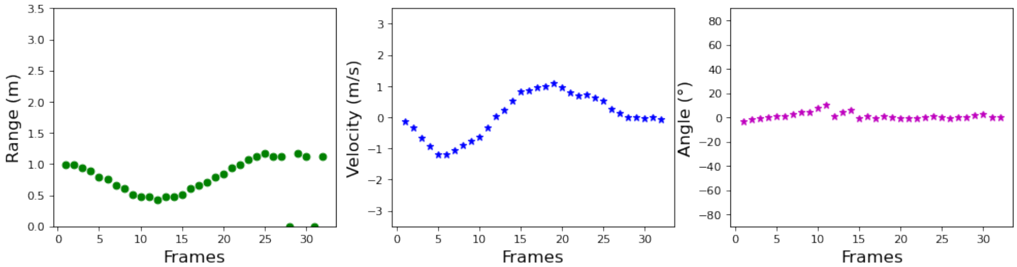


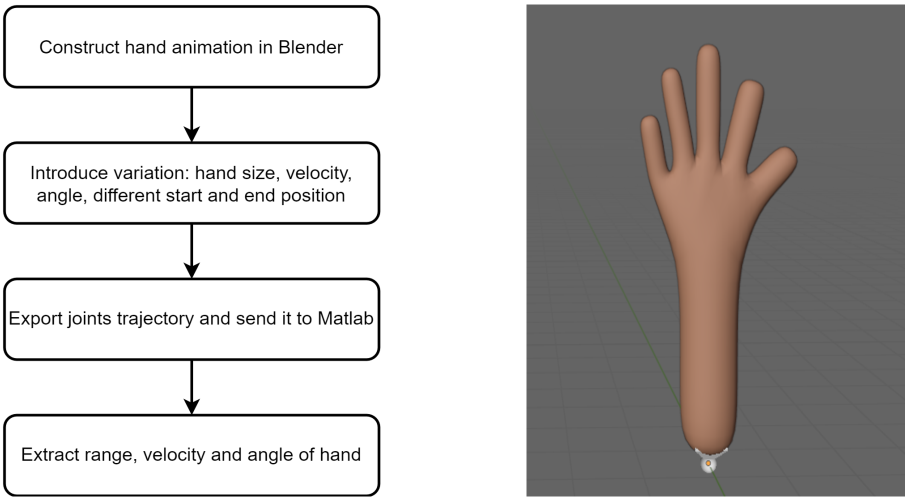
Figure 5. An example of gesture "push pull".

4. Synthetic Feature Generator

To match the gesture features with the real data, a synthetic generator of hand gestures is adapted. In this section, the workflow of the gesture feature synthesiser is proposed as illustrated in Figure 6a.

4.1. Generator Architecture

Our work uses Blender to animate hands. Blender provides the armature API and Python scripting. Python scripts make it easier to produce a large number of hand gestures. In Blender, the skeletal structure and joints of the hand are constructed, as depicted in Figure 6b.


Construct hand animation in Blender
Introduce variation: hand size, velocity, angle, different start and end position
Export joints trajectory and send it to Matlab
Extract range, velocity and angle of hand

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

Figure 6. (**a**) The workflow of the gesture feature synthesiser; (**b**) an example of hand animation.

Human hands vary in size and have different habits of movement. Therefore, the simulator can reproduce different joint types and the hand can be simulated with varying velocities, angles and start–stop positions. The trajectories are fed into Matlab to calculate the features of the hand movement, namely range, velocity and angle.

4.2. Feature Extraction

The process of extracting features from a trajectory is illustrated in Algorithm 2. Figure 7 displays the radar and hand in the 3D space. The orange dot represents the radar with the location represented as (x_r, y_r, z_r). The position of one joint is denoted as (x_i, y_i, z_i). The distance between these two points in space is calculated by (10):

$$d = \sqrt{(x_i - x_r)^2 + (y_i - y_r)^2 + (z_i - z_r)^2}. \quad (10)$$

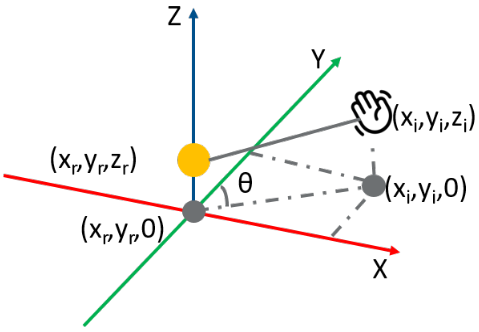


Figure 7. Target and radar in the coordinate system.

The distance from each joint of the simulated hand to the radar is calculated according to (10) and averaged as the distance between the hand and the radar for one frame.

For velocity, the initial value of velocity is set to 0. The formula for the calculation is given in (11). d_{k} is the distance between the radar and the hand in the current frame. $d_{\mathrm{k}-1}$ is the distance in the previous frame while T_{m} represents the duration of a frame. The difference between the distances of adjacent frames divided by the frame duration gives the hand velocity in the current frame.

$$v = \frac{d_{\mathrm{k}} - d_{\mathrm{k}-1}}{T_{\mathrm{m}}}, \mathrm{k} \in \{2, 3, \ldots, 32\}. \tag{11}$$

Algorithm 2: Synthetic feature extraction.

```
Data: Trajectories: Target_pos
Result: Range, velocity and angle feature vectors
initialisation;
for $frame\_counter = 1, 2, \ldots, 32$ do
    if $frame\_counter == 1$ then
        v(1) = 0;
        d($frame\_counter$) = Euclidean distance(Target_pos($frame\_counter$,:),
         Rad_pos);
        $\theta(frame\_counter) = \frac{180}{\pi}$ arctan $(\frac{\text{Target\_pos}(frame\_counter,1)}{\text{Target\_pos}(frame\_counter,2)})$.;
    else
        d($frame\_counter$) = Euclidean distance(Target_pos($frame\_counter$,:),
         Rad_pos);
        v($frame\_counter$) = $\frac{\mathrm{d}(frame\_counter)-\mathrm{d}(frame\_counter-1)}{T_m}$;
        $\theta(frame\_counter) = \frac{180}{\pi}$ arctan $(\frac{\text{Target\_pos}(frame\_counter,1)}{\text{Target\_pos}(frame\_counter,2)})$.;
    end
end
```

Our radar can only detect the azimuth of the object. In Figure 7, the radar and the hand are projected onto the same plane. The grey dot is the projection of the hand. The projection of the radar is then at the origin of the coordinate system and the azimuth angle θ between radar and a joint is derived by (12). The angle between the hand and the radar is averaged over all joints of the hand.

$$\theta = \frac{180}{\pi} \arctan(\frac{\mathrm{x_i}}{\mathrm{y_i}}). \tag{12}$$

To make the synthesised hand gesture features more realistic, random noise is added to the extracted features.

4.3. Recognition Pipeline

The features of the hand gestures are fed directly into the support vector machine (SVM) [29] after they have been extracted based on the approach described previously. The support vector machine algorithm is particularly efficient in terms of memory and it performs better if there is a significant margin of separation between hand gestures.

5. Experiment and Evaluation

5.1. Radar Settings

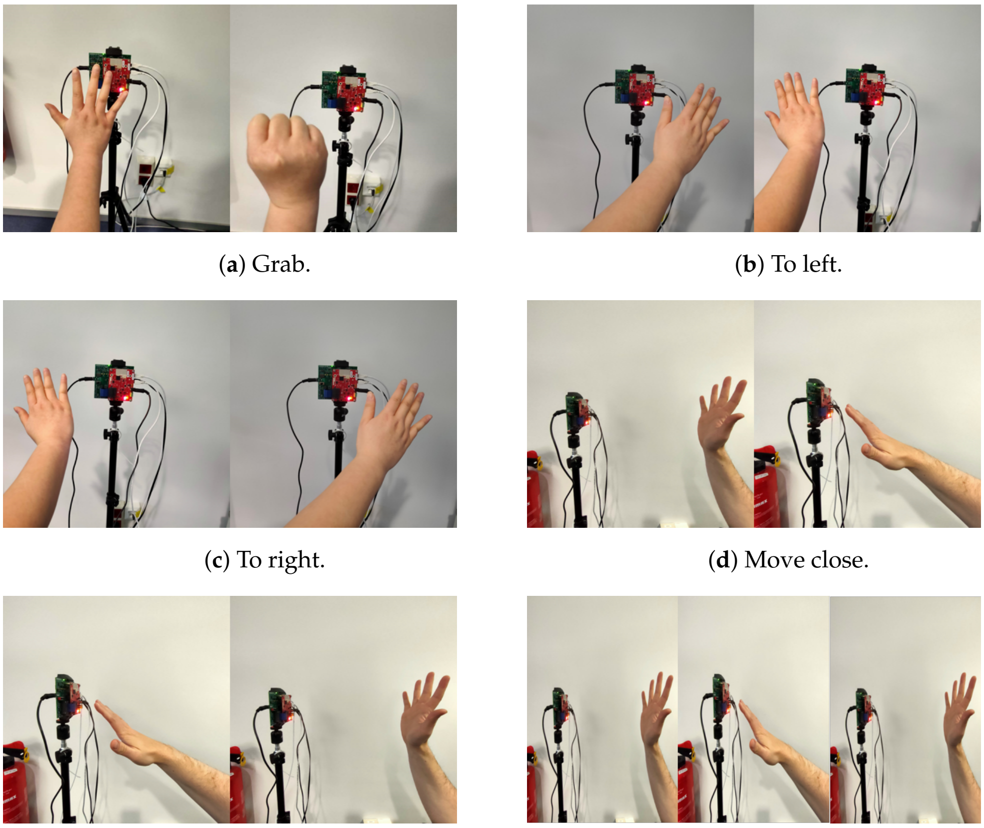
The radar used for the experiments is a Texas Instruments (TI) AWR1642 [30], which operates at a starting frequency of 77 GHz and a maximum bandwidth of 4 GHz. It has two transmitting and four receiving antennas. The specific parameters used for the experiments are listed in Table 1. This radar also needs a raw data acquisition board. For this reason, we use TI DCA1000EVM [31].

5.2. Dataset

Two datasets are collected in this study, a real dataset based on AWR1642 and a synthetic dataset synthesised by the gesture feature simulator. There are six hand gestures in the dataset as shown in Figure 8: "grab", "to left", "to right", "move close", "move away" and "push pull". These are gestures that are commonly used in daily life. The real data set contains 250 samples of each gesture gathered from two participants in an indoor environment. Our low computational complexity approach takes an average of 0.0157 s to extract features from the raw data of each sample. The synthetic data set has 2700 samples per gesture.

Table 1. Radar parameters.

Parameter	Value
Starting frequency	77 GHz
Transmitting antennas	2
Receiving antennas	4
Number of range bins	64
Number of chirps per frame	255
Bandwidth	3.8 GHz
Chirp duration	38 µs
Frequency slope	100 MHz/µs
Frame duration	71 ms
Number of frames per gesture	32



(**a**) Grab.

(**b**) To left.

(**c**) To right.

(**d**) Move close.

(**e**) Move away.

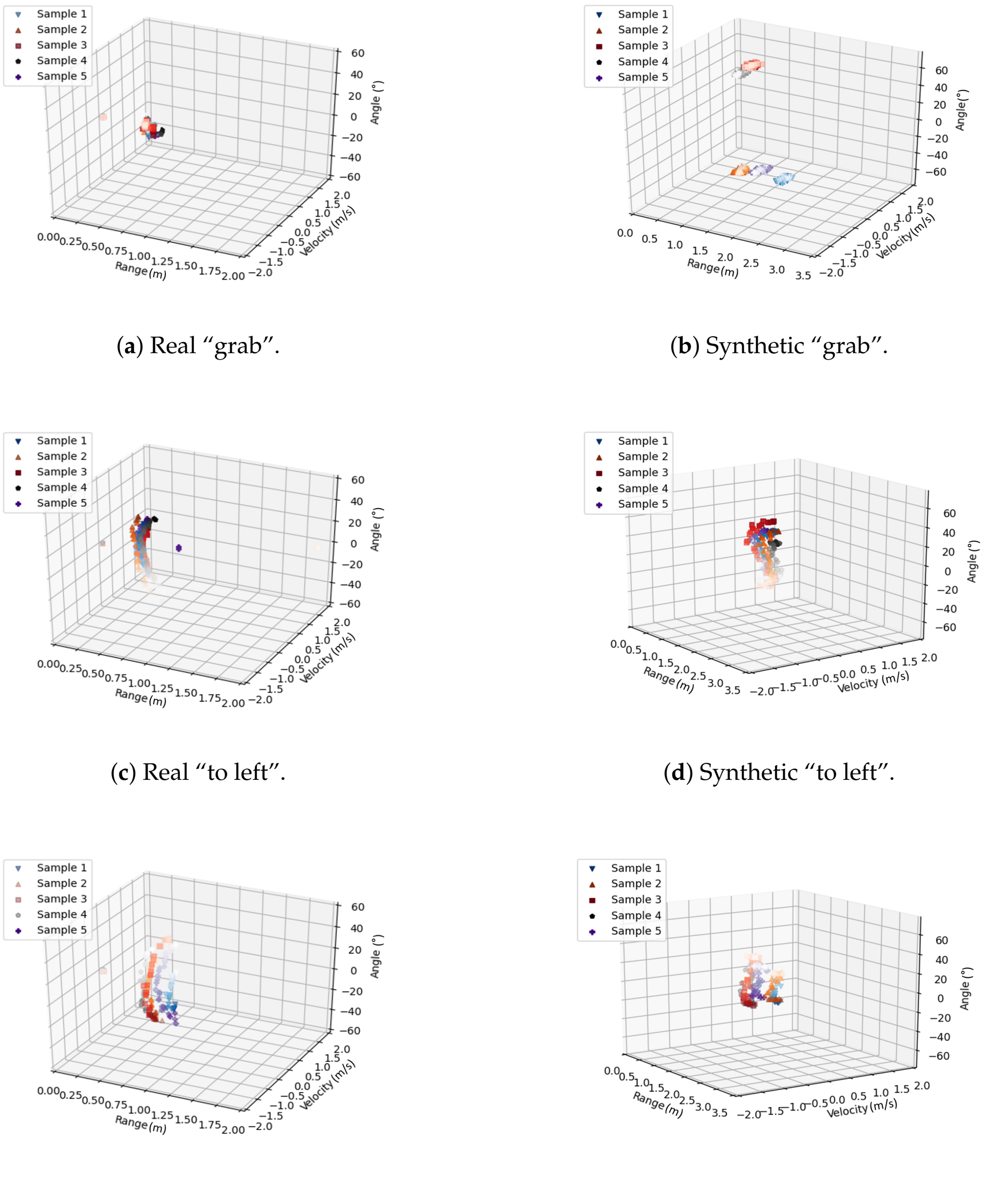
(**f**) Push pull.

Figure 8. Hand gesture type.

5.3. Feature Comparison

Next, features from the real dataset and synthetic features are analysed and compared. The features of the hand gestures are illustrated in a 3D scatter plot. For readability, only five samples of each gesture are displayed. A gradient colour has been used for the feature scatter plot, from dark to light. The starting frame of the feature for each sample is the darkest colour and the last frame is the lightest colour.

Figure 9a,b indicate the features of the real "grab" and the synthetic "grab". This gesture does not change much in terms of range, velocity and angle and that is because the grabbing, with the fingers slowly closing together, does not change much in motion.



(**a**) Real "grab".

(**b**) Synthetic "grab".

(**c**) Real "to left".

(**d**) Synthetic "to left".

(**e**) Real "to right".

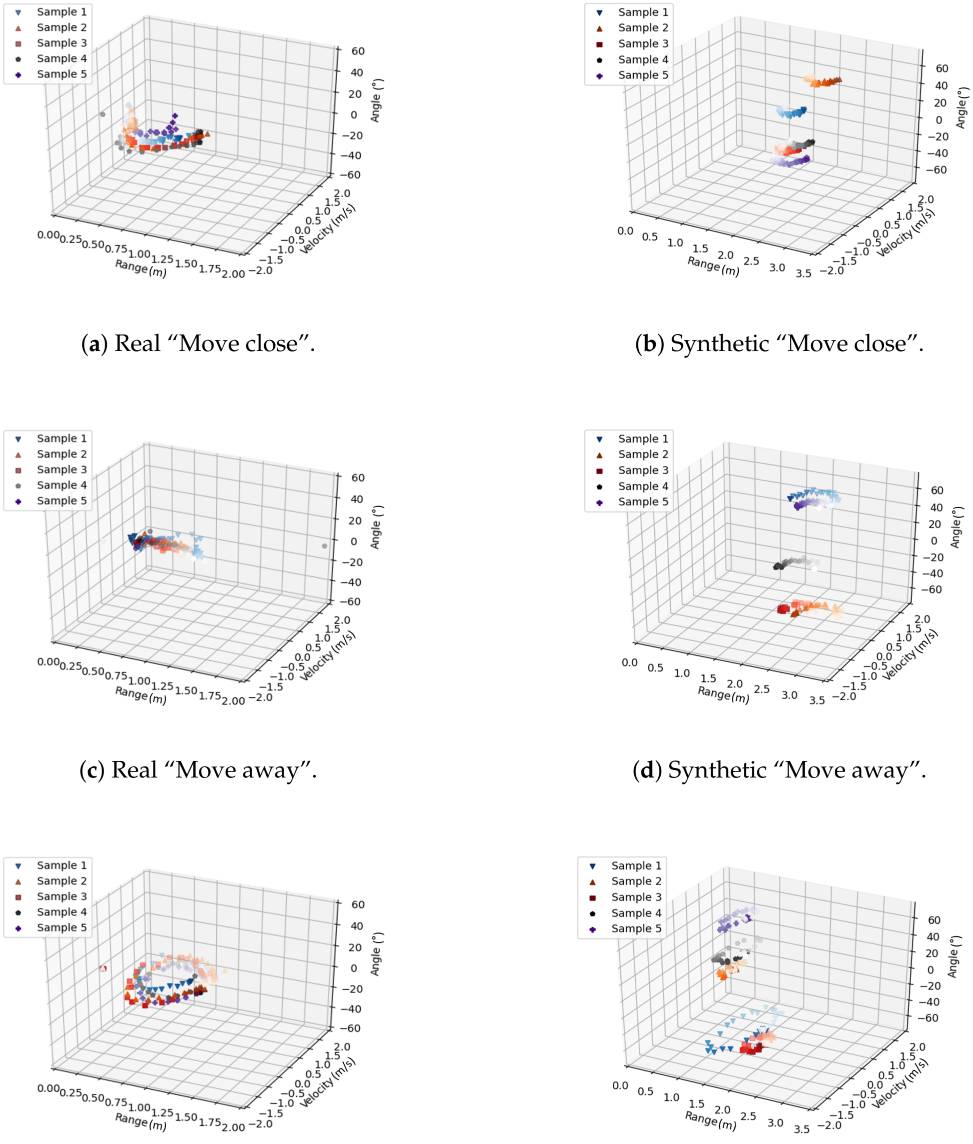
(**f**) Synthetic "to right".

Figure 9. Comparison of "grab", "to left" and "to right".

"To left" gesture is compared in Figure 9c,d. The real "to left" and the synthetic "to left" both have a drastic change in the dimension of the angle. A trend from larger to smaller angles can be seen on both figures. The hand moves from right to left and there will surely be a changing of angle relative to the radar. In contrast, there is not a lot of variation but a small decreasing and increasing trend in the range. This comes from the fact that the midpoint of the gesture is closer to the radar than the start and end points. Some small variation is expected because human movements are not perfectly aligned with the radar. The same is true for the velocity because only movements towards or away from the radar will influence the measured velocity.

The change in angle features for “to right” is the opposite of “to left” in Figure 9e,f. The real “to right” angle changes from a negative to a positive value. The other features of the gesture are identical to the “to left” gesture.

Figure 10a,b present the 3D features of the real “move close” and the synthetic “move close”. The characteristic element of this gesture is the change of the range from larger to smaller values. This is because the hand is gradually moving closer to the radar. The velocity changes from zero to a negative value and then back to zero when the movement is finished. There is almost no change in angle. Some variation can be observed because the target is not perfectly positioned to move directly towards the radar.



(**a**) Real “Move close”.

(**b**) Synthetic “Move close”.

(**c**) Real “Move away”.

(**d**) Synthetic “Move away”.

(**e**) Real “Push pull”.

(**f**) Synthetic “Push pull”.

Figure 10. Comparison of “move close”, “move away” and “push pull”.

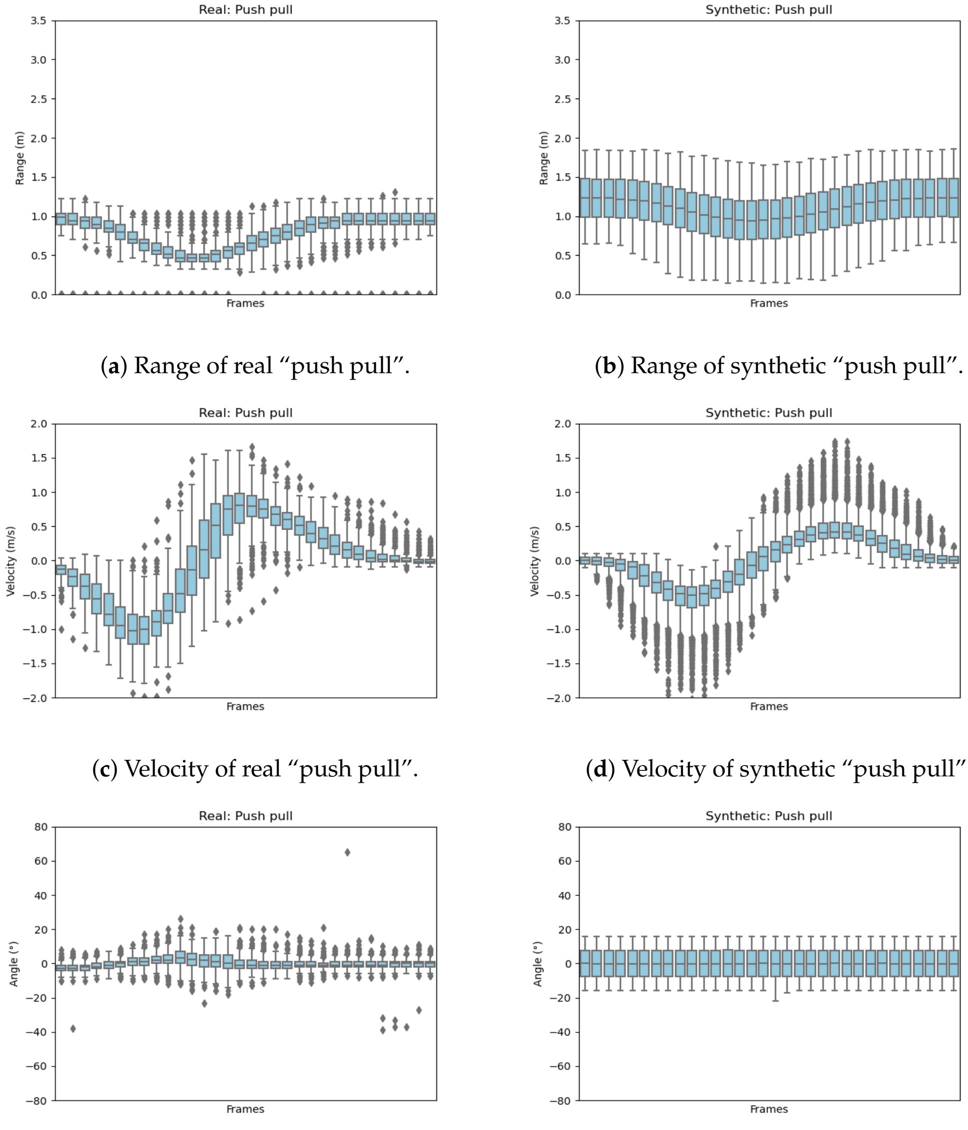
For the “move away” gesture, the range between the hand and radar gradually increases as the hand moves in Figure 10c,d. In addition, the trend of the velocity is the opposite of that of “move close”. The angle does not have much variation and the little variation observed behaves in the opposite manner of “move close”.

“Push pull” is a combination of “move close” and “move away”. The hand first approaches the radar and then moves away. The distance decreases and then increases.

The corresponding velocity changes towards positive and negative values. The angle does not change obviously. The feature of "push pull" in Figure 10e,f is in the form of a closed circle. There is a high similarity between the real and synthetic features.

5.4. Feature Distribution

To show the distribution of features for one gesture from the entire data set, the distribution of features for the gesture "push pull" is illustrated in Figure 11.



(**a**) Range of real "push pull".

(**b**) Range of synthetic "push pull".

(**c**) Velocity of real "push pull".

(**d**) Velocity of synthetic "push pull".

(**e**) Angle of real "push pull".

(**f**) Angle of synthetic "push pull".

Figure 11. Box-plot for feature distribution of real and synthetic "push pull".

This "push pull" distribution is based on 250 samples in the real data set and 2700 samples in the synthetic data set. It can be concluded from Figure 11a,b that the distribution zone of the range features for most of the "push pull" samples first decreases and then increases over time. The distribution of synthetic range features is relatively wider; this is due to the richness of the hand start and end position variations during the simulation of the trajectory. The synthetic data are purposefully created with as much variation as possible while still performing the hand gesture. The goal of this is to represent as many different ways to perform this gesture as possible. The speed of the real gestures seems to be higher than most of the synthetic data but the velocity of the real gesture is still included in the synthetic data. The velocity pattern for this gesture can be seen in both data sets. For the angle, the synthetic data cover a wider range of values; this will lead to a better classification performance if the gesture is performed from different angles. The overall pattern is similar to the real data. It can be seen that the real gesture was usually measured directly from the front. The synthetic data also cover the gesture if it is performed at an angle.

5.5. Feature Impact Analysis

From the radar raw data, three features have been extracted, namely range, velocity and angle. In this subsection, the effect of feature combinations on the accuracy of gesture recognition will be analysed. To analyse the impact of one type of feature, a data set containing a single feature is fed into the SVM for training and testing. A total of 50% of the real data set was randomly selected as the training set and the other 50% as the test set.

The experiment is repeated ten times and the recognition rates for the test set are summarised in Figure 12 and Table 2. As can be seen in the figure, velocity plays a significant role. For velocity alone, the average recognition accuracy in the test set was as high as 87.59%.

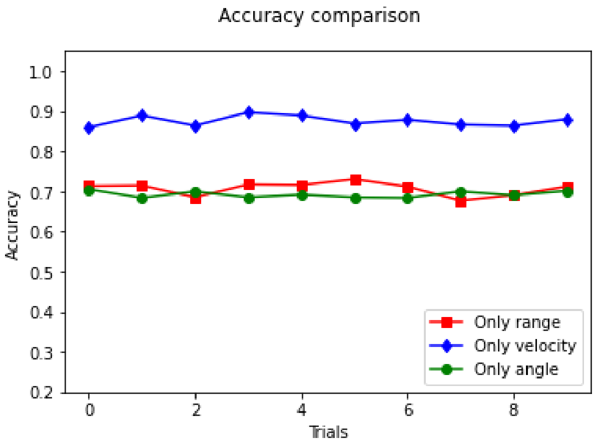


Figure 12. Accuracies based on single feature.

Table 2. Comparison of accuracy based on a single feature.

Feature Type	Average Accuracy	Standard Deviation
Only range	70.69%	1.59%
Only velocity	87.59%	1.23%
Only Angle	69.28%	0.78%

Furthermore, the combination of features is evaluated. The results based on the combination of different features are illustrated in Figure 13 and Table 3. Recognition rates based on multiple features are higher compared to a single feature. The average accuracy based on the three features is as high as 98.48%. The combination of velocity and angle can achieve accuracies of up to 98.15% on the test set.

From the outcomes, it can be derived that the velocity is more recognisable among the three features. However, based on velocity alone, the recognition rate of gestures is below 90%. When range and angle are also taken into account, the recognition accuracy on the test set is improved considerably.

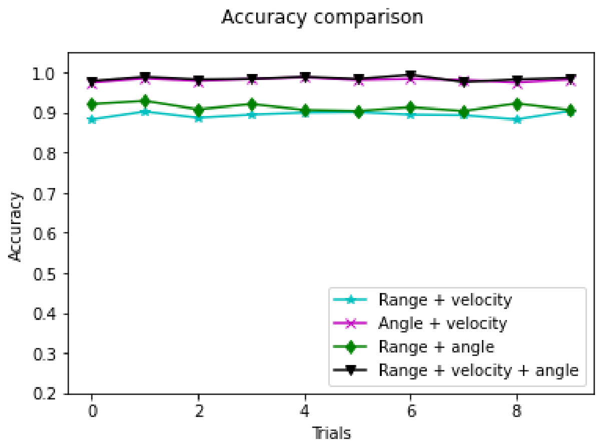


Figure 13. Accuracies based on multi features.

Table 3. Comparison of accuracy based on feature combination.

Feature Combination	Average Accuracy	Standard Deviation
Range + velocity	89.43%	0.76%
Velocity + angle	98.15%	0.42%
Range + angle	91.37%	0.87%
Range + velocity + angle	98.48%	0.52%

5.6. Synthetic Feature Impact Analysis

Three evaluation scenarios are defined. Scenario I is a random selection of 2% of the real data as the training set and the rest of the real data are used as the test set. Scenario II uses the synthetic data set as the training set and the entire real data set as the test set. Scenario III takes 2% of the real data and the entire synthetic data set as the training set and the remaining 98% of the real data set as the test set.

The recognition accuracy of the SVM on the test set is given in Figure 14 and Table 4. The experiment is repeated ten times.

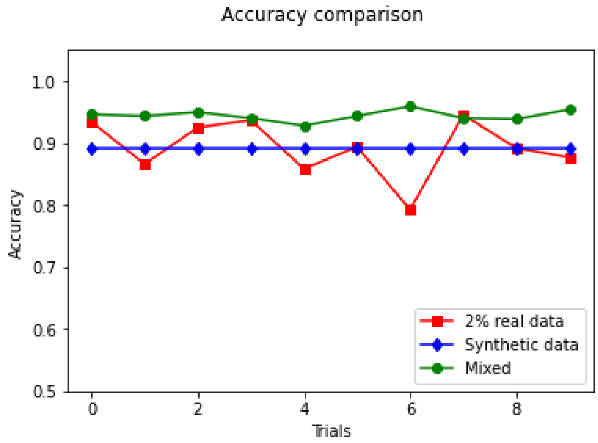


Figure 14. Accuracy of test set.

Table 4. Comparison of accuracy.

Scenarios	Average Accuracy	Standard Deviation
Scenario I: 2% real data	89.21%	4.43%
Scenario II: Synthetic data	89.13%	0.00%
Scenario III: Mixed	94.43%	0.82%

The red line represents the accuracy of scenario I on the test set. It can be seen that the high fluctuation is due to a low amount of real data and the quality of the selected data varies. If the 2% randomly selected real samples from the real data set cover a large variation and have a high quality, then it is likely to yield good results on the test set. If the randomly selected samples have low quality or little variation, there is a chance that worse results can be obtained on the test set. The results for scenario II are indicated by the blue line. The blue line is stable because the test set and training set have not changed. The green line represents the result of scenario III. These results are the best among the scenarios. This shows that the synthetic data are able to enhance data sets of real data to achieve a performance that is higher than the individual data sets.

The average accuracy of ten attempts for the three scenarios is given in Table 4. When the SVM classifier uses only the synthetic data set as the training set, the model has an average accuracy of 89.13% on the test set. When the synthetic data set is combined with a small amount of real data, better performance can be obtained, with an average recognition rate of 94.43%.

One more advantage of synthetic data is that they cover a wider range of possible gestures. The training and test sets are very similar, so a small amount of real data can lead to high performance in the test set. A test set that differs in these features but still includes valid gestures may result in poorer performance for a model trained on real data. Synthetic data has a wider range of feature values, so it is able to generalise better and cover different test sets in a better way.

5.7. Comparison with Other Works on Synthetic Feature Generators

In Table 5, a comparison with other works is summarised. Previous work such as [17] combined animation modelling in Blender with a simulated FMCW radar sensor to simulate velocity–time heatmap features for seven types of macro-gesture. In [18], the human model was constructed and Doppler spectrograms for eight macro-gestures were simulated. The authors synthesised six activities in [19] by transforming the data from the camera into Doppler heat maps. The number of samples in their test set is 720.

Table 5. Comparison with other works.

Ref.	Sample Size	Real Samples	Testset Average Accuracy
[17]	-	1050	84.2%
[18]	-	3354	80.4%
[19]	-	720	81.4%
[20]	8–12 KB	1500	87.53%
This work	1 KB	1500	89.13%

Reference [20] simulated range–time, velocity–time and angle–time heatmap features for six types of hand gestures by combining gestures constructed in Blender and a simulated FMCW radar. For both, the classifiers were trained with synthetic data sets and the models were tested with real data sets. However, this does not suit our features extracted from the radar raw data.

The features of our sample are saved in a csv [32] file. In Table 5, it can be observed that the features extracted from a sample of [20] are 8 to 12 times larger than the features extracted by our new approach. Compared to [17–20], the average recognition accuracy on the real data set is higher, which indicates the strength of this work. In addition to this, our synthetic feature generator does not contain radar signal simulation, which reduces the computational effort significantly. More importantly, the dataset extracted by our method does not need to be fed further into the machine learning algorithm for feature extraction. This will help to save hardware resources significantly.

6. Conclusions and Future Work

In this work, a low computational complexity hand gesture feature extraction methodology based on FMCW radar and a synthetic hand gesture feature generator are proposed.

The impact of range, velocity and angle on the accuracy of gesture recognition is analysed. The combination of the three features leads to the highest accuracy.

Compared to other works, our synthetic gesture feature process avoids the generation of radar raw data and is more straightforward. In contrast to heat map-based recognition, additional feature extraction can be skipped in the training phase.

The synthetic data set is used to train a SVM classifier to recognise six different hand gestures in the real data set, which are acquired with an AWR1642 FMCW radar. The algorithm achieves an average recognition accuracy of 89.13% on the real data set. By combining a small amount of the real data set with the synthetic data set, an average recognition accuracy of 94.43% is obtained on the real data set. Thus, we demonstrate the effectiveness of our raw data pre-processing approach and feature synthetisation process.

In the future, more gesture types, different frequency bands and cross-platform gesture recognition should be investigated. We would like to verify the performance of our selected models in different aspects in future work, for example, by adding more datasets with different characteristics.

Author Contributions: Data curation, Y.Z.; Methodology, Y.Z.; Supervision, V.S., M.K. and E.G.; Writing—original draft, Y.Z.; Writing—review & editing, V.S., M.K. and E.G. All authors have read and agreed to the published version of the manuscript.

Funding: This work is partially funded by the Federal Ministry of Education and Research of Germany (BMBF) within the iCampus Cottbus project (grant number: 16ME0424) and the “Open6GHub” project (grant number: 16KISK009).

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

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

References

1. Ahmed, S.; Kallu, K.D.; Ahmed, S.; Cho, S.H. Hand Gestures Recognition Using RADAR Sensors for Human-Computer-Interaction: A Review. *Remote Sens.* **2021**, *13*, 527. [CrossRef]
2. Xu, Z. Bi-Level l_1 Optimization-Based Interference Reduction for Millimeter Wave Radars. *IEEE Trans. Intell. Transp. Syst.* **2022**. [CrossRef]
3. Molchanov, P.; Gupta, S.; Kim, K.; Pulli, K. Short-range FMCW monopulse RADAR for hand-gesture sensing. In Proceedings of the 2015 IEEE RADAR Conference, Johannesburg, South Africa, 27–30 October 2015.
4. Molchanov, P.; Gupta, S.; Kim, K.; Pulli, K. Multi-sensor system for driver’s hand-gesture recognition. In Proceedings of the 2015 11th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG), Ljubljana, Slovenia, 4–8 May 2015.
5. Lien, J.; Gillian, N.; Karagozler, M.E.; Amihood, P.; Schwesig, C.; Olson, E.; Raja, H.; Poupyrev, I. Soli: Ubiquitous gesture sensing with millimeter wave RADAR. *ACM Trans. Graph.* **2016**, *35*, 142. [CrossRef]
6. Wang, S.; Song, J.; Lien, J.; Poupyrev, I.; Hilliges, O. *Interacting with Soli: Exploring Fine-Grained Dynamic Gesture Recognition in the Radio-Frequency Spectrum*; ACM: New York, NY, USA, 2016; pp. 851– 860.
7. Wang, Y.; Shu, Y.; Jia, X.; Zhou, M.; Xie, L.; Guo, L. Multifeature Fusion-Based Hand Gesture Sensing and Recognition System. *IEEE Geosci. Remote Sens. Lett.* **2021**. [CrossRef]
8. Wang, Y.; Ren, A.; Zhou, M.; Wang, W.; Yang, X. A Novel Detection and Recognition Method for Continuous Hand Gesture Using FMCW RADAR. *IEEE Access* **2020**, *8*, 167264–167275. [CrossRef]
9. Zhao, Y.; Sark, V.; Krstic, M.; Grass, E. Novel Approach for Gesture Recognition Using mmWave FMCW RADAR. In Proceedings of the 2022 IEEE 95th Vehicular Technology Conference: (VTC2022-Spring), Helsinki, Finland, 19–22 June 2022; pp. 1–6. [CrossRef]
10. Sun, Y.; Fei, T.; Li, X.; Warnecke, A.; Warsitz, E.; Pohl, N. Realtime RADAR-based gesture detection and recognition built in an edge computing platform. *IEEE Sens. J.* **2020**, *20*, 10706–10716. [CrossRef]
11. Ahmed, S.; Kim, W.; Park, J.; Cho, S.H. Radar-Based Air-Writing Gesture Recognition Using a Novel Multistream CNN Approach. *IEEE Internet Things J.* **2022**, *9*, 23869–23880. [CrossRef]

12. Salami, D.; Hasibi, R.; Palipana, S.; Popovski, P.; Michoel, T.; Sigg, S. Tesla-Rapture: A Lightweight Gesture Recognition System from mmWave Radar Sparse Point Clouds. *IEEE Trans. Mob. Comput.* **2022**. [CrossRef]
13. Boulic, R.; Thalmann, N.M.; Thalmann, D. A global human walking model with real-time kinematic personification. *Vis. Comput.* **1990**, *6*, 344–358. [CrossRef]
14. Trommel, R.P.; Harmanny, R.I.A.; Cifola, L.; Driessen, J.N. Multitarget human gait classification using deep convolutional neural networks on micro-doppler spectrograms. In Proceedings of the 2016 European RADAR Conference (EuRAD), London, UK, 5–7 October 2016; pp. 81–84.
15. Ishak, K.; Appenrodt, N.; Dickmann, J.; Waldschmidt, C. Human motion training data generation for RADAR based deep learning applications. In Proceedings of the 2018 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM), Munich, Germany, 16–17 April 2018; pp. 1–4.
16. Blender. Free and Open Source 3D Creation Suite. Available online: https://www.blender.org/ (accessed on 27 July 2022).
17. Ninos, A.; Hasch, J.; Alvarez, M.; Zwick, T. Synthetic RADAR Dataset Generator for Macro-Gesture Recognition. *IEEE Access* **2021**. [CrossRef]
18. Kern, N.; Aguilar, J.; Grebner, T.; Meinecke, B.; Waldschmidt, C. Learning on Multistatic Simulation Data for Radar-Based Automotive Gesture Recognition. *IEEE Trans. Microw. Theory Tech.* **2022**, *70*, 5039–5050. [CrossRef]
19. Ahuja, K.; Jiang, Y.; Goel, M.; Harrison, C. Vid2Doppler: Synthesising Doppler radar data from videos for training privacy-preserving activity recognition. In Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems, Yokohama, Japan, 8–13 May 2021; pp. 1–10.
20. Zhao, Y.; Sark, V.; Krstic, M.; Grass, E. Synthetic Training Data Generator for Hand Gesture Recognition Based on FMCW RADAR. In Proceedings of the 2022 23rd International Radar Symposium (IRS), Gdansk, Poland, 12–14 September 2022; pp. 463–468. [CrossRef]
21. Simonyan, K.; Zisserman, A. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv* **2014**, arXiv:1409.1556.
22. Chen, T.; Guestrin, C. XGBoost: A scalable tree boosting system. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794.
23. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32.:1010933404324. [CrossRef]
24. Zhou, T.; Tao, D. Greedy Bilateral Sketch, Completion; Smoothing. In Proceedings of the Sixteenth International Conference on Artificial Intelligence and Statistics, Scottsdale, AZ, USA, 29 April–1 May 2013; pp. 650–658.
25. Sit, Y.L. MIMO OFDM Radar-Communication System with Mutual Interference Cancellation. Ph.D. Thesis, Karlsruher Institut für Technologie (KIT), Karlsruhe, Germany, 2017. [CrossRef]
26. Zhao, Y.; Sark, V.; Krstic, M.; Grass, E. Low Computational Complexity Algorithm for Hand Gesture Recognition using mmWave RADAR. In Proceedings of the 2022 International Symposium on Wireless Communication Systems (ISWCS), Hangzhou, China, 19–22 October 2022; pp. 1–6. [CrossRef]
27. Zheng, J.; Chen, R.; Yang, T.; Liu, X.; Liu, H.; Su, T.; Wan, L. An Efficient Strategy for Accurate Detection and Localization of UAV Swarms. *IEEE Internet Things J.* **2021**, *8*, 15372–15381. [CrossRef]
28. Zheng, J.; Yang, T.; Liu, H.; Su, T.; Wan, L. Accurate Detection and Localization of Unmanned Aerial Vehicle Swarms-Enabled Mobile Edge Computing System. *IEEE Trans. Ind. Inform.* **2021**, *17*, 5059–5067. [CrossRef]
29. Chang, C.; Lin, C. LIBSVM: A library for support vector machines. *ACM Trans. Intell. Syst. Technol.* **2011**, *2*, 27. [CrossRef]
30. AWR1642. Single-Chip 76-GHz to 81-GHz Automotive RADAR Sensor Evaluation Module. Available online: http://www.ti.com/tool/AWR1642BOOST (accessed on 8 July 2022).
31. DCA1000EVM. Real-Time Data-Capture Adapter for RADAR Sensing Evaluation Module. Available online: https://www.ti.com/tool/DCA1000EVM (accessed on 8 July 2022).
32. *RFC 4180: Common Format and MIME Type for Comma-Separated Values (CSV) Files*; SolidMatrix Technologies, Inc.: Staten Island, NY, USA, 2005. [CrossRef]

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.