sensors

Article

A Novel Gesture Recognition System Based on CSI Extracted from a Smartphone with Nexmon Firmware

Tao Li [1], Chenqi Shi [1], Peihao Li [1] and Pengpeng Chen [1,2,*]

[1] School of Computer Science and Technology, China University of Mining and Technology, Xuzhou 221116, China; li_t@cumt.edu.cn (T.L.); shichenqi@cumt.edu.cn (C.S.); lph@cumt.edu.cn (P.L.)
[2] China Mine Digitization Engineering Research Center, Ministry of Education, Xuzhou 221116, China
* Correspondence: chenp@cumt.edu.cn

Abstract: In this paper, we propose a novel gesture recognition system based on a smartphone. Due to the limitation of Channel State Information (CSI) extraction equipment, existing WiFi-based gesture recognition is limited to the microcomputer terminal equipped with Intel 5300 or Atheros 9580 network cards. Therefore, accurate gesture recognition can only be performed in an area relatively fixed to the transceiver link. The new gesture recognition system proposed by us breaks this limitation. First, we use nexmon firmware to obtain 256 CSI subcarriers from the bottom layer of the smartphone in IEEE 802.11ac mode on 80 MHz bandwidth to realize the gesture recognition system's mobility. Second, we adopt the cross-correlation method to integrate the extracted CSI features in the time and frequency domain to reduce the influence of changes in the smartphone location. Third, we use a new improved DTW algorithm to classify and recognize gestures. We implemented vast experiments to verify the system's recognition accuracy at different distances in different directions and environments. The results show that the system can effectively improve the recognition accuracy.

Keywords: gesture recognition; WiFi; channel state information; nexmon firmware

Citation: Li, T.; Shi, C.; Li, P.; Chen, P. A Novel Gesture Recognition System Based on CSI Extracted from Smartphone with Nexmon Firmware. *Sensors* **2021**, *21*, 222. https://doi.org/10.3390/s21010222

Received: 22 November 2020
Accepted: 29 December 2020
Published: 31 December 2020

1. Introduction

Human gesture recognition and activity recognition are gradually becoming prominent promoters of human-computer interaction technology development. Traditionally, gesture recognition is usually implemented using technologies such as depth and infrared cameras [1–3] or wearable devices [4–6]. However, these solutions require significant overhead and non-negligible costs when deployed. The camera-based method mainly uses computer vision processing technology limited by the line of sight and light intensity, requires high surrounding environmental conditions, and has the risk of privacy leakage. Although the wearable device-based method can achieve high accuracy, the wearable sensing device needs to be attached to the user's hand or body, which may cause inconvenience in some cases and requires higher additional costs. To overcome these limitations, WiFi-based gesture recognition systems have emerged in recent years [7–13].

Existing WiFi-based gesture recognition research can be roughly divided into two categories. One is based on Channel State Information (CSI), which is extracted from computers equipped with Intel 5300 network cards or Atheros 9580 network cards [14,15]. The other is based on a Received Signal Strength Indicator (RSSI), which can be extracted directly from a smartphone.

In the former, because the Intel 5300 network card can only collect 30 subcarriers, which cannot meet the fine-grained recognition of gestures, so most of them use multiple links to increase the number of subcarriers for CSI acquisition. Multiple transmitters or receivers with multiple antennas must be deployed. In addition, the person performing gesture recognition needs to prepare a laptop in advance and walking to a specific range near the laptop for gesture recognition. On account of the hard moving of the transceiver, the difficulty of gesture recognition is significantly increased when the relative position of the person and the transceiver is changed, and the recognition accuracy will be greatly reduced.

In the latter, the mobility of the smartphone can ensure the relative position of the personnel and the receiving terminal to a certain extent, which can better solve these problems. However, due to the CSI at the bottom of the smartphone cannot be obtained, the existing gesture recognition based on smartphone WiFi exploits RSSI. For example, WiGest [16] uses the influence on the RSSI of gestures close to the mobile receiver, realize gesture recognition, and then maps gestures to different applications operations. However, limited by the low dimensionality and coarse granularity of RSSI, in order to eliminate the influence caused by different distances between people and equipment, WiGest needs to cooperate with computer-side CSI to ensure accuracy. These limitations make these methods difficult to deploy into a practical and user-friendly system for gesture recognition. As RSSI is rough and susceptible to environment and distance, it is not easy to achieve stable high precision for fine gesture recognition.

To break through these limitations, some researchers [17,18] try to realize cross-domain gesture recognition by using transfer learning and antagonistic learning. However, these schemes basically require additional data collection and model retraining each time when a new target domain is added to the recognition model. Moreover, Widar 3.0 [19] realizes cross-domain gesture recognition in different locations and environments by deriving and estimating the speed profile of gestures at lower signal levels. However, such a scheme requires the deployment of six transceivers in a 2 × 2-m area to achieve higher accuracy.

In response to these existing problems, we hope to establish an easy-to-deploy, low-cost gesture recognition system that can be deployed on smartphones and only requires one transmitter and one receiver, leveraging the convenient mobility of smartphones to better adapt to the environment and changes in personnel positions. However, to implement such a system, we face three main technical challenges. First, the existing smartphones only can obtain RSSI and cannot extract CSI. The coarse-grained RSSI cannot guarantee the accuracy of gesture recognition. Second, the interference of the environment and personnel will affect the accuracy of the system. Finally, when different users and even the same users perform the same action, there are individual differences in gesture speed, action amplitude, and duration.

To address these challenges, a novel gesture recognition system called MDGest based on smartphones is proposed. It only needs one transmitting end and one receiving end for rapid deployment. More importantly, it supports 802.11ac and allows for bandwidths of up to 80 MHz to extract 256 CSI subcarriers from smartphones exploiting nexmon firmware [20,21], which is more than the sum of the subcarriers collected by the Intel 5300 network card with eight links. Furthermore, in daily life, almost everyone carries a smartphone, so we can take out the smartphone for gesture recognition at any time, instead of preparing a laptop in advance and walking near the laptop or bringing it to us for gesture recognition. To select subcarriers that better reflect the characteristics of gestures, we designed a key subcarrier screening algorithm. In response to environmental changes, we propose an environmental noise removal mechanism that uses a wavelet-based denoising method to filter out environmental noise while keeping the CSI mode due to the influence of gestures as much as possible. Aiming at the problems of individual diversity and inconsistency of gestures, the main pattern components of CSI are extracted and identified. An improved DTW algorithm is used to reduce the influence of smartphones distance and angle.

We implemented MDGest on a Google Nexus 5 phone and evaluated its performance with sixteen different users in three typical environments. Moreover, we conduct extensive experiments, including eight Los distances, eight gesture execution distances from the smartphone, four orientations, human interference situations, and comparisons with WiGest [16] and WiFinger [9].

In a nutshell, the main contributions of the paper are four-fold:

- firstly, we propose a novel gesture recognition system that extracts 256 CSI subcarriers from a smartphone in IEEE 802.11ac mode on 80 MHZ bandwidth to implement gesture recognition using nexmon firmware.
- secondly, a method of extracting the main mode of CSI using cross correlation function and cross power spectral density is proposed, which can extract the time domain and frequency domain features of CSI.
- thirdly, we use a new improved DTW algorithm with higher computing efficiency, making our system have better adaptability to individual diversity and gesture inconsistency.
- fourthly, we conducted extensive experiments in three typical environments, with multiple participants under different conditions. The results show that MDGest achieves consistently high accuracies across different users in different environments and locations.

The rest of the paper is organized as follows. Section 2 introduce related works regarding wireless sensing system. Section 3 presents the details of the MDGest system. We implement the MDGest system and evaluate its performance in different scenarios in Section 4. In Section 5, we discuss some of the limitations of the MDGest system. Finally, we conclude the paper in Section 6.

2. Related Work

Our work is highly related to wireless human sensing technology. We break these tasks into two parts: wireless sensing systems, including WiFi-based indoor positioning, tracking, and general human activity recognition; gesture recognition systems, including gesture recognition, finger gesture recognition, and handwriting recognition systems.

2.1. Wireless Sensing System

WiFall uses features such as activity duration and rate of change in CSI values [22] to detect whether users fall down. E-eyes [23] is a pioneer in identifying human activities using commercial WiFi signal strength distribution and KNN. E-eyes generates histograms of CSI values and uses them as features to recognize gestures, such as brushing teeth, taking a bath, etc. HeadScan [24] and BodyScan [25] mounts the antenna on the users. HeadScan uses the average, median, and standard deviation of CSI values to identify gestures related to the mouth, such as coughing and eating. In terms of signals, existing methods extract various parameters of human reflection or occlusion signals, including DFS [26–28], ToF [29–32], AoA, AoD [29,32–34] and attenuation [35,36]. Depending on the type of equipment used, parameters with different accuracy and resolution can be obtained. WiDeo [32] customized full-duplex WiFi joint estimation of ToF and AoA of the main reflector. WiTrack [37] develops a wideband FMCW radar to estimate the ToFs of the reflected signals accurately, but special equipment is needed.

There are other systems that can recognize more subtle activities, such as EQ-Radio [38], which can infer the emotions of people from radio frequency signals reflected by the human body. WiSleep [39] supports non-invasive breathing monitoring on a single wireless link with a directional antenna. UbiBreathe [40] uses off-the-shelf WiFi devices to monitor breathing frequency and detect some forms of apnea. In addition, a system that uses radio wave signals [41] to learn the sleep stage has recently been proposed.

2.2. Gesture Recognition System

With the rapid development of WiFi-based activity recognition technology, gesture recognition has gradually become a research hotspot. WiFinger uses time series Discrete Wavelet Transform coefficients combined with CSI values to recognize gestures with different numbers of finger stretches and folds [9]. WiDraw tracks the hands of users by monitoring changes in signal strength arriving from different angles [7]. WiVi [33] uses the inverse synthetic aperture radar method to track human body movements, achieving the radar-like vision and simple through-wall gesture communication. WiSee [42] is a DopLink-based fine gesture recognition system [43,44] uses the Doppler frequency shift in the narrowband OFDM signal extracted to recognize nine different human gestures. WiKey [13] uses unique patterns in CSI to identify the keys of a finger.

Although these solutions provide high accuracy, they all require gesture recognition at a specified location that is relatively fixed to the transmit and receive links, and most solutions require special hardware or controlled environments. Therefore, the research of the environment and location-independent gesture recognition has gradually come into people's field of vision. WiAG proposed a translation function to make gesture recognition independent of position and orientation [8]. Widar 3.0 [27] proposes a cross-domain gesture recognition system that achieves gesture recognition in different locations and different environments. However, such schemes usually rely on high-density deployments, which is impractical for large-scale deployments. The transmit–receive end is poor in mobility. When the position of the person changes, the position and orientation of the person need to be located first according to the send-receive link. On the basis of this, the cumulative error usually occurs. However, the mobile terminal can solve these problems very well. When the person moves, the receiving end can change according to the change of the person's position. For example, WiGest [16] is based on the RSSI of the mobile terminal. However, WiGest only uses RSSI, which greatly limits its gesture recognition accuracy and resistance to environmental interference. According to the latest nexmon, our system can extract 234 CSI subcarriers on the mobile phone side and has a strong location and environmental adaptability while ensuring high recognition accuracy.

3. System Design

Figure 1 shows the architecture and workflow of MDGest. The collected CSI signals are processed by three key components: gesture filter, critical subcarrier screening, and gesture recognizer. The signals not related to gesture operations are filtered out through gesture filter.. Simultaneously, after a gesture operation is detected, the MDGest system will segment the signal so that each segment corresponds to a gesture input. The gesture recognizer performs a recognition operation. The algorithm extracts the main mode features of each group of gestures, stores them in the form of a configuration file, and uses an improved DTW algorithm that can adapt to the individual differences in the speed and duration of gesture operations not synchronized to match pre-stored profiles.

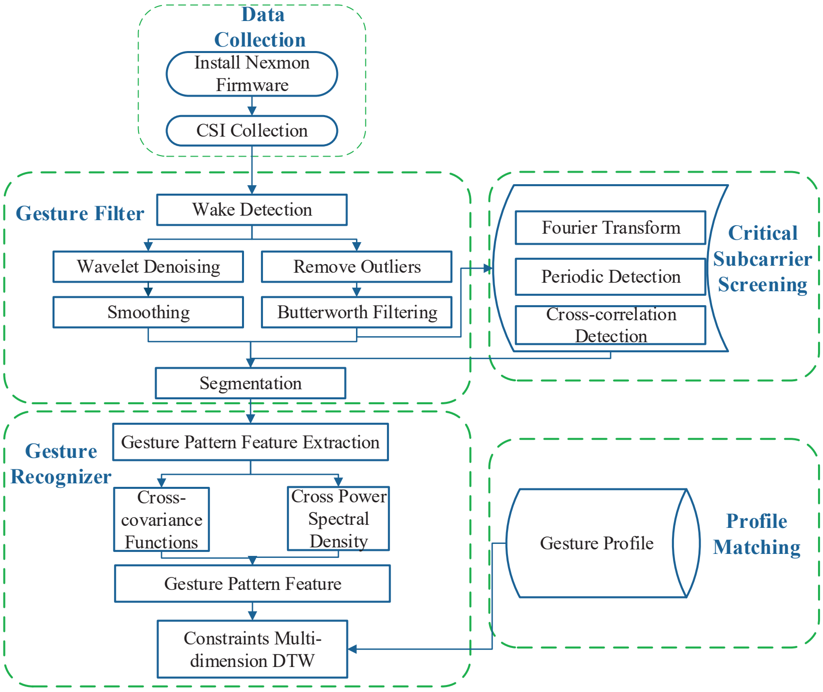

Figure 1. MDGest architecture.

3.1. Data Collection

We implemented MDGest on Android smartphones (Nexus 5) and commercial routers. We turn on monitor mode by installing the nexmon firmware on Nexus 5 phones, accessing the physical layer transmission information, and store physical layer CSI by creating new UDP frames. Then, we upload them to the host, using a Nexus 5 mobile phone to collect CSI with a bandwidth of 80 MHz in the 5 GHz of the commercial router. The total number of subcarriers collected can reach 256, and the number of available subcarriers is 234 after removing the guard bands, empty subcarriers, and pilot frequency signals. Comparing with gesture recognition using an Intel 5300 network card to collect only 30 subcarriers under a single link, MDGest dramatically improves the sensitivity and accuracy of gesture recognition.

CSI is the channel property of a communication link. It describes the shadow, multipath propagation, distortion and other information that wireless signals experience during propagation. Wireless channels can be described by channel frequency response in terms of amplitude–frequency characteristics and phase–frequency characteristics. The channel frequency response can be expressed using the following formula:

$$\mathrm{H(k)} = \|\mathrm{H(k)}\| e^{j\angle H(k)}. \tag{1}$$

$H(k)$ represents the CSI of the subcarrier k.$\|H(k)\|$ and $\angle H(k)$ are the amplitude response and the phase response of subcarrier k, respectively. In this paper, the proposed MDGest is based on the amplitude of the CSI.

3.2. Gesture Filter

The gesture filter removes signals not related to gesture operations and passes signals caused by gesture operations to the next stage. We set the wake-up operation part before gesture recognition starts. After the MDGest system recognizes the wake-up operation, it starts to perform the filtering operation. The wake-up operation can quickly wake up the system, thereby ensuring a low-energy sleep state to reduce energy consumption during the non-wake phase.

3.2.1. Wake Detection

To correctly determine when a user is performing a gesture, we set up a wake-up detection. The device is woken up when the user puts a hand near the receiver and then making n fast up and down gestures. Quick up and down gestures can cause rapid changes in energy amplitude. We detect the start of the wake-up action by setting a threshold and then start searching for n-1 periodic energy changes. In order to better set the adaptive threshold, we used a Nexus 5 mobile phone to perform multiple sets of up and down gesture calibration experiments at a fast speed. As shown in the Figure 2, the amplitude produced by the gesture shows an evident periodic change, and the peak value is about to be determined. Such periodic peak changes occur when the gesture begins to move up and down quickly. Therefore, MDGest uses this unique periodic amplitude change as a wake-up operation. In order to set a user-friendly wake-up operation, we chose a combination of up and down gestures with n of 4. In the experimental evaluation section, we will introduce the reasons for choosing 4.

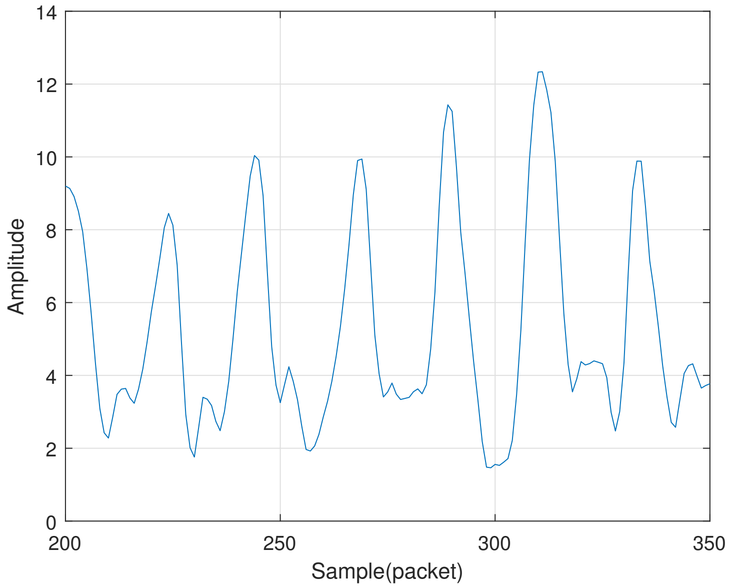

Figure 2. Rapid changes in amplitude.

3.2.2. Filtering

We first employ the Hampel algorithm to perform outlier removal on the original signal and then use a smoothing filter algorithm to smooth the signal. For each sample of the input vector, the Hampel function calculates the median of the window consisting of the sample and six surrounding samples, three on each side. The standard deviation of the median of each sample pair is estimated by using the median absolute value. If a sample is more than three standard deviations away from the median, the sample is replaced with the median. After that, a Butterworth bandpass filter is adopted to remove signals not related to gesture operations. Butterworth filter is chosen for bandpass filtering because it has the maximally flat frequency response in the passband and will not cause the largest distortion to the gesture signal. Finally, wavelet denoising is used to remove the residual noise components further while keeping the detailed patterns of the CSI. Figure 3 shows the original signal and the signal after filtering and denoising.

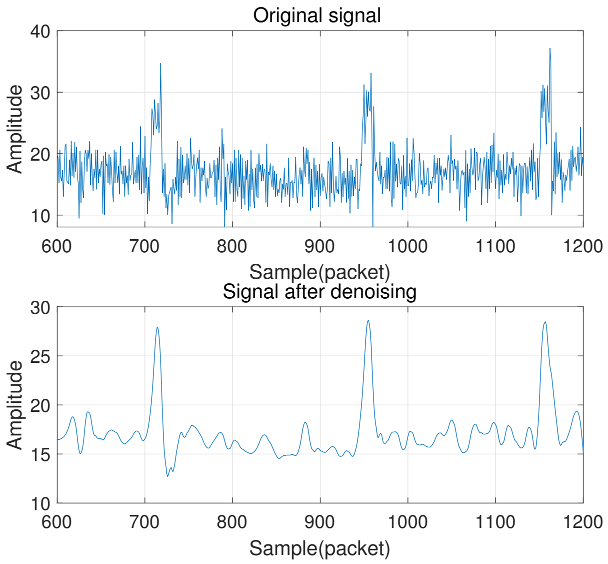

Figure 3. The original signal and the signal after filtering and denoising.

3.2.3. Segmentation

The system requires the user to have a short static interval after the wake-up action as a marker signal. We accumulate amplitude differences between adjacent time points. The cumulative value is then compared with the threshold value of the sentinel signal. We segment the gesture signal by detecting the two sentinel signals before and after. As shown in Figure 4, this algorithm is used to process the acquired CSI signals.

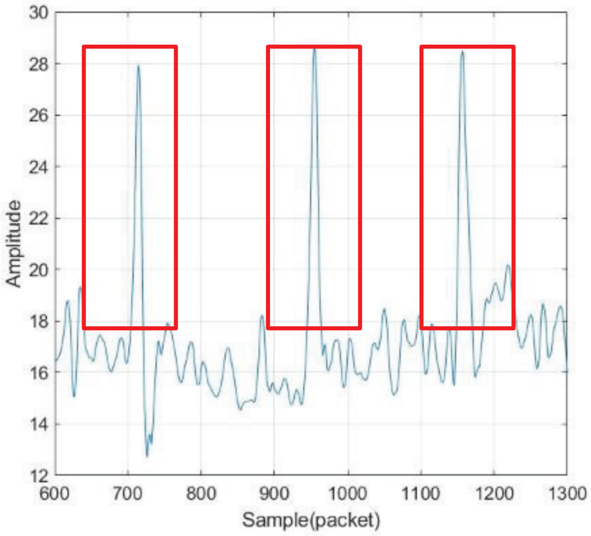

Figure 4. Segment the sample.

3.3. Key Subcarriers Selection

The selection of crucial subcarriers is mainly divided into the following steps: Fourier transform removes subcarriers whose frequencies are not in the frequency range of gesture recognition; periodic detection screens out subcarriers that do not correspond to the number of peaks and the number of gestures; cross-correlation detection filters out subcarriers with high correlation between different gestures. Through the screening of the above three links, we have selected the key subcarriers that are most sensitive to gesture recognition and assigned weights according to the degree of cross-correlation between these subcarriers for the gesture recognizer.

3.4. Gesture Recognizer

We expect that our system can not only perform gesture recognition at a fixed relative position to the transmitting end but also perform gesture recognition after the position changes of users, adapting to individual differences between different users. Therefore, we need unique characteristics to describe each gesture that a user operates.

3.4.1. Gesture Mode Feature Extraction

Operating the same gesture, different users, or the same users will have different operating speeds and motion amplitudes, and the change of the users position will also cause the same gesture CSI to appear differently. To remove these individual and positional differences, we use Cross Covariance Function (CCF) and Cross Power Spectral Density (CPSD) to extract the main mode features of gestures . The CCF is defined as follows:

$$R_{xy}(t_1, t_2) \overset{\text{def}}{=} \mathrm{E}\{x(t_1)y^*(t_2)\}. \tag{2}$$

$$\begin{aligned} C_{xy}(t_1, t_2) &\overset{\text{def}}{=} \mathrm{E}\Big\{[x(t_1) - \mu_x][y(t_2 - \tau) - \mu_y]^*\Big\} \\ &= R_{xy}(\tau) - \mu_x \mu_y^*, \end{aligned} \tag{3}$$

where R_{xy} and C_{xy} are Cross Correlation Function and Cross Covariance Function of x and y, which are two CSI measurements. u is their mean value. τ represents the time difference between the two CSI measurements, called lag. The CCF can extract the common parts between the two CSI measurements and suppress the non-common parts. We perform CCF calculations on the CSI measurements of the same gesture to extract common parts.

The CCF involves a multiplication between two different CSI measurements. These two CSI measurements minus the mean have a common part and a non-common part, and the multiplication of the common part always takes the same sign, so that the part is strengthened and retained. In contrast, the non-common parts of the two CSI measurements are random. Their products are sometimes positive and sometimes negative. After the average operation of mathematical expectations, they tend to "cancel" each other. This means that the CCF can extract the common parts between the two CSI measurements and suppress the non-common parts. We perform CCF calculations on the CSI measurements of the same gesture to extract common parts.

The CCF describes the statistical properties of the CSI measurements in the time domain. However, it belongs to the time domain characteristic and cannot reflect the change of the gesture in the frequency domain. So we introduced CPSD, which describes the statistical properties of CSI measurements in the frequency domain.

The CPSD is defined as the Fourier transform of the CCF, as follows:

$$P_{xy}(f) = \int_{-\infty}^{\infty} C_{xy}(\tau) e^{-j2\pi f\tau} \mathrm{d}\tau. \tag{4}$$

where $P_{xy}(f)$ is the CPSD. The real part of CPSD is called the in-phase spectrum, and the imaginary part becomes the orthogonal spectrum, which is recorded as:

$$P_{xy}(f) = \left|P_{xy}(f)\right| \exp\left[j\phi_{xy}(f)\right]. \tag{5}$$

$$\dot{\phi}_{xy}(f) = \frac{\mathrm{d}}{df}\phi_{xy}(f). \tag{6}$$

where $\left|P_{xy}(f)\right|$ and $\phi_{xy}(f)$ respectively represent the amplitude and phase of CPSD; $\dot{\phi}_{xy}(f)$ is called group delay.

After calculating and intersecting CCF and CPSD of the CSI, the primary mode characteristics of the same gesture signal in the time and frequency domains are obtained and stored in a library.

3.4.2. Operation Identification

The MDGest system uses an improved DTW algorithm to classify gestures. DTW algorithm is a nonlinear warping method that combines time warping and distance measurement. The DTW algorithm calculates the minimum distance for two time series of different lengths. The smaller the distance between the two-time series is, the more similar the two-time series are.

When using the DTW algorithm for classification, it is necessary to test the distance between gesture sequences and template gesture sequences stored in the library. The category of the template sequence with the smallest distance is the gesture category of the test sequence.

The description of the test sequence and template sequence is shown in Equations (6) and (7):

$$E(M) = \{e_1, e_2, \cdots, e_M\}. \tag{7}$$

$$H(N) = \{h_1, h_2, \cdots, h_N\}. \tag{8}$$

$E(M)$ and $H(N)$ respectively represent test sequences of M data points and template sequences of N data points after feature extraction and data processing. Euclidean distance was used to calculate the distance between two sequence data points. The distance between the ith data e_i of the test sequence and the jth data h_j of the reference template sequence was calculated as shown in Equation (8):

$$d\left(e_i, h_j\right) = \sqrt{\sum_{l=1}^{L}\left(e_{il} - h_{jl}\right)^2}. \tag{9}$$

We calculate the Euclidean distance of the test sequence and the template sequence of data points to create a $M \times N$ distance matrix D. The value (e_i, h_j) of the element $d(e_i, h_j)$ represents the Euclidean distance of the data point e_i and h_j. We use the DTW algorithm to calculate the distance between the two sequences, essentially looking for a suitable regular function $j = f(i)$, and satisfy the function type (9):

$$P(E, H) = \min_{f(i)} \sum_{i=1}^{M} d\left(e_i, h_{f(i)}\right). \tag{10}$$

$P(E, H)$ is the optimal matching distance between the test sequence $E(M)$ and the template sequence $H(N)$. The principle of the DTW algorithm is that the matrix D is to find a path from the starting point (e_1, h_1) to the ending point (e_M, h_N), and the cumulative distance of the path is the smallest. In order to achieve the optimal cumulative distance, the warping function of the DTW algorithm needs to meet the constraints of the global constraint and the local constraint. Among them, the local constraints that the warping function needs to satisfy are as follows:

(1) End point constraint, which requires that the starting point and ending point of the two sequences are consistent.

$$\begin{cases} f(1) = 1 \\ f(N) = M \end{cases}. \tag{11}$$

(2) Monotonic constraint. The generation of gesture data has a sequence, and the regularization function must ensure that the matching path does not violate the generation chronological sequence of gesture data, so it must satisfy Equation (11).

$$f(n+1) \geqslant f(n). \tag{12}$$

(3) Constraint of continuity. In order to ensure the minimum loss of matching information, the warping function cannot skip any matching point. Solving the optimization problem through the DTW algorithm, the cumulative distance of the best path can be obtained as:

$$p\left(\mathrm{e}_i, h_j\right) = d\left(\mathrm{e}_i, h_j\right) + \min\begin{cases} p\left(\mathrm{e}_i - 1, h_j - 1\right) \\ p\left(\mathrm{e}_i - 1, h_j\right) \\ p\left(\mathrm{e}_i, h_j - 1\right) \end{cases}. \tag{13}$$

Among them, $p(e_i, h_j)$ represents the minimum cumulative distance of the path sought from point $(1, 1)$ to point (i, j) of matrix D, so the value of $p(e_M, h_N)$ is the minimum cumulative distance of the test sequence $E(M)$ and the template sequence $H(N)$, that is, the DTW distance between the two.

When the classic DTW algorithm is used to calculate the distance between two sequences, the DTW method requires calculation and storage of a larger matrix, and the calculation time complexity is $O(mn)$. In order to improve the calculation efficiency of the DTW algorithm to calculate the distance of the gesture sequence, we introduce a global constraint window in the sequence bending calculation to avoid invalid path searching. The Constraints Multi-dimension Dynamic Time Wrapping (CM-DTW) algorithm uses the Sakoe–Chiba window to reduce the calculation of the distance between invalid data points [45,46], thereby improving the efficiency of calculating the two sequences.

The Sakoe–Chiba global constraint can be understood as a restriction on the subscript in the point (e_i, h_j), so that $i - f \leqslant j \leqslant i + f$ is satisfied, and f is a constant.

Under the Sakoe–Chiba constraint, the two sequences of the test sequence $E(M)$ and the template sequence $H(N)$ calculate the matching and regular paths of the DTW distance as $c_1, c_2, \ldots, c_K$ (where $c_k = (i, j)$).

Since the length of the gesture data sequence may vary greatly, it is necessary to make certain constraints on the slope of the search path so that the subscript satisfies:

$$\frac{N * i}{M} - f \leqslant j \leqslant \frac{N * i}{M} + f. \tag{14}$$

The CM-DTW we adopted overcomes the problem of different gesture speeds and improves the calculation efficiency of DTW algorithm to calculate the distance of the gesture sequence, thereby providing a reliable measurement method for detecting the degree of matching between the CSI mode and multiple subcarriers.

4. Evaluation

We evaluated the accuracy and robustness of MDGest based on the distance between the receiving and transmitting ends, the distance between users and equipment, the direction of users, and the environments. A total of 16 volunteers were recruited, including six females and ten males, aged between 20 and 28. The experiments were conducted in three indoor environments, including a laboratory, a large hall, and an office with furniture such as tables and sofas. Figure 5 shows the environmental characteristics of different rooms. The laboratory is 6.5 m × 12 m. There are experimental desks and chairs on both sides of the wall, and the middle corridor is empty. The hall is 6 m × 4.8 m, and it is empty. The office is 6 m × 5 m. There is a set of office desks, chairs, sofas, bookcases and so on. As shown in Figure 5, we placed a router and a Nexus 5 phone in the empty space of three rooms for data collection.

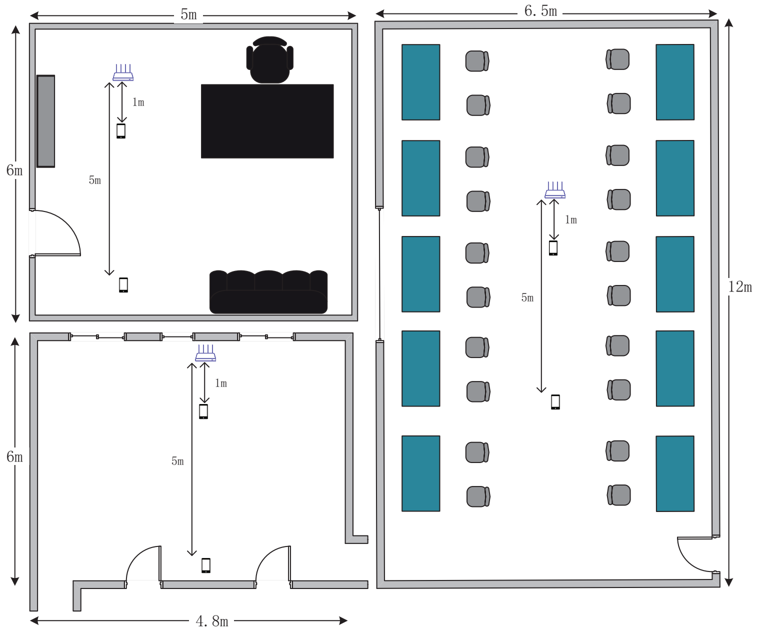

Figure 5. Layouts of three evaluation environments.

4.1. *Implementation and Experimental Setup*

We implemented the MDGest system on a Google Nexus 5 smartphone, which runs on Android 6.0. Furthermore, its firmware is modified with nexmon [21]. A TP-LINK TLWDR5620 wireless router is set as an AP. The AP possesses four omnidirectional antennas, two at 2.4 GHz and two at 5 GHz. We set the AP to IEEE 802.11ac mode at 5.21 GHz (channel number 42) on 80 MHZ bandwidth. We set the Nexus 5 smartphone as MP and turn on its monitor mode through the nexmon firmware. The nexmon firmware can access the physical layer transmission information, and store physical layer CSI by creating new UDP frames. Then, it will upload them to the host. All collected CSI are eventually stored in a PCAP file containing 256 complex pairs. However, the number of available subcarriers is 234 after removing the guard bands, empty subcarriers, and pilot frequency signals. The prototype system is implemented in MATLAB.

We fixed the position of the router at one end of the room, and the height was about 1 m, which is accorded with the typical indoor scene. Then, we used a Nexus 5 smartphone to collect gesture data at different distances and directions from the router. The types of gestures collected include push-pull, sweep, clap, slide, circle, and zigzag. Sixteen users collected 2160 gestures over three weeks, including different rooms, distances, and directions.

There is no fixed execution time for each gesture. The timing of each gesture depended on the habits of the volunteers. This makes the system more robust to users with different gesture execution speeds. In each scenario, each user continuously collects the same gesture for 20 times, and there is a 2–3 s static interval between each time, so that the system can better divide the data.

4.2. *Overall Accuracy*

The overall recognition accuracy of six gestures in three environments is shown in Figure 6 and Table 1, which indicates that the overall average accuracy of the MDGest system reaches 92%. Furthermore, we compared MDGest with WiGest and WiFinger, both gesture recognition systems using only one transmitter and one receiver. WiGest [16] is based on RSSI extracted from a smartphone and using Discrete Wavelet Transform for frequency feature extraction. WiFinger [9] is based on CSI extracted from a laptop equipped with an Intel 5300 network card, using Dynamic Time Warping for waveform comparisons. We compared the three approaches in a typical office environment. The Figure 7 results show that our system can provide high accuracy of gesture recognition in the three systems. Moreover, we also evaluated the performance of MDGest system in different environments. The Figure 8 shows the average recognition accuracy of the system in laboratory, hall, and office environments. The results show that the MDGest system can maintain an average accuracy of more than 90% in three different environments.

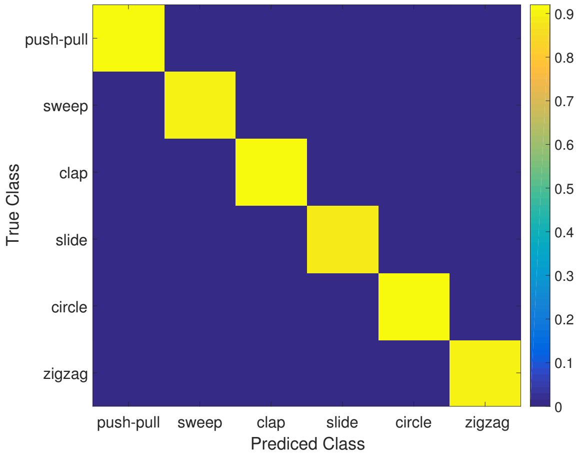

Figure 6. Confusion matrix of different operations.

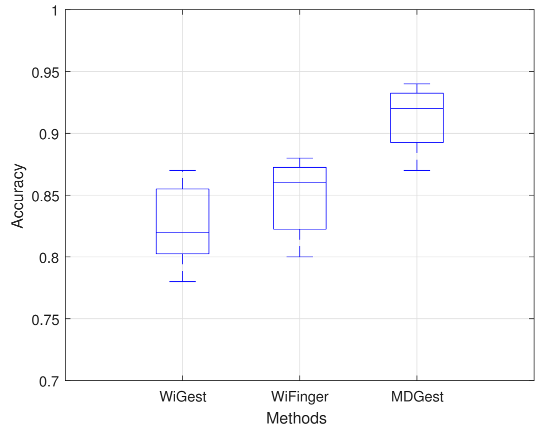

Figure 7. Accuracy of different methods.

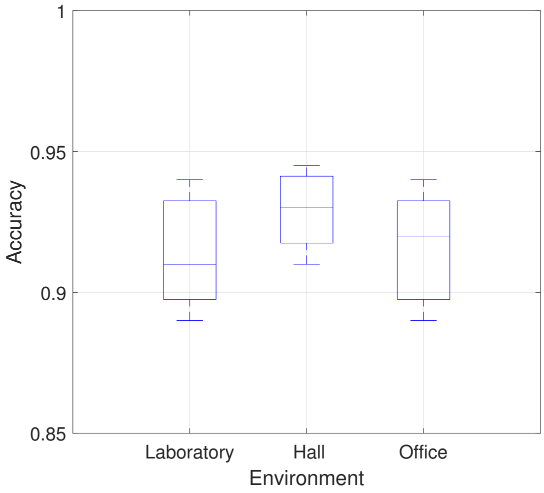

Figure 8. Accuracy of different environments.

Table 1. The average recognition accuracy of each gesture in the three environments.

EG	**Push-Pull**	**Sweep**	**Clap**	**Slide**	**Circle**	**Zigzag**
Laboratory	94.7%	89.5%	95.7%	87.0%	94.8%	88.4%
Hall	96.2%	90.3%	94.5%	89.5%	96.0%	89.1%
Office	94.1%	90.8%	92.4%	89.2%	95.0%	87.0%

4.3. Impact Factors

Since we are using a smartphone as the receiving end, we have to consider changes in the relative position of the receiving end and the transmitting end, such as distance, angle, etc. In addition, changes in the distance, direction, and environment between users and equipment also affect system recognition performance. We also consider the false detection rate caused by environmental factors in the absence of gestures. Multiple factors have been taken into consideration from distance, over orientation to environmental disturbance, which we think could affect the recognition accuracy of the system.

4.3.1. Impact of Distance Between Transmitting and Receiving ends

Since the location of the smartphone changes with the user, it is essential that the performance of the system remains stable at different distances from the smartphone to the AP. We evaluated the accuracy under the distances from 1 m to 5 m in 0.5 m steps between the transmitter and receiver. We fixed the distance between the user and the smartphone at 1 m.

As shown in Figure 9, the overall accuracy of the system is maintained above 90%. In the case of distances of 2.5 m and 3 m, the recognition accuracy of the system reaches up to 95%, which is different from our common sense that the closer the receiver and transmitter are, the higher the accuracy of gesture recognition is. We guess that this may be due to the fact that the CSI signal of the direct view path cannot fully cover all the motion information of the gesture when the distance between the receiving end and the transmitting end is too close.

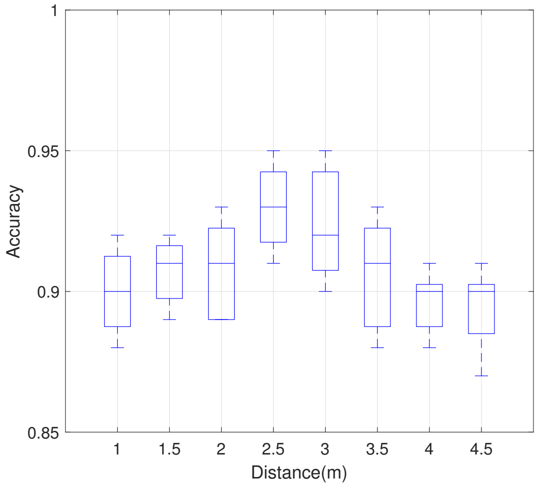

Figure 9. Impact of distance between transmitting and receiving ends on the recognition accuracy.

4.3.2. Impact of Distance between Hand and Device

When users perform gestures, the same gesture will lead to different CSI changes at different distances from the smartphone. Therefore, it is vital to evaluate system performance at different distances from the smartphone. The distance between the smartphone and the AP is set to 2.5 m. The user performs gestures in the direct path between smartphone and AP under distances from 20 cm to 160 cm in 20 cm steps between hand and smartphone.

Figure 10 shows the relationship between MDGest system's gesture recognition accuracy and the distance of the hand relative to the device. As the distance between the hand and device increases, the accuracy of recognition also increases, reaching an accuracy of 95% at 1 m. This is very intuitive. As the distance of the gesture from the device increases, the amplitude of the edge of the gesture also increases, resulting in a better signal-to-noise ratio and more accurate gesture detection. As the distance increases further, the accuracy decreases, but the overall accuracy remains 90%.

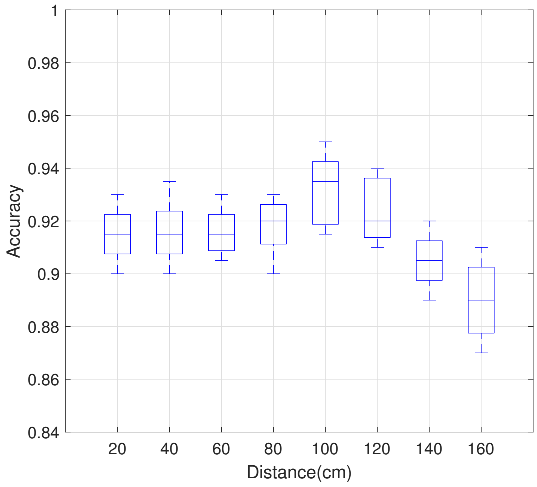

Figure 10. Impact of distance between hand and device on the recognition accuracy.

4.3.3. Impact of User Orientation around the Device

Although the user is in the direct path of the smartphone and router by default when performing gestures, we still evaluate the performance of the system when the user performs gestures in other relative areas of the device. In this section, we show the effect of the user performing gestures in different orientation areas around the device on the accuracy when the device's position and orientation are fixed. We evaluated the performance of the user in four directions of the device, which is shown in Figure 11. The center of region A1, A2, A3, and A4 is 1 m away from the smartphone center. The distance between the mobile device and the AP is set to 2.5 m.

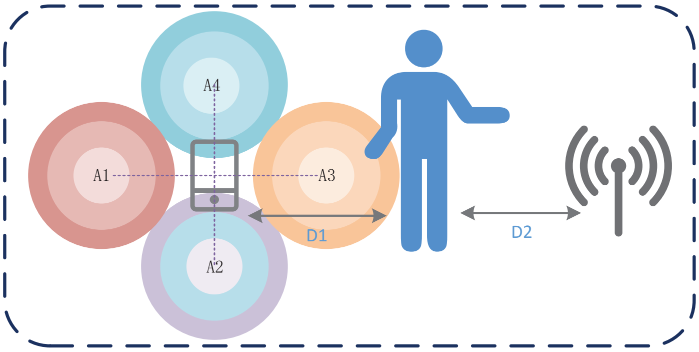

Figure 11. Different positions and regions of gestures input.

As shown in Figure 12, the recognition accuracy for the four regions is all above 90%, which indicates the MDGest system can maintain high accuracy in the surrounding areas. Among the four regions, region A3 has the highest recognition accuracy because the location of region A3 is exactly in the direct path of smartphone and AP, and gesture motion information can be better captured by CSI. In contrast, the other three regions can only reflect signals by the gesture to cause CSI changes. Therefore, the identification accuracy is lower than the region A3.

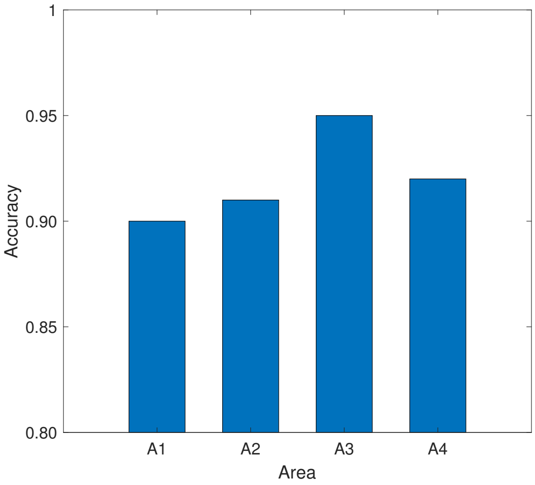

Figure 12. Impact of input region on recognition accuracy.

4.3.4. Impact of Environmental Interference

The MDGest system realizes gesture recognition depending on the changes of CSI over the LOS path. Therefore, the gesture recognition accuracy of the system is seriously affected by the interference of people in or near the LOS path. Considering the existence of human interference in the real environment, we evaluated the robustness of the system to humans disturbances. We set the distance between the receiver and the transmitter to

2.5 m. The ambient interference we set up is an interfering person walking at different distances from the transceiver link's vertical line.

Figure 13 reflects the results, showing that when the distance of the interference user is less than 3 m, the accuracy will be seriously affected. When the interferer is more than 4 m away, the accuracy will be reduced in the absence of denoising. After the removal of environmental noise, the impact on accuracy is insignificant. In general, the system can maintain high accuracy after noise removal.

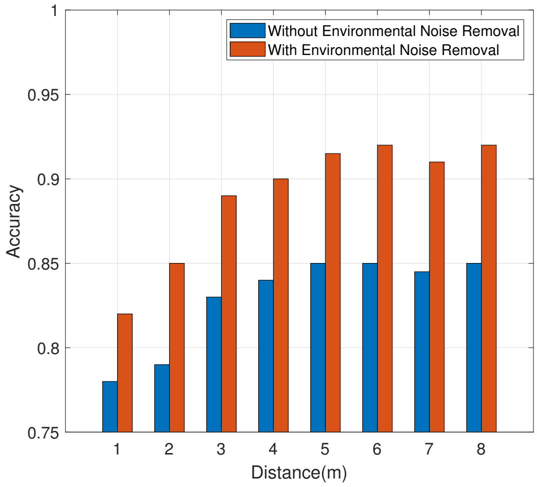

Figure 13. Impact of environmental interference.

4.3.5. False Detection Rate Without Gestures

As mentioned in Section 3.2.1, MDGest uses unique gestures with periodic changes as wake-up operations to reduce energy consumption. We evaluated the false detection rate of the MDGest system in the absence of actual gestures. We conducted experiments in the laboratory. There were seven students in the room for daily activities. There were other visitors coming in and out during the period, and the number of people in the room can reach up to 10 at most.

Figure 14 shows the average false detection rate per hour. We have observed that accuracy increases as the number of wake-up actions increases. When the number of wake-up actions is 4, it can reach about 0 error detections per hour. Nevertheless, four wake-up actions are too tedious for the user. Figure 14 shows that when the number of wake-up actions reaches three, the average false detection rate per hour can reach 3.4, which can meet the requirements of the system. So we used three wake-up actions in the paper.

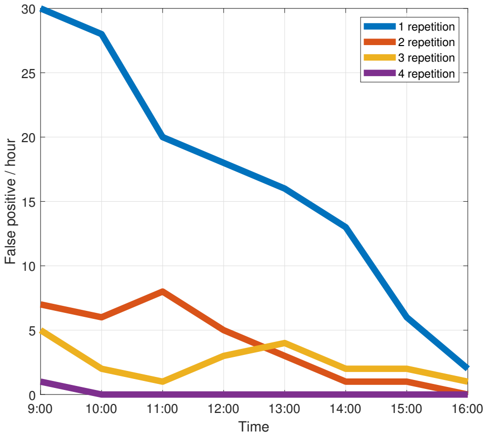

Figure 14. False detection rate in the absence of actual gestures.

5. Discussion

5.1. Why Use Smartphones for Gesture Recognition?

First, when we use a PC equipped with Intel 5300 network card or Atheros 9580 network card to recognize gestures, we need to walk to a certain range near the computer to get accurate gesture recognition. This is necessary because the location of the executing gesture must be in at least one Los link so that the WiFi signal can accurately capture the signal changes caused by the gesture. It is difficult to achieve high recognition accuracy for gestures that are completely out of the Los link. However, in daily life, almost everyone carries a smartphone, so we can take out the smartphone for gesture recognition at any time, instead of preparing a laptop in advance and walking near the laptop or bringing it to us for gesture recognition. When we take out the phone, we can easily put ourselves in the Los link near the phone. From this point of view, we believe that the advantages of smartphones are far greater than laptops. Second, supporting 802.11ac and allowing for bandwidths of up to 80 MHz is a huge advantage. Using the nexmon CSI extraction tool can extract up to 256 subcarriers, which is currently only possible with 802.11ac and 80 MHz bandwidth.

5.2. The Limitations of Nexmon Firmware

The current nexmon firmware is applicable to smartphones, Raspberry Pi B3+/B4, and Asus RT-AC86U with Broadcom bcm43xx series chips. For smartphones, the nexmon firmware can support many smartphone brands, including Samsung, Apple, Huawei, Google, SONY, etc. However, only some of these smartphone brands are equipped with Broadcom bcm43xx series WiFi chip. In addition, there are many other smartphone brands that cannot be supported. The previous Linux 802.11n CSI Tool [14] and Atheros CSI Tool [15] are in cooperation with Intel and Qualcomm to obtain CSI. Nexmon firmware was developed through reverse engineering of Broadcom bcm43XX series WiFi chips with open-source firmware, so if nexmon wants to support more smartphones, it will need more WiFi chip vendors with open-source firmware. However, we believe that with the rapid development of WiFi-based wireless sensing technology, more and more chip manufacturers will expand the opening of the underlying CSI of the chip.

5.3. The Energy Consumption Problem

Realizing gesture recognition on smartphones can provide users with a more convenient gesture recognition experience. However, the problem of energy consumption will still bring great trouble to users. The main energy sources of the MDGest system in smartphone deployment are from two aspects. One is that the phone needs to receive the packet all the time and determine whether it is a gesture performed by the user. The other is to run feature extraction and recognition algorithms. In the former case, we have reduced some energy consumption by setting up a wake-up detection method. For the latter, we will further optimize the feature extraction and recognition algorithm in future work to reduce the computational complexity and reduce energy consumption.

6. Conclusions

In this paper, we presented MDGest, a WiFi-based single-link gesture recognition system that employs CSI extracted from smartphones. MDGest leverages nexmon firmware to extract 234 CSI subcarriers from the smartphone in IEEE 802.11ac mode at 5.21 GHz (channel number 42) on 80 MHz bandwidth. Then MDGest extracts time domain and frequency domain features and exploits the improved CM-DTW algorithm to recognize the gestures. We implemented MDGest on off-the-shelf consumer smartphones and conducted comprehensive experiments. Experimental results show that MDGest can maintain high recognition accuracies across different environments, users, and locations. MDGest is robust to environmental and individual differences.

In future work, we will invite volunteers of different ages to participate in the experiment, so as to expand the coverage of experimental data and improve the robustness of the system to different age groups. Furthermore, we will continue to optimize feature extraction and gesture recognition algorithms to reduce algorithm complexity and system operation energy consumption.

Author Contributions: Conceptualization, T.L., C.S., P.L. and P.C.; methodology, T.L., C.S., P.L. and P.C.; software, T.L., C.S. and P.L.; validation, T.L., C.S., P.L. and P.C.; formal analysis, T.L., C.S., P.L. and P.C.; investigation, T.L., C.S., P.L. and P.C.; resources, P.C.; data curation, T.L., C.S., P.L. and P.C.; writing—original draft preparation, T.L., C.S., P.L. and P.C.; writing—review and editing, T.L., C.S., P.L. and P.C.; visualization, T.L., C.S., P.L. and P.C.; supervision, T.L., C.S., P.L. and P.C.; project administration, T.L., C.S., P.L. and P.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Natural Science Foundation of China under Grant No.51774282

Informed Consent Statement: All participants in the experiment were informed and consented to the publication of the paper.

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

References

1. Gkioxari, G.; Girshick, R.; Dollár, P.; He, K. Detecting and recognizing human-object interactions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, UT, USA, 18–23 June 2018; pp. 8359–8367.
2. Li, T.; Liu, Q.; Zhou, X. Practical human sensing in the light. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 26–30 June 2016; pp. 71–84.
3. Wang, M.; Ni, B.; Yang, X. Recurrent modeling of interaction context for collective activity recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA, 21–26 July 2017; pp. 3048–3056.
4. Bulling, A.; Blanke, U.; Schiele, B. A tutorial on human activity recognition using body-worn inertial sensors. *ACM Comput. Surv. (CSUR)* **2014**, *46*, 1–33. [CrossRef]
5. Guan, Y.; Plötz, T. Ensembles of deep lstm learners for activity recognition using wearables. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2017**, *1*, 1–28. [CrossRef]
6. Shen, S.; Wang, H.; Roy Choudhury, R. I am a smartwatch and i can track my user's arm. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 26–30 June 2016; pp. 85–96.
7. Sun, L.; Sen, S.; Koutsonikolas, D.; Kim, K.H. Widraw: Enabling hands-free drawing in the air on commodity wifi devices. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 77–89.
8. Virmani, A.; Shahzad, M.Position and orientation agnostic gesture recognition using wifi. In Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services, Niagara Falls, NY, USA, 19–23 June 2017; pp. 252–264.
9. Tan, S.; Yang, J. WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition. In Proceedings of the 17th ACM International Symposium on Mobile Ad Hoc Networking And Computing, Paderborn, Germany, 5–8 July 2016; pp. 201–210.
10. Yu, N.; Wang, W.; Liu, A.X.; Kong, L. QGesture: Quantifying gesture distance and direction with WiFi signals. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–23. [CrossRef]
11. Venkatnarayan, R.H.; Page, G.; Shahzad, M. Multi-user gesture recognition using WiFi. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018; pp. 401–413.
12. Shahzad, M.; Zhang, S. Augmenting user identification with WiFi based gesture recognition. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2018**, *2*, 1–27. [CrossRef]
13. Ali, K.; Liu, A.X.; Wang, W.; Shahzad, M. Keystroke recognition using wifi signals. In Proceedings of the 21st Annual International Conference on Mobile Computing and Networking, Paris, France, 7–11 September 2015; pp. 90–102.
14. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool release: Gathering 802.11 n traces with channel state information. *ACM Sigcomm Comput. Commun. Rev.* **2011**, *41*, 53. [CrossRef]
15. Xie, Y.; Li, Z.; Li, M. Precise power delay profiling with commodity Wi-Fi. *IEEE Trans. Mob. Comput.* **2018**, *18*, 1342–1355. [CrossRef]
16. Abdelnasser, H.; Youssef, M.; Harras, K.A. Wigest: A ubiquitous wifi-based gesture recognition system. In Proceedings of the 2015 IEEE Conference on Computer Communications (INFOCOM), Hong Kong, China, 26 April–1 May 2015; pp. 1472–1480.
17. Zhang, J.; Tang, Z.; Li, M.; Fang, D.; Nurmi, P.; Wang, Z. CrossSense: Towards cross-site and large-scale WiFi sensing. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 305–320.
18. Jiang, W.; Miao, C.; Ma, F.; Yao, S.; Wang, Y.; Yuan, Y.; Xue, H.; Song, C.; Ma, X.; Koutsonikolas, D.; et al. Towards environment independent device free human activity recognition. In Proceedings of the 24th Annual International Conference on Mobile Computing and Networking, New Delhi, India, 29 October–2 November 2018; pp. 289–304.

19. Zheng, Y.; Zhang, Y.; Qian, K.; Zhang, G.; Liu, Y.; Wu, C.; Yang, Z. Zero-effort cross-domain gesture recognition with Wi-Fi. In Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services, Seoul, Korea, 17–21 June 2019; pp. 313–325.
20. Schulz, M.; Link, J.; Gringoli, F.; Hollick, M. Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi. In Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services, Munich, Germany, 10–15 June 2018; pp. 256–268.
21. Gringoli, F.; Schulz, M.; Link, J.; Hollick, M. Free Your CSI: A Channel State Information Extraction Platform For Modern Wi-Fi Chipsets. In Proceedings of the 13th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, Los Cabos, Mexico, 25 October 2019; pp. 21–28.
22. Han, C.; Wu, K.; Wang, Y.; Ni, L.M. WiFall: Device-Free Fall Detection by Wireless Networks. In Proceedings of the IEEE INFOCOM 2014—IEEE Conference on Computer Communications, Toronto, ON, Canada, 27 April–2 May 2014.
23. Yan, W.; Jian, L.; Chen, Y.; Gruteser, M.; Liu, H. E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures. In Proceedings of the 20th Annual International Conference on Mobile Computing and Networking, Maui, HI, USA, 7–10 September 2014.
24. Fang, B.; Lane, N.D.; Zhang, M.; Kawsar, F. HeadScan: A Wearable System for Radio-Based Sensing of Head and Mouth-Related Activities. In Proceedings of the IEEE 2016 15th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Vienna, Austria, 11–14 April 2016; pp. 1–12.
25. Fang, B.; Lane, N.D.; Zhang, M.; Boran, A.; Kawsar, F. BodyScan: Enabling Radio-based Sensing on Wearable Devices for Contactless Activity and Vital Sign Monitoring. In Proceedings of the 14th Annual International Conference on Mobile Systems, Applications, and Services, Singapore, 25–30 June 2016.
26. Li, X.; Zhang, D.; Lv, Q.; Xiong, J.; Li, S.; Zhang, Y.; Mei, H. IndoTrack: Device-free indoor human tracking with commodity Wi-Fi. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2017**, *1*, 1–22. [CrossRef]
27. Qian, K.; Wu, C.; Yang, Z.; Liu, Y.; Jamieson, K. Widar: Decimeter-level passive tracking via velocity monitoring with commodity Wi-Fi. In Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Chennai, India, 10–14 July 2017; pp. 1–10.
28. Wang, W.; Liu, A.X.; Shahzad, M.; Ling, K.; Lu, S. Device-free human activity recognition using commercial WiFi devices. *IEEE J. Sel. Areas Commun.* **2017**, *35*, 1118–1131. [CrossRef]
29. Adib, F.; Hsu, C.Y.; Mao, H.; Katabi, D.; Durand, F. Capturing the human figure through a wall. *ACM Trans. Graph. (TOG)* **2015**, *34*, 1–13. [CrossRef]
30. Adib, F.; Kabelac, Z.; Katabi, D. Multi-person localization via RF body reflections. In Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, Berkeley, CA, USA, 4–6 May 2015; pp. 279–292.
31. Adib, F.; Kabelac, Z.; Katabi, D.; Miller, R.C. 3D tracking via body radio reflections. In Proceedings of the 12th USENIX Symposium on Networked Systems Design and Implementation, Oakland, CA, USA, 4–6 May 2015; pp. 317–329.
32. Joshi, K.; Bharadia, D.; Kotaru, M.; Katti, S.WiDeo: Fine-grained Device-free Motion Tracing using {RF} Backscatter. In Proceedings of the 12th USENIX Conference on Networked Systems Design and Implementation, Oakland, CA, USA, 4–6 May 2015; pp. 189–204.
33. Adib, F.; Katabi, D. See through walls with WiFi! In Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM, Hong Kong, China, 12–16 August 2013; pp. 75–86.
34. Li, X.; Li, S.; Zhang, D.; Xiong, J.; Wang, Y.; Mei, H. Dynamic-music: accurate device-free indoor localization. In Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Heidelberg, Germany, 12–16 September 2016; pp. 196–207.
35. Bocca, M.; Kaltiokallio, O.; Patwari, N.; Venkatasubramanian, S. Multiple target tracking with RF sensor networks. *IEEE Trans. Mob. Comput.* **2013**, *13*, 1787–1800. [CrossRef]
36. Wang, J.; Jiang, H.; Xiong, J.; Jamieson, K.; Chen, X.; Fang, D.; Xie, B. LiFS: low human-effort, device-free localization with fine-grained subcarrier information. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 3–7 October 2016; pp. 243–256.
37. Chen, T.H.; Sou, S.I.; Lee, Y. WiTrack: Human-to-Human Mobility Relationship Tracking in Indoor Environments Based on Spatio-Temporal Wireless Signal Strength. In Proceedings of the 17th IEEE International Conference on Dependable, Autonomic and Secure Computing (DASC 2019), Fukuoka, Japan, 5–8 August 2019; pp. 788–795.
38. Zhao, M.; Adib, F.; Katabi, D. Emotion recognition using wireless signals. In Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking, New York, NY, USA, 3–7 October 2016; pp. 95–108.
39. Liu, X.; Cao, J.; Tang, S.; Wen, J. Wi-Sleep: Contactless sleep monitoring via WiFi signals. In Proceedings of the 2014 IEEE Real-Time Systems Symposium, Rome, Italy, 2–5 December 2014; pp. 346–355.
40. Abdelnasser, H.; Harras, K.A.; Youssef, M. UbiBreathe: A ubiquitous non-invasive WiFi-based breathing estimator. In Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing, Hangzhou, China, 22–25 June 2015; pp. 277–286.
41. Zhao, M.; Yue, S.; Katabi, D.; Jaakkola, T.S.; Bianchi, M.T. Learning sleep stages from radio signals: A conditional adversarial architecture. In Proceedings of the International Conference on Machine Learning, Sydney, Australia, 6–11 August 2017; pp. 4100–4109.

42. Pu, Q.; Gupta, S.; Gollakota, S.; Patel, S. Whole-home gesture recognition using wireless signals. In Proceedings of the 19th Annual International Conference on Mobile Computing & Networking, Miami, FL, USA, 30 September–4 October 2013; pp. 27–38.
43. Aumi, M.T.I.; Gupta, S.; Goel, M.; Larson, E.; Patel, S. DopLink: using the doppler effect for multi-device interaction. In Proceedings of the 2013 ACM International Joint Conference on Pervasive and Ubiquitous Computing, Zurich, Switzerland, 8–12 September 2013; pp. 583–586.
44. Kim, Y.; Ling, H. Human activity classification based on micro-Doppler signatures using a support vector machine. *IEEE Trans. Geosci. Remote. Sens.* **2009**, *47*, 1328–1337.
45. Li, X.; Xu, L.; Li, R. Dynamic Gesture Identity Authentication of Smart Phone Based on CM-DTW Algorithms. *Comput. Syst. Appl.* **2019**, *28*, 17–23.
46. Sakoe, H.; Chiba, S. Dynamic Programming Algorithm Optimization for Spoken Word Recognition. *IEEE Trans. Acoust. Speech Signal Process.* **1978**, *26*, 43–49. [CrossRef]