WiSign: Ubiquitous American Sign Language Recognition Using Commercial Wi-Fi Devices

LEI ZHANG and YIXIANG ZHANG, College of Intelligence and Computing, Tianjin University, China
XIAOLONG ZHENG, School of Computer Science, Beijing University of Posts and Telecommunications, China

In this article, we propose WiSign that recognizes the continuous sentences of American Sign Language (ASL) with existing WiFi infrastructure. Instead of identifying the individual ASL words from the manually segmented ASL sentence in existing works, WiSign can automatically segment the original channel state information (CSI) based on the power spectral density (PSD) segmentation method. WiSign constructs a five-layer Deep Belief Network (DBN) to automatically extract the features of isolated fragments, and then uses the Hidden Markov Model (HMM) with Gaussian mixture and Forward-Backward algorithm to recognize sign words. In order to further improve the accuracy, WiSign also integrates the language model N-gram, which uses the grammar rules of ASL to calibrate the recognized results of sign words. We implement a prototype of WiSign with commercial WiFi devices and evaluate its performance in real indoor environments. The results show that WiSign achieves satisfactory accuracy when recognizing ASL sentences that involve the movements of the head, arms, hands, and fingers.

CCS Concepts: • **Human-centered computing → Gestural input**;

Additional Key Words and Phrases: American Sign Language, sentence-level recognition, channel state information, deep belief network

ACM Reference format:
Lei Zhang, Yixiang Zhang, and Xiaolong Zheng. 2020. WiSign: Ubiquitous American Sign Language Recognition Using Commercial Wi-Fi Devices. *ACM Trans. Intell. Syst. Technol.* 11, 3, Article 31 (April 2020), 24 pages.
https://doi.org/10.1145/3377553

This work is supported by the NSFC under Grant No. 61772364 and No. 61672240. This work is also supported by National Instrument Program under Grant 2013YQ030915. This work is also partially supported by Tianjin Key Laboratory of Advanced Networking (TANK), College of Intelligence and Computing, Tianjin University, Tianjin China, 300350.
Authors' unit: L. Zhang and X. Zhang are with College of Intelligence and Computing, Tianjin Key Laboratory of Advanced Network Technology and Application, Tianjin University, Tianjin, China; L. Zhang is with State Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China; L. Zhang is also with Key Laboratory of Grain Information Processing and Control, Henan University of Technology, Henan, China; X. Zheng (corresponding author) is with Beijing Key Laboratory of Intelligent Telecommunications Software and Multimedia, Beijing University of Posts and Telecommunications, Beijing, China.
Authors' addresses: L. Zhang and Y. Zhang, College of Intelligence and Computing, Tianjin University, No.135 Yaguan Road, Jinnan, Tianjin, 300350, China; emails: {lzhang, Z840764787}@tju.edu.cn; X. Zheng, School of Computer Science, Beijing University of Posts and Telecommunications, No. 10 Xitucheng Road, Haidian, Beijing, China; email: zhengxiaolong@bupt.edu.cn.

2157-6904/2020/04-ART31 $15.00
https://doi.org/10.1145/3377553

1 INTRODUCTION

Sign language, as the common language of the deaf-mute community, is also an important bridge of communication between the deaf-mute and normal-hearing people. However, normal people without special training cannot understand sign language. The communication barrier between the two groups still exists. The deaf-mute, as an important community of each country, deserve a more convenient way to communicate with normal-hearing people. If sign language can be converted into speech through recognition techniques, it will greatly facilitate the communication between the deaf-mute community and normal-hearing people.

Existing sign language recognition approaches can be divided into two categories: device-based and device-free based recognition. Device-based signal language recognition methods include the vision-based [12, 22] and sensor-based solution [4, 14, 17, 21]. The vision-based approaches usually use a camera [12] or a Kinect [22] to record the video and exploit video processing techniques to recognize the sign languages. But the vision-based solutions are seriously affected by the light conditions and can only cover the areas where Line-Of-Sight (LOS) video is available. Besides, video processing requires a large amount of computation resources, which hinders the quick response. Sensor-based approaches leverage Leap Motion [17] and motion sensors embedded in dedicated gloves [9] and other wearable devices [32, 38] to detect the user motions, and then recognize the sign language. However, to accurately detect the motions, a user has to wear a lot of sensors because sign language can involve the motions of many body parts, such as the hand, arm, and fingers. Besides, the reading invasive sensors can be greatly influenced due to the body motions that are not related to sign language.

The recent advance of wireless sensing, which leverages the variations of Radio Frequency (RF) signal propagation to infer the human gesture, has shed light on device-free recognition. The RF-based recognition methods can provide ubiquitous service even in the Non-Line-Of-Sight (N-LOS) scenarios. Besides, they do not require users to wear any motion sensors on their bodies. Many RF-based activity recognition methods that use dedicated [2, 3, 7, 13, 23, 44] or commercial RF devices [5, 19, 20, 29, 33] have been proposed. But most of those systems only provide coarse-grained information [2, 3, 23, 29, 33, 36] or information of a specific body part such as a hand [7, 19, 20, 29]. For example, WiDraw [29] uses 25 densely deployed RF devices to identify large-scale hand movements. Wikey [5] and the system proposed in Ref. [13] are designed to recognize specific finger typing motions but fail to capture the motions of other body parts.

A few works [16, 18] focusing on RF-based sign language recognition have been proposed. WiFinger [16] is proposed to leverage k-NN with Dynamic Time Warping (DTW) to recognize ASL number words. SignFi [18] leverages CNN to extend the recognition ability to ASL words. However, most of the existing RF-based sign language recognition approaches focus on isolated word recognition instead of continuous sentence recognition. Sign language recognition of isolated words refers to the recognition of a single sign language word from a segmented Channel State Information (CSI) sequence in which no more than one word exists. On the other hand, the continuous word sign language recognition refers to the recognition of a series of sign language words that are combined according to certain grammar rules. More than one word may exist in the sampled sequence. How to segment the collected RF signals into segments that each corresponds one word is challenging because a sign word can be composed by various numbers of motions. RF-based continuous sentence recognition is still an open problem.

In this article, we propose WiSign, a novel sign language recognition system using continuous sentence recognition by commercial WiFi devices. We focus on American Sign Language (ASL), a widely used sign language. But achieving the idea is non-trivial. First, the CSI samples are noisy due to the environmental factors. How to adapt to different operating environments and identify

different locations is a challenge. Second, time domain segmentation is another problem of continuous sign language recognition. The common scheme of continuous sign language recognition is to decompose sentences into isolated words, which requires time domain segmentation. Time domain segmentation is not simple because there are a variety of transition actions, and they are difficult to detect. Moreover, time domain segmentation is taken as a pre-processing step, and if the segmentation is not accurate, it will lead to errors in the subsequent steps. Third, it is challenging to recognize the complicated sign language involving multiple body parts by a limited number of manually selected features. In addition, the effective features for different persons can be different due to the diversified sizes of hands, heights, and habits of performing the sign language. Such a diversity can also deactivate the manually selected features.

To address the above-mentioned issues, WiSign leverages deep learning techniques to achieve continuous sentence recognition. First, WiSign integrates a series of noise reduction methods to enable WiSign to achieve ideal recognition accuracy in different environments. Second, we find that even though the words in a sentence are continuous and the users usually pause for a short time when they perform a complete word. This observation inspires us to use power spectral density (PSD) to capture the transitory pause and then segment a sign sentence into sign words. Unfortunately, there will be a transition action in the middle when making another sign word after making a sign word. Since the transition action is hard to be separated from the action of the latter sign word, we adopt the cross-correlation algorithm to reduce the error caused by the transition action. The principle is to remove the transition gesture by making a cross-correlation between sentence-level sign language actions and template actions. Third, Stokoe in 1960 proposed that a symbol consists of three cheremes at the same time: tab (the position of the sign), dez (the handshape), and sig (the movement). In the ASL dictionary, the word "THROW," which is mainly created by the shape of hands, starts with an S-shaped hand, but ends with an H-shaped hand and the word "PARENT," which is mainly caused by the position of the sign, starting from the chin to the forehead. In addition, according to Stokoe's initial observation, there are some physical signs such as "YEAR," which is first a circular movement followed by a contact movement. This inspired us to use the Deep Belief Network (DBN) and Hidden Markov Model (HMM) with Gaussian mixture to extract features and predict the optimal state sequence. The HMM divides a sign action into a plurality of key poses, and extracts the feature vectors of the frames of the sign action through the DBN model to predict the state of the maximum probability of each frame. Finally, we also use the language model N-gram to further improve the recognition accuracy.

This article has three main contributions:

—We propose WiSign, a novel ASL recognition system that achieves the continuous sentence recognition. We propose a segmentation method based on power spectral density to capture the transitory pause between successive sign words and segmenting a sign sentence into isolated sign words. We also propose the method of using the cross-correlation function to reduce the error in time domain segmentation caused by the transition gesture.
—We combine the DBN and HMM with Gaussian mixture, which is a method of feature extraction and classification of time and space information. This hybrid model extracts effective features under different operating environments at multiple levels, avoiding inappropriate features selected by manual experience, and selecting the most likely hidden state sequence build models by using the Viterbi algorithm.
—We implement a prototype of WiSign with commercial WiFi devices and evaluate its performance in a real indoor environment. Experimental results show that in different environments, the rate reaches up to 92% and 87% with a personalized model and 69% with the general model for 30 users.

The rest of this article is organized as follows. We first present the related work in Section 2. We elaborate our designs in Section 3. We then present the evaluation results in Section 4 and, finally, conclude our work in Section 5.

2 RELATED WORK

Wearable sensor based approaches. Grobel et al. [6] established an isolated hand word recognition system. They use colored gloves to obtain the position, direction, and shape of a specific person's hand, and then use HMM to model the sign words. They achieve a recognition rate of 94% in the 262 sign language words. Gao et al. [11] utilize gloves and a position tracker to collect the motion data, and, for the first time, use a self-organizing feature map as a feature extractor to obtain the differential expression of low-dimensional features of the sign language samples. They realize 5,113 isolated words recognition with a recognition rate of 82.9%. They further use a simple circular network to segment continuous sign language words, and build a recognition system for continuous sentence recognition, which achieves a recognition rate of 86.3%. In Ref. [10], Gao et al. put forward the Transfer Motion Model (TMM) to transfer two adjacent hand words. With the sequential clustering, iterative segmentation, and HMM algorithms, they improve the recognition rate to 91.9%. All those wearable sensor based approaches require the users to wear dedicated devices on their bodies, which brings inconvenience to the users.

Computer vision based approaches. Lang et al. [15] leverage Kinect to collect the color information and depth mapping data to recognize the body motions of the user. And computer vision techniques are utilized to analyze the motions to recognize specific sign words. Sun et al. [28] also leverage Kinect but design a discriminant exemplar coding algorithm to extract features. Using AdaBoost, they achieve a recognition rate of 86.8% on 72 ASL sign words. Cameras can also be used to capture the digital images for user motion analysis [25]. With the help of depth information, the light and background interference in the scene can be reduced. The recognition accuracy of isolated words is more than 90% in Ref. [25]. However, computer vision based approaches require clear LOS images for accurate analysis. Besides, the heavy computation tasks of vision techniques require the high-performance computers to provide a quick response.

RF signal based approaches. Recent development of wireless networks promotes the application of using Radio Frequency (RF) signals to infer the human activity. The researchers are especially interested in the widely deployed WiFi. Many works have shown that Channel State Information (CSI) can recognize the motions of hands [1, 19, 23, 24, 29, 30] and fingers [16, 19, 40, 41]. Among the works, three papers focus on ASL recognition [1, 16, 19]. However, all three works only focus on the simple ASL words with only motions of hands and fingers. Besides, most of them deal with the isolated word recognition. Different from existing works, in this article, we focus on the continuous sentence recognition with the motions of arms, fingers, and hands.

3 DESIGN OF WISIGN

In this section, we first present an overview of WiSign and then introduce the design details of WiSign.

3.1 Overview

Figure 1 presents the overview of WiSign. In the environment where a wireless transmitter and receiver are deployed, the user stands in front of the receiver to perform an ASL sentence. The raw CSI is collected by the receiver. The CSI samples provided by commodity Intel 5300 Network Interface Controllers (NICs) are inherently noisy because of the internal state transitions in the transmitter and receiver WiFi NICs such as transmission power changes, transmission rate adaptation, and internal CSI reference level changes as well as the noise caused by the environment.

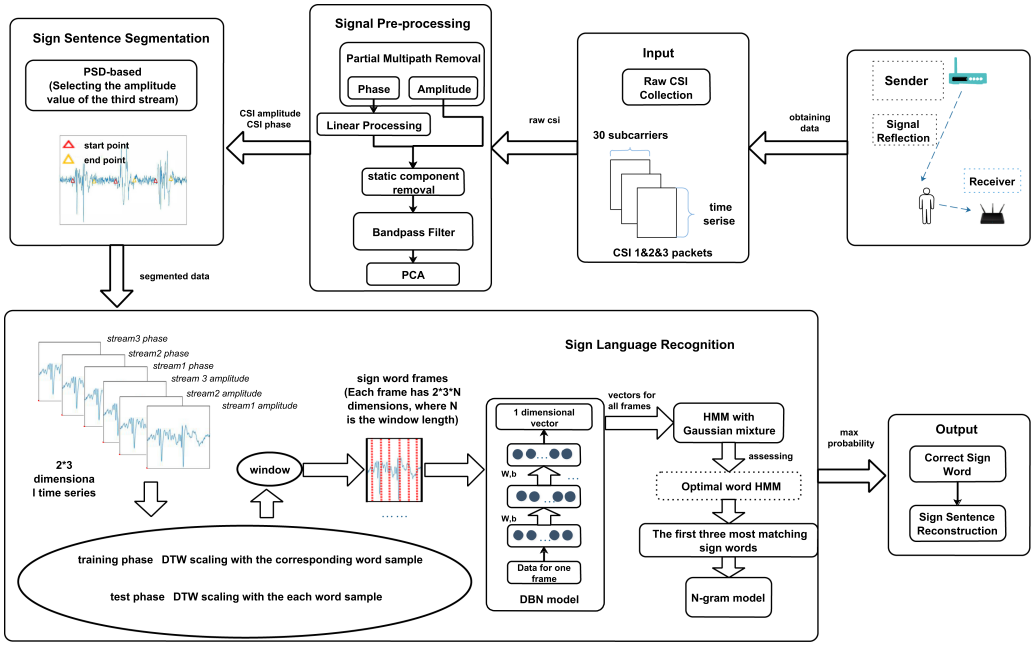


Fig. 1. Overview of WiSign.

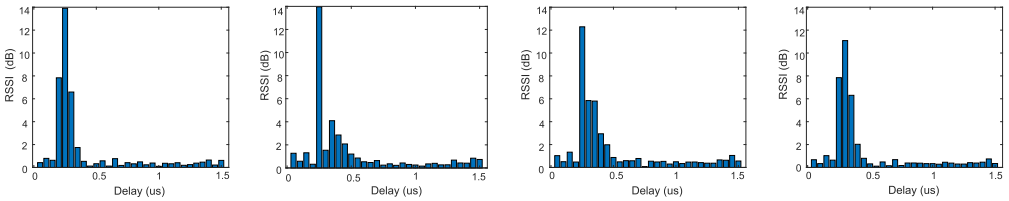
Hence, WiSign processes the raw CSI traces by a series of noise removal methods, including the partial multipath removal, static component removal, phase linear processing, bandpass filter, and principal component analysis. After noise removal, the filtered CSI traces are segmented by the sign sentence segmentation component to isolate the sign words. WiSign adopts a PSD-based segmentation method to find the beginning and end of the sign words. The segmented sign words are then forwarded to the sign language recognition component. In the training phase of the sign language recognition component, we use the DTW method with CSI waveform of the same label for the minimum distance alignment. This nonlinear mapping method makes the duration of the same sign scale consistently. In the test phase, we first make a cross-correlation between the test sample and the template to eliminate the transition gestures. Then, the waveform of the test sample after DTW scaling with each sign template is extracted. In WiSign, an improved DBN model is designed to extract the features of isolated words in each frame, and then the HMMs with mixed Gaussian are used to process. The ranked top three most likely sign words identified by probability are forwarded to the N-gram model, and the sign words are corrected according to the grammar. Finally, WiSign puts together all the corrected sign words to reconstruct the sign sentence and outputs the sentence by voice assistant on the smartphone or other interactive devices.

3.2 Data Collection

CSI tools [8] are developed to obtain a sampled version of Channel Frequency Response (CFR) in the form of CSI on Intel 5300 NIC, a commercial WiFi NIC. Specifically, each CSI sample contains the amplitudes and phases of the whole Orthogonal Frequency Division Multiplex (OFDM) subcarriers.

$$H(f) = \|H(f)\| e^{j\angle H(f)}, \tag{1}$$

where $H(f)$ represents the CSI of $K = 30$ subcarriers, and $\|H(f_k)\|$ and $\angle H(f_k)$ are the amplitude and phase of the k-th subcarrier, respectively. Hence, $H(f)$ can also be described in the following



(a) The time-domain histogram at 0.5m (b) The time-domain histogram at 1m (c) The time-domain histogram at 2m (d) The time-domain histogram at 3m

Fig. 2. The time-domain histogram at 0.5m, 1m, 2m, and 3m, respectively.

format:

$$H(f) = [H(f_1), H(f_2), \ldots, H(f_K)] \tag{2}$$

With the development of Multiple-Input and Multiple-Output (MIMO) [27] techniques, multiple antennas are applied to IEEE 802.11 *n* devices to leverage the spatial diversity of wireless communication.

To obtain more information, we use $M = 3$ antennas on the receiver and $N = 1$ antenna on the transmitter. Hence, the CSI data is a 1×3 matrix of CSIs $\{H_{mn}(f)\}_{M \times N}$, in which each element $\{H_{mn}(f)\}_{M \times N}$ is the CSI vector defined in Equation (2).

3.3 Signal Pre-processing

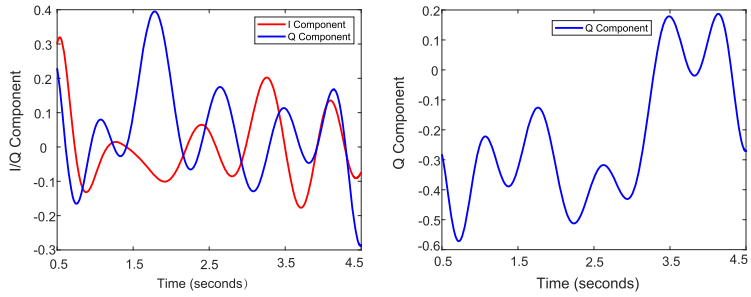
The raw CSI samples collected by CSI tools are intrinsically noisy, as shown in Figure 5(a). To recognize the fine-grained motions of ASL, it is necessary to provide a high-quality data for the classification model. Otherwise, the noise will confuse the classification model and lead to low accuracy. Hence, the collected raw CSI traces are firstly processed by our signal pre-processing component to remove the noise.

3.3.1 Partial Multipath Removal. Even though multipath provides rich information about the environment, it also brings too many noises in the CSI traces. Hence, we remove the long-delayed reflections, which do not have the appropriate information in our scenario. (It is mainly due to remote artificial noise interference and reflection of some objects and walls.) First, the raw CSI is calculated by Inverse Fast Fourier Transform (IFFT) to obtain the power delay curve that is approximated in the time domain. Then, the multipath components whose delay is not within the threshold interval are removed. However, if the threshold cannot be well determined, unnecessary noise may be introduced, and relevant information may be deleted from the removed multipath. We refer to a refined indoor propagation model [35], which represents the relationship between CSI and distance d.

$$d = \frac{1}{4\pi}\left[\left(\frac{c}{f_0 \times |CSI|}\right)^2 \times \sigma\right]^{\frac{1}{n}}, \tag{3}$$

where f_0 is the central frequency, d represents the distance between the transmitter and the receiver, c is the wave velocity, σ is the environment factor, and n is the path loss fading exponent. Both of the two parameters are dependent on distinctive indoor environments.

When d is different, the optimal interval of delay retention range should be found correspondingly. We determined the c and σ values of these two free space scenarios by collecting data from multiple locations in the rest hall and office. As shown in Figure 2, we get the time-domain histograms of different distance d. As can be seen from Figure 2, with the increase of distance d, the energy is shifted by the direction of the greater delay. This is because when the body near the receiver is away from the transmitter, the strong reflection path of the signal will be longer, and the energy



(a) I/Q waveforms (b) Imaginary part after LEVD

Fig. 3. I/Q waveforms and Q waveforms after LEVD.

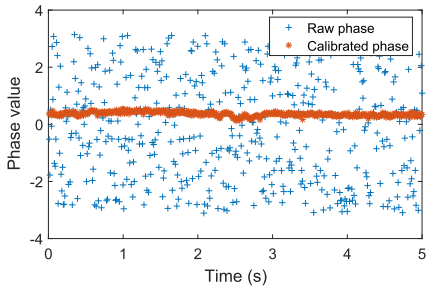


Fig. 4. The raw phase and calibrated phase of subcarrier #2.

of the direct reflection and short reflection paths will be reduced. After repeated observation and verification, we obtain the optimal delay retention interval between $0m \sim 1.2m$, $1.2m \sim 2m$, $2m \sim 3.5m$ as $[50us, 800us]$, $[100us, 1000us]$, $[150us, 1200us]$, respectively. As shown in Figure 6, through the comparison of Figure 6(a) and (b), we find that the characteristic of the waveform of the second sign language action becomes more obvious after the reasonable removal of the time domain components.

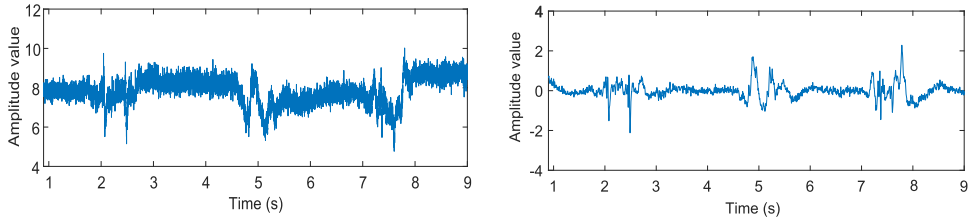
3.3.2 Phase Sanitization. When the signal encounters obstacle blockages, the CSI amplitude can be significantly weakened, providing limited information. But the phase of the signal with the periodical change over the propagation distance is more robust. Hence, WiSign also leverages the phase to extract information.

However, due to random noise and the unsynchronized time clock between the transmitter and receiver, raw phase information shows extreme randomness over the feasible field. What's worse, the phase is also sensitive to temperature and hardware conditions, bringing more noise. To eliminate the random noise, we leverage the linear transformation method similar to Ref. [34]. The linear transformation formula is as follows.

$$\tilde{\varphi}_i = \varphi_i - \frac{\varphi_n - \varphi_1}{k_n - k_1} k_i - \frac{1}{n} \sum_{j=1}^{n} \varphi_j, \tag{4}$$

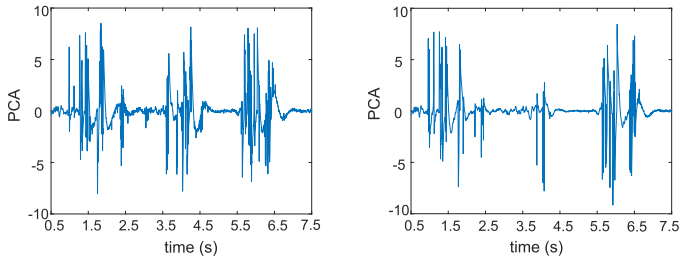
where φ_i and $\tilde{\varphi}_i$ are the raw phase and calibrated phase, k_i denotes the subcarrier index i^{th} (ranging from -28 to 28 in IEEE 802.11n) of the subcarrier, and n equals 56.

Figure 4 presents the raw phase (blue points) of the second subcarrier and the calibrated phase (red points) by our phase sanitization component. We can find that the raw phases are randomly distributed in the feasible range $[-\pi, +\pi]$. Directly using such a random, raw phase cannot provide



(a) Raw amplitude (b) After bandpass filtering

Fig. 5. Raw amplitude waveform and after a series of denoised waveforms.



(a) Waveform processed by partial multipath filtering (b) Waveform that has not been subjected to partial multipath filtering

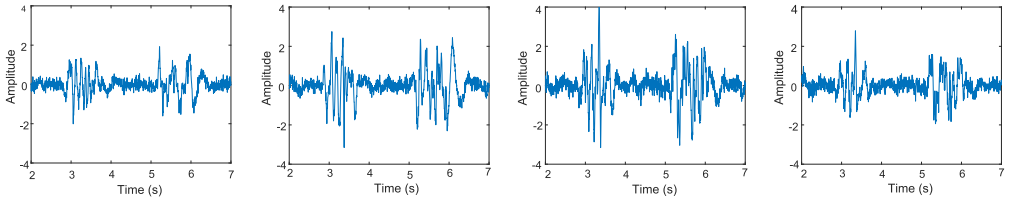
Fig. 6. The feature waveform affected by partial path removal.

valuable information. Our linear transformation method in the phase sanitization component can effectively calibrate the chaotic phases into the orderly waveform, as shown in Figure 4.

3.3.3 Removing Static Components. Wisign'next step is to distinguish the reflections generated by static objects in the surrounding environment from those generated by human sign language movement, and focus the CSI waveform changes on the hand, finger, and arm movements. We know that static components are not constant, which is mainly caused by the tiny movements of other body parts (such as the back, waist, etc.) of the sign language users. The Local Extreme Value Detection algorithm (LEVD) mentioned in Ref. [39] is adopted for each subcarrier for removing static components. The principle is that the local extreme values of CSI real part (I) and imaginary part (Q) are detected, and the values of static components are updated in real time by calculating the mean value of local maximum and minimum values. Figure 3(a) draws the CSI values captured by an Intel 5300 network card, and Figure 3(b) draws the imaginary part updated by the LEVD algorithm.

3.3.4 Bandpass Filter. After phase sanitization, we conduce the denosing process in the frequency domain. The majority of environment noises (e.g., electronic noises) and noise caused by hardware defects (e.g., carrier frequency offset) lie in the high-frequency range. On the other hand, the body motions (e.g., chest movements and winks) fall into a low-frequency range. Hence, we adopt a five-order Butterworth band-pass filter to remove out-band noises. According to experimental observation, the frequencies of sign language motions with different subjects lie in the range of $[2Hz, 35Hz]$. To be conservative, we adopt a band-pass filter with cut-off frequency of 1Hz and 40Hz. After the bandpass filter, the CSI traces are smoothed, as shown in Figure 5(b).

3.3.5 Dimensionality Reduction Processing Based on PCA. In the wireless channel, the actions of the hands, fingers, and arms affect each subcarrier. In Figure 7, we draw the sign language actions



(a) Subcarrier 1 (b) Subcarrier 10 (c) Subcarrier 20 (d) Subcarrier 30

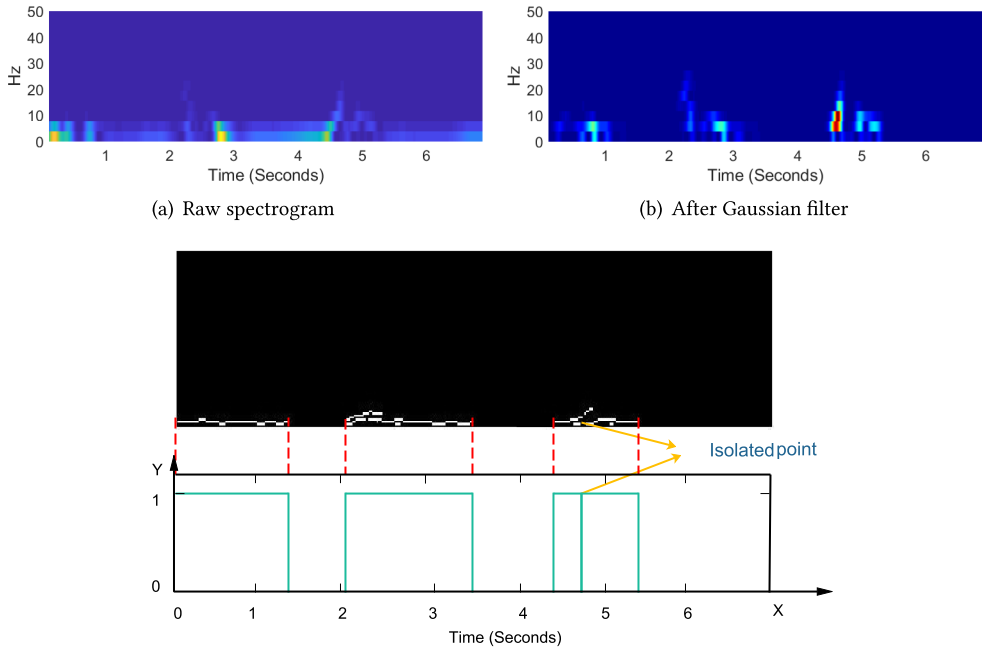
Fig. 7. Correlations and sensitivity differences among different subcarriers.

of #1, #10, #20, and #30. Based on the two facts [42, 43]—each subcarrier has different sensitivities to different parts of the human body and each subcarrier is highly correlated in waveform. In addition, because we use DBN for feature extraction, multi-dimensional data is easy to cause over-fitting of deep networks. We use PCA for dimensionality reduction instead of using all subcarrier data for feature extraction. By calculating the eigenvalues of the correlation matrix of 30 subcarriers, we find that the first eigenvalue is much larger than the other eigenvalues. So we select the first principal component as the data to be processed next. The waveform after PCA processing is as shown in Figure 6.

3.4 Sign Sentence Segmentation

After signal preprocessing, we can obtain $\hat{M} \times \hat{N} \times R$ dimension data, where $\hat{M}$ represents the characteristic information of CSI, which refers to the amplitude and phase of CSI; $\hat{N}$ represents the number of CSI streams; and R represents the sample duration. Here, $\hat{M} = 2$ and $\hat{N} = 3$. To understand the sign sentence, we have to firstly segment the sentence into sign words. Since the signal reflections of different body parts (hands, fingers, arms) in the CSI waveform after PCA denoising are mixed together, it is still difficult to extract the sign language action information from the CSI waveform after PCA denoising. Due to the different speeds of different parts of the human body, the radio signal reflection frequencies of different parts are also different. WiSign converts the CSI waveform denoised by PCA to the time-frequency domain. By using the Short-Time Fourier Transform (STFT) techniques, the waveform is converted to a spectrum that allows the CSI waveform to be analyzed in the time-frequency domain. The amplitude of the third stream is observed to be more sensitive to sign language. Here, we select the amplitude information of the third stream for processing. We first use the edge detection method to detect the start and end points of the sign language action. Due to the existence of the transition gesture, the transition gesture and the sign language action are difficult to distinguish. We find that the waveforms of the sum value of the spectrum in the same sign language action at a certain threshold are similar, so we propose to use the cross-correlation function to alleviate the influence of the transition gesture. We will introduce the details from the following three steps.

3.4.1 Spectrogram Generation. We use STFT technology to generate a PSD map. The spectrum has three dimensions: time, frequency, and FFT amplitude. The window size of the FFT determines the tradeoff between the frequency resolution and the time resolution of the STFT. As the window size increases, the STFT has higher frequency resolution and lower time resolution. We find that the frequency of sign language action is between 1 and 30 Hz, and the change time is a few milliseconds. Therefore, we chose to use the FFT size of 256 samples and the sliding window step size of 6 samples because it gives a suitable frequency resolution of 3.90 Hz and a time resolution of 6 ms to track changes in the sign language action signal. Figure 8(a) shows the raw spectrum of continuous sign language.



(a) Raw spectrogram

(b) After Gaussian filter

(c) Binary detection graph

Fig. 8. Sign language edge detection process: (1) extracting sign language patterns from raw spectrogram—the result is shown in (a); (2) using Gaussian filter to filter raw spectrogram; (3) after the use of non-maximal suppression method and double threshold detection, the obtained binary detection graph is shown in (c).

3.4.2 Detecting the Beginning and End. WiSign implements the edge detection algorithm based on image processing to detect the start and end points of sign language action. The principle is that there will be a huge difference between the CSI value of sign language action and background noise value. We first scale the obtained PSD values to the specific interval of $[0, 1]$, and its normalized formula is as follows:

$$y_{ij} = \frac{x_{ij} - min_{1\le i\le n, 1\le j\le m}\{x_{ij}\}}{max_{1\le i\le n, 1\le j\le m}\{x_{ij}\} - min_{1\le i\le n, 1\le j\le m}\{x_{ij}\},} \tag{5}$$

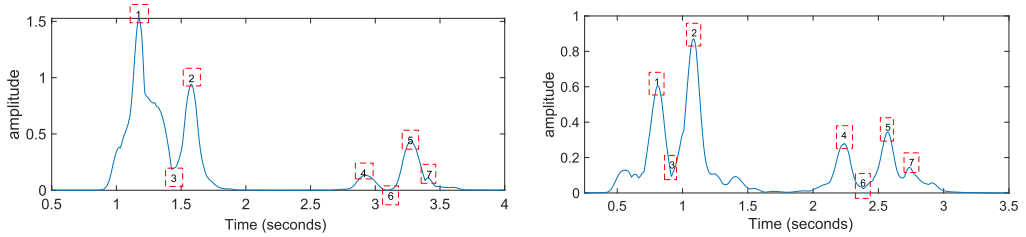
where x_{ij} is the element value of PSD matrix A, and y_{ij} is the standardized value of x_{ij}.

The reason why we chose the Canny edge detection algorithm is that this method uses two different thresholds to detect the strong edge and weak edge, respectively, and only when the weak edge is connected with the strong edge, the weak edge is included in the output image. This method is more accurate in detecting edges. The principle is mainly divided into the following four steps:

Gaussian filter. Using a Gaussian filter to smooth the image and remove noise. The generating equation of a Gaussian filter kernel with size $(2k + 1) \times (2k + 1)$ is given as follows:

$$H_{ij} = \frac{1}{2\pi\sigma^2} exp\left(-\frac{i - (k + 1)^2 + (j - (k + 1))^2}{2\sigma^2}\right), 1 \le i, j \le (2k + 1) \tag{6}$$

After selecting the σ value, the spectral value matrix is convolved with the Gaussian filter kernel matrix by moving the step size, and the filtered values of each element of the spectral value matrix are calculated. The filtered image is shown in Figure 8(b).



(a) The first high energy spectrum (b) The second high energy spectrum

Fig. 9. High-energy spectrum of two sign language movements of different people.

Gradient intensity and direction. Calculating the gradient intensity and direction of each pixel in the image. For the smoothed matrix $I(x, y)$, the two matrices of its partial derivatives in x and y directions are, respectively, as follows:

$$P_x[i,j] = (I[i,j+1] - I[i,j] + I[i+1,j+1] - I[i+1,j])/2 \tag{7}$$

$$P_y[i,j] = (I[i,j] - I[i+1,j] + I[i,j+1] - I[i+1,j+1])/2 \tag{8}$$

The magnitude of the gradient is $M[i,j] = \sqrt{P_x[i,j]^2 + P_y[i,j]^2}$, and the direction of the gradient is $\theta[i,j] = arctan\left(P_y[i,j]/P_x[i,j]\right)$.

Non-maximal suppression. Non-maximal suppression is applied to eliminate the stray response caused by edge detection. The Canny algorithm uses the size 3×3 neighborhood including the 8-direction, to perform gradient amplitude interpolation along the gradient direction for all pixels in gradient amplitude array $M[i,j]$. If the gradient intensity of the current pixel is the largest compared with the two pixels in the direction of positive and negative gradient, the pixel point is reserved as an edge point; otherwise, the pixel point will be suppressed.

Double threshold detection. Double threshold detection is applied to determine the real and potential edges, and then the edge detection is finally completed by inhibiting the isolated weak edges. We determine the value of double threshold through continuous experiments.

After edge detection, we will get a 0-1 matrix that is consistent with the dimension of the spectral image. The value 1 (white in the first subgraph of Figure 8(c)) represents the edge. We recognize at least one value greater than 0 for each frequency as a sign language action and record the current time point as 1. As shown in the second subgraph of Figure 8(c), the presence of isolated point x is found in the third sign language action. We set the threshold value W. If there is a sign language action within a range of $[x - W, x + W]$, the isolated point is considered as sign language action; otherwise, it is considered as noise and removed.

3.4.3 Transition Gesture Removal. After edge detection and segmentation, the sign language movement includes two parts—transition gesture and real sign language movement. In real life, transitional gestures are ignored by the deaf-mute because of their irregularity. However, the combination of transition gestures and real sign language gestures will have a great impact on the subsequent feature extraction of sign language according to the time sequence. We find from the segmented spectrogram that the waveforms formed by the larger spectral values of each time of the same sign language action are similar, as shown in Figure 9. This is because when a person is doing different signs, their position changes, hand shape changes, and movement changes will cause different hand speed changes. Different speed variations in different time periods constitute different signs. In general, the transition gesture is mainly formed by the change of the position of the hand, so we adopt the high energy spectrum value defined by the following paragraph, which

corresponds to the change of the speed of the palm and the arm. The corresponding Numbers (1)-(7) in Figure 9(a) and Figure 9(b) indicate the similarity between the two sign language movements. We find that the magnitude of the crest is mainly determined by the speed and standard of the sign language user, but the key transition point position for speed doesn't change.

Based on this observation, we first adopt an empirical threshold value T, and the value $x\,(i.j)$ above the threshold value T in each time is retained. We sum over the values of spectral density at each time point to get the waveform $Q\,(t)$, which is called the high-energy spectrum value waveform. Since the transition gestures are all located before the real sign language actions, the high-energy spectrum waveforms of $Q\,(t)$ and the sign language word templates are firstly made by cross-correlating to find the time delay. The cross-correlation function between $Q\,(t)$ and the i^{th} word template $W_i\,(t)$ is defined as follows:

$$R\,(\tau) = \int_{\infty}^{\infty} Q\,(t)\,W_i\,(t+\tau)\,dt, 0 \le \tau \le E - 0.5, \tag{9}$$

where E is the time length of waveform $Q\,(t)$, and we assume that each sign language action lasts at least 0.5s. We calculate the τ value that can make $R\,(\tau)$ greater than a certain threshold, and the first τ seconds are removed as the waveform part of the transition gesture. Finally, we will get multiple waveforms removed by the transition gesture and send them to the sign language recognition module for identification.

3.5 Sign Language Recognition

First, we use a high-energy spectral value waveform after removing the transition gesture for macro classification. Then, we use the microscopic classification of the PCA waveform after the transition gesture is removed. From a multi-scale perspective, the DTW matching algorithm is suitable for macroscopic observation, while the HMM is more suitable for microscopic detail analysis. So, it is practical to combine these two methods. In this module, the time spent by different sign language actors may be different when the same sign language is used. The time required for the same sign language actor to do the same sign language action may also be different, which leads the number of sampled data of the same sign language action frames to tend to vary. We use the DTW method to align and classify the high energy spectrum waveforms of test samples and templates.

For a more sophisticated classification of the sign language, we use the PCA waveform after removing the transition gesture to extract the feature. With the successful application of deep learning, it has achieved good success in the field of motion recognition. Compared with the traditional artificial extraction feature method that relies on artificially defined preset features, the neural network automatically extracts features directly from the original signal, which greatly covers all effective features, and does not lose information too much. WiSign leverages both DBN and HMM to build the recognition model of a multiple sign words composition because we find that the spectrum of the different sign words that a user makes is different. DBN is used to extract effective features for the recognition. The HMM model is used for recognizing the sign words because it is suitable to establish the state transition model by using the time-varying characteristics. Since two different ASL words may have similar motions, the HMM models can also be similar. To correct the similar models, we leverage the N-gram language model to correct the sign words for an improved accuracy.

3.5.1 DTW. DTW is a typical optimization problem. It describes the time correspondence between the test template $X = (x_1, x_2, \ldots, x_n)\,, n \in N^+$, and the reference template $Y = (y_1, y_2, \ldots, y_m), m \in N^+$ with the Time Warping Function satisfying certain conditions. To solve the regularization function corresponding to the minimum cumulative distance when two templates are matched. The DTW distance is the Euclidean distance of the optimal distortion path

between the two waveforms and the local path constraints calculated under the boundary conditions. For the scene where the user has different sign language speeds at a specific position, we will match the test sample and each sign language word template to calculate DTW distance and then filter out the larger DTW template. Then, we scale the test sample with the remaining sign language templates. If one sample point of the test sample corresponds to multiple sample points of the template, e.g., x_i corresponds to $y_j, y_{j+1}, \ldots, y_{j+t}$, we perform cubic spline interpolation where the number of interpolation is t between $x_i - 1$ and $x_i + 1$. If multiple sampling points of a test sample correspond to one sampling point of a template, e.g., $x_j, x_{j+1}, \ldots, x_{j+t}$ corresponds to y_j, we average the time series $x_j, x_{j+1}, \ldots, x_{j+t}$. Thus, the time series of the test sample and the sign language template are aligned. This provides help for the next microscopic processing based on timing HMM.

Then, WiSign uses a PCA filtered waveform for micro-classification. The reason is because the waveform contains rich information. When the two sign language actions are very similar, more valuable information can be extracted through the original signal. We scale the waveforms filtered by PCA according to the above method analogy. Since one sampling point of the spectrogram corresponds to six sampling points of the original waveform processed by the PCA, the sampling point of six times should be interpolated at the position corresponding to the original waveform timing or the average of six positions should be respectively obtained, and, finally, the test is performed. The sample has the same sample data as the template.

3.5.2 DBN-based Feature Extraction. After DTW processing, we add windows to the obtained waveform. Each frame has $\hat{M} \times \hat{N} \times N$ dimensions, where N is the length of each window. Here, we take N equal to 180. Then, we merge the data from each frame into one dimension and send it to DBN. DBN is selected as the feature extraction of microscopic classification. The reason why DBN is selected is because we hope the receiver may be a resource-constrained device such as a mobile phone, so it is necessary to design an algorithm with low computational cost; secondly, we may not get a large number of training data tags and need unsupervised learning methods.

DBN is a deep network structure composed of multi-layer restricted Boltzmann Machines (RBM). The core idea of DBN is to extract and abstract the input data from each RBM from bottom to top, and to keep important information as far as possible. Hence, it is used in WiSign for feature extraction.

In the process of the training model, DBN is mainly divided into two steps: pre-training and fine-tuning. The pre-training process trains each layer of RBM network separately, so as to ensure that feature vectors are mapped to different feature spaces and to retain feature information as much as possible.

For the layer i RBM model, which is a network consisting of a layer of visual neurons and a layer of hidden neurons, free energy [37] could be defined as

$$Energy\,(v, b) = b^T v + c^T h + h^T W v, \tag{10}$$

where v and h are the visual variables and hidden variables, b and c are the offset vectors of the visible layer and hidden layer, respectively, and W is the matrix of weights between the visual layer and the hidden layer. When training RBM, our purpose is to find the proper weight W and offset c, b to maximize the $p\,(v, h)$. $p\,(v, h)$ is given by

$$p\,(v, h) = \frac{1}{Z} e^{-Energy} = \frac{e^{-Energy}}{\sum\limits_{h,v} e^{-Energy(v,h)}}, \tag{11}$$

where Z is the normalization factor. Since the calculation of Z costs too much, Equation (11) cannot be directly applied. Therefore, it is challenging to obtain the optimal weights by using the

ALGORITHM 1: k-step contrastive divergence

Input: RBM $V_1, \ldots, V_m, H_1, \ldots, H_n$, training batch S
Output: gradient approximation Δw_{ij}, Δb_j and Δc_i for $i = 1, \ldots, n, j = 1, \ldots, m$

```
init Δw_ij = Δb_j = Δc_i = 0 for i = 1..., n, j = 1, ..., m
for all the v ∈ S do
    v^(0) ← v
    for t = 0 ... k − 1 do
        for i = 1, ..., n do
            sample h_i^(t) ~ p(h_i | v^(t))
        end for
    end for
    for i = 1, ..., n, j = 1, ..., m do
        Δw_ij ← Δw_ij + p(H_i = 1 | v^(0)) · v_j^(0) − p(H_i = 1 | v^(k)) · v_j^(k)
        Δb_j ← Δb_j + v_j^(0) − v_j^(k)
        Δc_i ← Δc_i + p(H_i = 1 | v^(0)) − p(H_i = 1 | v^(k))
    end for
end for
```

maximum likelihood method. We adopt a greedy layer by a layer-learning algorithm called CD-k. The main steps of RBM's rapid learning based on k-step contrastive divergence (CD-k) is shown in Algorithm 1.

After we obtain the near optimal weights for the deep network, we need to set a back propagation network in the last layer. The back propagation algorithm is used to calculate the error term of each unit in the neural network. After all the error terms have been calculated, we can update the weight again. This step is called fine-tuning.

3.5.3 HMM. A vector composed of classification probabilities is generated by the DBN for each time window. Then, we apply HMMs on the probability vector sequences to identify each sign word. The value of the observed variable corresponds to the probability vector, and its output distribution is represented by Gaussian mixture density.

The hidden variable of HMM corresponds to the internal state of the motion at each moment. In the model, the hidden variable at time t depends only on the previous hidden variable at time $t-1$. In addition, variables observed in time t only depend on hidden variables of time t. Hence, time rules for events can be captured using HMMs. We used the Baum-Welch algorithm [31] to estimate the HMM parameter. Because the HMM from left to right is prepared for each state, the Viterbi algorithm [31] finds state transitions throughout HMM. This means that we consider the state transition of a sign word from one arbitrary HMM to another's first state. When testing the data for a sign word, it matches the HMM of each sign word to get a corresponding probability.

3.5.4 N-gram Model. N-gram is based on the assumption that the N^{th} word occurs in relation to the last $n-1$ words and is not in relation to any other word (this is also the assumption in HMM). The probability of the whole sentence is equal to the product of probabilities of all the words. The probability of each word can be calculated by statistics in the corpus. Assuming that sentence T is composed of word sequences $w_1, w_2, w_3 \ldots, w_n$, then the N-gram language model can be expressed as follows.

$$P(T) = P(w_1) * P(w_2) * P(w_3) \cdots * P(w_n) = P(w_1) * P(w_2 | w_1) * P(w_3 | w_1 w_2) * \cdots P(w_n | w_1 w_2 \ldots w_{n-1}) \quad (12)$$

Table 1. ASL Test Lexicon

part of speech	*vocabulary*
pronoun	I, you, he, we, you (pl), they
verb	want, like, lose, don't want, don't like, love, pack, hit, loan
noun	box, car, book, table, paper, pants, bicycle, bottle, can, wristwatch, umbrella, coat, pencil, shoes, food, magazine, fish, mouse, pill, bowl
adjective	red, brown, black, gray, yellow

In WiSign, we set $N = 1$. That is, a word in a sentence is only related to the previous word. Then, the above formula can be simplified as follows.

$$P(T) = P(w_1 | \text{begin}) * P(w_2 | w_1) * P(w_3 | w_2) * \cdots * P(w_n | w_{n-1}) \tag{13}$$

Note that the calculation of the above probability, $P(w_1 | \text{begin})$, is defined as the ratio of the total number beginning with w_1 in all sentences to the total number of sentences; $P(w_2 | w_1)$ is defined as the ratio of the number of simultaneous occurrences of w_1, w_2 to the number of simultaneous occurrences of w_1 in all sentences. But note that even though we have a large training set, there will still be a lot of linguistic phenomena that do not occur in the training set, resulting in that a lot of parameters (the probability of an n-pair) are 0. Hence, we smooth the data, adding a constant δ ($0 < \delta \leq 1$) to the number of occurrences of each n pair. The formula is as follows:

$$P(W_i | W_{i-n+1}, \ldots, W_{i-1}) = \frac{(count(W_{i-n+1}, \ldots, W_{i-1}, W_i) + \delta)}{(count(W_{i-n+1}, \ldots, W_{i-1}) + N\delta)} \tag{14}$$

We filter out the HMM models with a lower probability, and then we select the recognition result with the highest probability from the remaining model as the final classification result.

4 EVALUATION

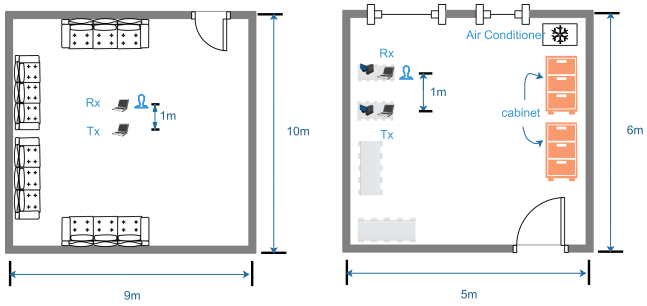
In this section, we present the evaluation of WiSgin. We first present the evaluation methodology and then show the experimental results.

4.1 Evaluation Methodology

4.1.1 Implementation. We implement a prototype of WiSign with commercial WiFi devices. In our implementation, we use two laptops with Intel 5300 network cards and Ubuntu system. In particular, one of the laptops is equipped with an antenna that runs at a frequency of 5.825GHz in IEEE 802.11n monitor mode as the transmitter, while the other laptop equipped with three antennas works as the receiver. The firmware is modified to report the original CSI to the upper layers. During the measurement, the transmitter sends about 1,000 packets every second to the receiver via the WiFi router. We set a high sending packet frequency help to achieve a higher CSI sampling rate, which ensures the time resolution of the CSI value to capture subtle changes in the CSI stream and maximize the details of different sign word motions.

4.1.2 Experimental Methodology. Our evaluation is conducted in two scenarios, a $9m \times 10m$ rest hall in Figure 10(a) and a $5m \times 6m$ office in Figure 10(b). The rest hall is a relatively empty environment and the office is a more complicated environment. Receiver and transmitter are placed at $1.5m$ height. And the distance between them is 1m.

To understand the user diversity, we recruit 30 volunteers, including 15 males and 15 females. WiSign recognizes sentences containing personal pronouns, verbs, nouns, and adjectives. This structure allows a large number of meaningful sentences to be generated using words randomly selected from each class, as shown in Table 1. There are 40 words, including 6 personal pronouns,



(a) Rest hall (b) Office

Fig. 10. Experimental environments.

Fig. 11. Experiment setting.

9 verbs, 20 nouns, and 5 adjectives. A deep network with a five-layer (one visible layer that has 180 units and four hidden layers that have 400 units for each layer) structure is utilized in our experiment. In addition, we set the DBN learning rate to 0.5 and the epoch size to 40.

As shown in Figure 11, volunteers are asked not to operate the ASL on the LOS path between the receiver and the transmitter. Because NLOS situations are more common in reality and more difficult to identify, as the signal strength reflected by humans is lower than the direct path signal. We randomly composed 100 sets of ASL sentences by grammar rules, and each set of sentences is executed 100 times per person. Then, we get $100 \times 100 \times 30$ sets of ASL statements. After each person learns the sign language action carefully, the duration of their sign language action is between 0.5s and 2.5s. We fix the sampling time to 15 seconds per sample. The collected CSI is then stored in the receiver and processed with matlab, with one tenth of the sign language instances used for testing and the rest for the training of each person. Note that WiSign does not require training samples from the same user who must identify their sign language, and we are free to move the furniture in the room.

4.2 Overall Performance

Predictive statements can have errors, including word replacement, insert, and delete errors. The accuracy of sign word recognition is calculated by the following equation [26]:

$$Acc = \frac{N - D - S - I}{N}, \tag{15}$$

where N is the total number of words in the test set, D is the number of deletions, S is the number of substitutions, and I is the number of insertions. D is the number of sign language actions that are not detected, possibly because the subjects are too far away from the receiving end, or the sign language actions are too slow. S is the number of mistakes that occur in recognizing one word as another. I is the number of sign language words that may be discriminated due to noise. S mainly shows the rationality of WiSign technology, while D and I mainly shows the robustness of WiSign.

We first present the accuracy of WiSign for 30 volunteers. If we use the personalized model that is trained with the training set of a specific user, the accuracy on the same user's test set can be as high as 92% in the rest hall and 87% in the office. If we mix up the data collected from multiple users, namely training a general model, the recognition accuracy drops. We vary the group size of volunteers and measure the average accuracy. The experimental results are shown in Figure 12. We can find that the average recognition accuracy is 92%, 87%, 80%, 78%, 76%, and 75% in the rest hall and 87%, 81%, 75%, 73%, 71%, and 70% in the office for 1, 5, 10, 15, 20, and 30 volunteers, respectively.

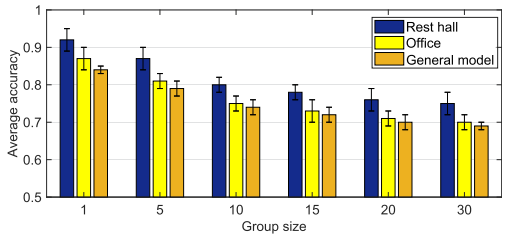


Fig. 12. Impact of group sizes on accuracy.

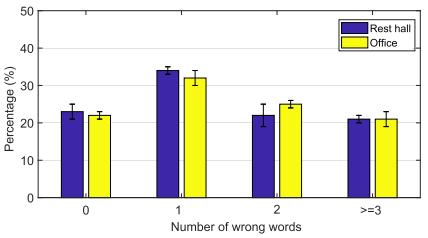


Fig. 13. Percentage of the number of wrong words.

As expected, the accuracy drops with the increase of the number of volunteers. This is because, in this experiment, the training set and the testing set have the same subjects. The subjects have different action habits even for the same ASL action. Hence, the more subjects we use to train, the more ambiguity we bring in the model. For example, for a action made by only one subject A, the over-fitting is caused when the training set made by subject A is also the test set of subject A. With the increase of group size, the more people participating in the training, the more noise each person introduces, and the accuracy decreases accordingly. Experimental results reveal that the personalized model is better than the general model.

However, when the group size increases from 10 volunteers to 30 volunteers, the slow degradation of the accuracy decline is acceptable. The experimental results demonstrate the feasibility of WiSign in different scenarios. In brief, the accuracy rate reaches 92% and 87% with the personalized model and 69% with the general model for 30 users in different experimental environments.

To further investigate WiSign's effect on sentence level ASL, we calculate the percentage of the number of word errors in each sentence in 100 groups of ASL sentences. We test it with a hybrid model of 10 people. From Figure 13, we can see that about 80% of each ASL sentence is less than or equal to two wrong words, and about 20% is that there is no wrong word in the whole sentence.

4.2.1 Impact of Different Distance. WiSign uses omnidirectional antenna and works in 5GHz frequency band. We put the transmitter in a fixed position and adjusted the position of the receiver. The user stands in front of RX and performs sign language motions. Figure 17(a)–(c) shows the waveform of the "car" sign word at different distances between the transceivers. We find that as the distance between the transceivers increased, the motion of the sign language detected becomes weaker and weaker. When the distance is more than 4 meters, the sign language generated in the CSI stream suddenly almost disappeared. We tested the classification results at different distances between the TX antenna and RX antenna. Figure 14 shows that the detection accuracy of sign language will decrease with the increase of receiver distance, because the weaker signal is difficult to respond to the hand movement, resulting in the decrease of sensitivity. However, WiSign can still achieve more than 80% accuracy for distances up to 3m.

4.2.2 Impact of Different Sampling Rates. The sampling rate is extremely important for fine-grained sign word recognition because the changes caused by the sign word can be subtle and rapid-changing. Hence, the precision of motion detection and extraction depends on the high temporal resolution of the CSI value. A higher sampling rate can provide more CSI values for each sign word, which contains more CSI waveform information and increases the motion extraction and classification accuracy of the sign word.

We evaluate the performance of WiSign with different sampling rates, varying from 500 to 2,000 packets/s. The average recognition accuracy is shown in Figure 15. The experiments use the mixed data of all 10 volunteers. We can find that with the increase of the sampling rate, the average

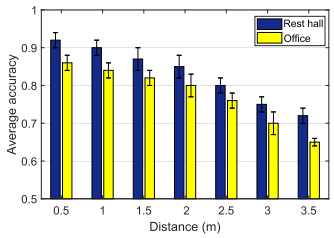


Fig. 14. Impact of distance on accuracy.

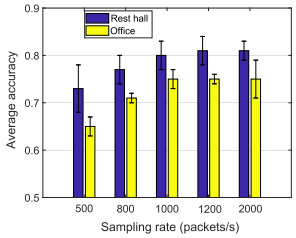


Fig. 15. Impact of sampling rates on accuracy.

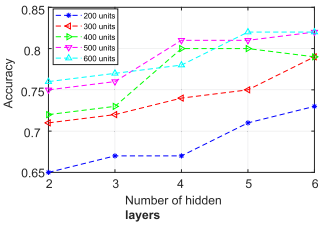
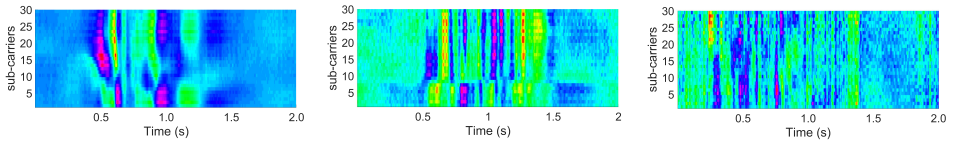
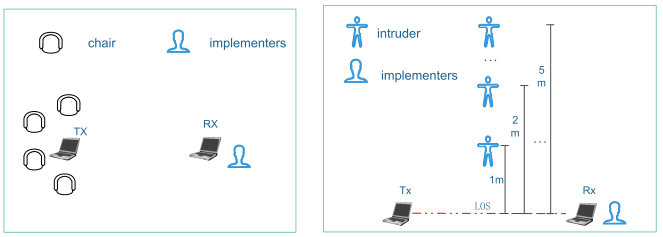


Fig. 16. The recognition accuracy with the number of hidden layers and hidden units.



(a) 1m (b) 2.5m (c) 4m

Fig. 17. Impact of different distances.



(a) Some chairs are placed next to the transmitter (b) Some persons are moving around

Fig. 18. Environment changing in the rest hall.

accuracy increases. When the sampling rate decrease from 2000HZ to 1000HZ, the average recognition accuracy is almost unchanged. When the sampling rate decreased from 1000HZ to 500HZ, the average recognition accuracy decreases significantly. Based on the above results, the sampling rate should not be less than 1000HZ.

4.2.3 Impact of the Deep Learning Parameters. We evaluate the performance of WiSign with different numbers of hidden layers and hidden units. We vary the number of the hidden units in each layer from 200 to 600. We also vary the number of hidden layers from 2 to 6. As shown in Figure 16, the experimental results in the rest hall reveal that a large number of hidden layers generally provide better accuracy. And increasing the number of hidden units in each layer also brings benefits. However, using more than 400 units in a layer will bring only limited improvement. Even though a complicated deep learning network can bring performance improvement, it is not free. A large number of hidden layers and hidden units will lead to a high computation cost, delaying the recognition response time. Hence, to balance the precision and the response time, we use 4 hidden layers and 400 hidden units as our deep learning parameters.

4.2.4 Impact of Epoch Size. Figure 19 shows the impact of different epoch sizes in the DBN. We can see that as the number of iterations increases, the detection accuracy increases. When the

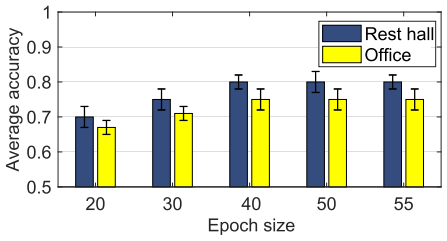


Fig. 19. Accuracy vs. different epoch.

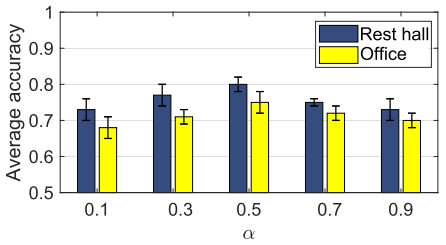


Fig. 20. Accuracy vs. different learning rate.

Table 2. Accuracy and Variance in Different Experimental Environments

Experimental environment	Accuracy	Standard Deviation
Initial scenario and environment	80%	2%
Environment (I)	79%	4%
Environment (II) (3m distance and 3 interfering humans)	76%	3%
Environment (III)	70%	4%

epoch reaches 40, the accuracy tends to converge. If we use a larger epoch, the benefit is limited and the training cost is too large. Besides, a too large epoch size may lead to over-fitting. So, in our current implemenation, we choose $epoch = 40$.

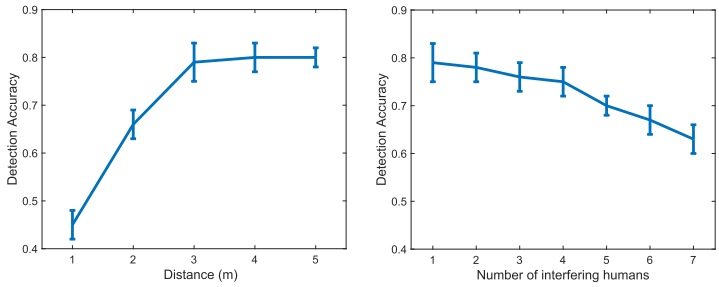
4.2.5 Impact of Learning Rate. The Figure 20 shows the impact of different learning rates in DBN, and we can see that when α rises from 0.1 to 0.5, the average accuracy increases. This is because the accuracy of the model tends to converge when the epoch is certain and the learning rate increases within a certain range. In addition, when it rises from 0.5 to 0.9, the accuracy becomes smaller and smaller. This is because when it is too large, the model cannot reach the optimal convergence point due to the BP algorithm jumping back and forth in the valley repeatedly.

4.2.6 Impact of Different Environments and Scenarios. To test the robustness of WiSign, we make several changes of the rest hall, including (I) some chairs are placed next to the transmitter (as shown in Figure 18(a)). (II) Some persons are moving around (as shown in Figure 18(b)). And we add a new scenario, (III) moving to the office in Figure 11.

In the above three different environments or scenarios, the training model is provided by the initial scenarios and environment, and the test data is provided by the new environment or scenario. We adopt a 10-person hybrid model and test 10 people in a new environment or scenario. We summarize the experimental results in Table 2 and Figure 21. From the results, we can find that subtle environment changes such as environment (I) and even the environment changes with other human activities such as environment (II) can only cause limited performance variations. But if we change another totally new environment such as environment (III), the accuracy drops significantly. The results demonstrate the WiSign can achieve satisfied accuracy when operating in the environment with small changes.

In the environment (II), we ask a person to walk around freely within 1m, 2m, 3m, 4m, and 5m ranges. The results are shown in Figure 21(a). We can find that when the person is near the receiver, the accuracy of WiSign can be seriously influenced. But with a distance larger than 3m, the influence on accuracy is limited.

We increased the number of interference to 2–7 people at a distance of 3m. As shown in Figure 21(b), WiSign can resist interference from a certain number of people when 1–4 people walk around freely. But when there are too many people, it doesn't work very well. So, in practical



(a) A person at different LOS locations with different distances

(b) People are moving around

Fig. 21. Impact of interfering humans on WiSign accuracy.

application, when users find inaccuracy, they take the initiative to change the position with the system to achieve better service.

4.2.7 Impact of Different Methods. We prepare the following methods to study the effectiveness of partial path removal, CSI phase, DBN, HMM and N-gram. In addition, we also use the current advanced methods of sign language recognition for comparison. Here, we use the 10-person hybrid model as the evaluation standard.

(1) W/o partial path removal (PPR). This method does not use the algorithm of PPR to process the data, but only uses the data that has been preprocessed by other methods after the original CSI data. (2) W/o phase. This method does not use the obtained CSI phase data. It only uses the acquired CSI amplitude time series data. (3) W/o DBN. This method does not use DBN to automatically extract features as input of HMM. Instead, 10 manually selected features extracted from time domain and frequency domain are used. The features of time domain include the normalized standard deviation, median absolute deviation, interquartile range, period of motion, signal entropy, velocity of signal changes, and offset of signal strength. The features of frequency domain include spectral energy, the spectral centroid, and spectral entropy. (4) W/o HMM+N-gram (H-NG). This method does not use the classification model combined with HMM and N-gram, but directly uses the method composed of DBN and softmax classifier [18]. (5) W/o N-gram (NG). Instead of using the language model of N-gram, we directly select the model of maximum probability HMM as the output result. (6) WF. This method adopts the method W/o DBN feature extraction, and coordinates with the traditional sign words recognition method of DTW with KNN [16] for sign language recognition. (7) SF. This method uses data processed by PCA, and then directly adopts a 9-layer convolutional neural network (CNN) [18] as a classification algorithm. (8) WiSign. This is our proposed method.

The experimental results are shown in Figure 22. By comparing methods (6), (7), and (8), we find that the approach adopted in this article achieved the best performance, with a significant improvement over the traditional approach.

Contribution of PPR. As can be seen from Figure 22, the precision of W/o PPR decreases slightly, which, combined with the phenomenon shown in Figure 6, is conceivable.

Contribution of phase. As can be seen from Figure 22, the precision of W/o phase in the rest hall space decreased slightly, while the precision in the office room decreased significantly. We suspect that in an environment with multiple obstacles, the phase information of CSI has stronger robustness because of its characteristics and will not be blocked by obvious obstacles and cause information loss.

Contribution of DBN, H-NG, and NG. Figure 22 shows W/o DBN and W/o H-NG accuracy is reduced by about 12%–15%, and W/o NG falls by about 4%. We can say that DBN and H-NG

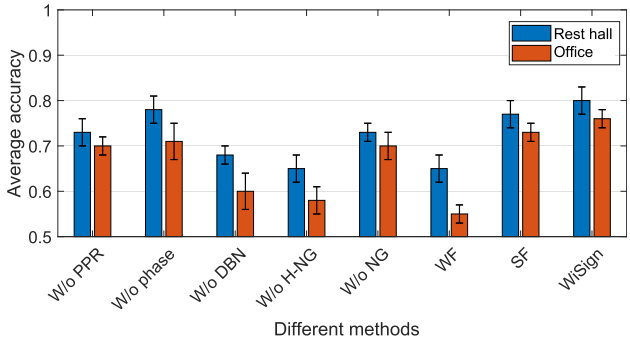


Fig. 22. The recognition accuracy of different methods.

Table 3. Subject-Independent Experiment

Case	Experiment group	Training Data Collection Phase			Testing Phase		
		Subject	Sample size	Times	Subject	Sample size	Times
(1)	$M\{\overline{a}\}->\hat{a}$	a	100	100	a	50	100
(2)	$M\{\overline{a}\}->\hat{b}$				b		
(3)	$M\{\overline{a},\overline{b}\}->\hat{b}$	a,b					
(4)	$M\{\overline{a},\overline{b}\}->\hat{c}$				c		
(5)	$M\{\overline{a},\overline{b},\overline{c}\}->\hat{c}$	a, b, c					
(6)	$M\{\overline{a},\overline{b},\overline{c}\}->\hat{d}$				d		
(7)	$M\{\overline{a},\overline{b},\overline{c}\}->\hat{e}$				e		
(8)	$M\{\overline{a},\overline{b},\overline{c}\}->\hat{f}$				f		

Contribution is very high. Although we also extract the features of numerous artificial hand picking, the precision of this decline is obvious. All of these results show the effectiveness of automatic feature extraction. Moreover, the combination of the HMM prediction model and N-gram grammar model gives us a new classification method.

4.2.8 Subject-independent Performance. Since different people conduct sign languages in slightly different ways, in order to investigate the performance when the set of training subjects and the set of testing subjects are different, a set of experiments in the rest hall and in the office is designed by changing the number of subjects belonging to the training set.

The experiment is divided into two stages: training data collection stage and test stage. As shown in the Table 3, we use eight experiment groups with different training subjects and testing subjects. $M\{x\}->y$ means using the model trained by subject set $\{x\}$ to test on the subject y. Each training subject performs 100 times for each ASL sentence. For example, for each ASL sentence, subject a executed 100 times to establish a set of the subject training set $M\{\overline{a}\}$. Then, for each ASL sentence, subject a did another 50 times to set up a test dataset and test the performance using model $M\{\overline{a}\}$ (see line 1 $M\{\overline{a}\}->\hat{a}$). In the second line, when the second subject b participated in the experiment, we asked subject b to do the same 50 times for each sign language sentence, we buily the test set $\hat{b}$, and tested the performance using model $M\{\overline{a}\}$ (see line 2 $M\{\overline{a}\}->\hat{b}$). The rest of the table is analogous.

From Figure 23, we can see the Case (1) has the highest accuracy. Obviously, this is because Case (1) is a personalized model. But the accuracy of Case (2) is greatly reduced, and Cases (6)–

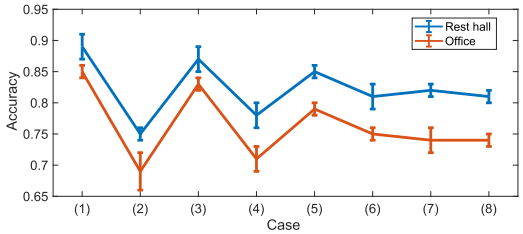


Fig. 23. Performance of the subject-independent experiment.

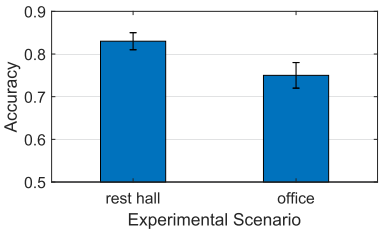


Fig. 24. Subject-independent performance when testing on 30 volunteers.

(8) use a group of three subjects a, b, c. When testing on d, e, f, accuracy tends to converge and the accuracy is slightly higher than Case (2). Generally speaking, when there are more subjects in the training set, the model will be likely to learn more different action habits from the training subjects who have similar action habits with the testing subjects. Hence, we can expect the Wisign model to work better when training more people. Comparing the results of Cases (2), (4), and (6) in Figure 23, we can find that when we test the model on a subject who is not in the training set, using more subjects in the training set can obtain a better performance.

In order to investigate the impact of more subjects in the training set on the results, 10-fold cross validation is used to obtain the results. Specifically, 30 subjects are divided into 10 groups. Each time, one group is selected as the testing set and the other nine sets are selected as the training/verification sets. Ten times of training and testing are performed, and the final experimental results are obtained by calculating the mean value of 10 tests. The results in Figure 24 show that the accuracy is around 83% and 75% in the rest hall and office, respectively. The obtained accuracy is similar to Cases (6), (7), and (8) in Figure 23. The results indicate that the training data of the three subjects may be good enough to obtain a relatively satisfactory general model when there is no personalized model.

5 CONCLUSION

In this article, we present WiSign, a CSI-based ASL recognition system that is able to recognize the continuous ASL sentences with commercial WiFi infrastructure. Different from existing works that use manually extracted features to recognize isolated ASL words, WiSign can automatically segment the raw CSI sequences into segments that correspond to sign words and then recognize the whole sentence. WiSign exploits several noise removal methods to eliminate different kinds of noises in the raw CSI. WiSign leverages a five-layer DBN to extract the features and combine the speech recognition acoustic model and HMM model to build the classification model. The recognized sign words are then corrected by N-gram, a language model, to further improve the accuracy. Then, WiSign puts together the sequential sign words to reconstruct the ASL sentence. We implement a prototype of WiSign and evaluate it in real environments. The experimental results show that WiSign can achieve the accuracy of 92% and 87% if the personalized model is adopted for each user, and the average accuracy of 69% if using a general model for thirty users.

REFERENCES

[1] Heba Abdelnasser, Moustafa Youssef, and Khaled A. Harras. 2015. WiGest: A ubiquitous WiFi-based gesture recognition system. In *IEEE Conference on Computer Communications*.

[2] Fadel Adib, Zachary Kabelac, Dina Katabi, and Robert C. Miller. 2014. 3D tracking via body radio reflections. In *11th USENIX Symposium on Networked Systems Design and Implementation*.

[3] Fadel Adib and Dina Katabi. 2013. See through walls with WiFi! *Computer Communication Review* 43, 4 (2013), 75–86.

[4] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank Deruyter. 2011. Using mobile phones to write in air. In *International Conference on Mobile Systems, Applications, and Services*.
[5] Kamran Ali, Alex X. Liu, Wei Wang, and Muhammad Shahzad. 2015. Keystroke recognition using WiFi signals. In *International Conference on Mobile Computing and NETWORKING*.
[6] Marcell Assan and Kirsti Grobel. 1997. *Video-based Sign Language Recognition Using Hidden Markov Models*. 97–109.
[7] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. 2015. Tracking keystrokes using wireless signals. In *International Conference on Mobile Systems, Applications, and Services*.
[8] Hanyi Dai and Gong Zhang. 2014. The research and tests of Wi-Fi interference based on IEEE 802.11n CSI-tool. In *Informatization Research*.
[9] Emre Ertin, Nathan Stohs, Santosh Kumar, Andrew Raij, Mustafa Al'Absi, and Siddharth Shah. 2011. AutoSense: Unobtrusively wearable sensor suite for inferring the onset, causality, and consequences of stress in the field. In *ACM Conference on Embedded Networked Sensor Systems*.
[10] Gaolin Fang, Wen Gao, and Debin Zhao. 2007. Large-vocabulary continuous sign language recognition based on transition-movement models. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans* 37, 1 (2007), 1–9.
[11] Wen Gao, Gaolin Fang, Debin Zhao, and Yiqiang Chen. 2004. A Chinese sign language recognition system based on SOFM/SRN/HMM. *Pattern Recognition* 37, 12 (2004), 2389–2402.
[12] Kazuyuki Imagawa, Hideaki Matsuo, Rinichiro Taniguchi, Daisaku Arita, Shan Lu, and Seiji Igi. 2000. Recognition of local features for camera-based sign language recognition system. In *International Conference on Pattern Recognition*.
[13] Bryce Kellogg, Vamsi Talla, and Shyamnath Gollakota. 2014. Bringing gesture recognition to all devices. In *Usenix Conference on Networked Systems Design and Implementation*.
[14] Hamed Ketabdar and Mehran Roshandel. 2010. Towards using embedded magnetic field sensor for around mobile device 3D interaction. In *International Conference on Human Computer Interaction with Mobile Devices and Services*.
[15] Simon Lang, Marco Block, and Raúl Rojas. 2012. Sign language recognition using Kinect. In *International Conference on Artificial Intelligence and Soft Computing*.
[16] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. 2016. WiFinger: Talk to your smart devices with finger-grained gesture. In *ACM International Joint Conference on Pervasive and Ubiquitous Computing*.
[17] Mohandes M., Aliyu S., and Deriche M. 2014. Arabic sign language recognition using the leap motion controller. In *IEEE International Symposium on Industrial Electronics*.
[18] Yongsen Ma, Gang Zhou, shuangquan Wang, Hongyang Zhao, and Woosub Jung. 2018. SignFi: Sign language recognition using WiFi. In *ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*.
[19] Pedro Melgarejo, Xinyu Zhang, Parameswaran Ramanathan, and David Chu. 2014. Leveraging directional antenna capabilities for fine-grained gesture recognition. In *2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*.
[20] Rajalakshmi Nandakumar, Bryce Kellogg, and Shyamnath Gollakota. 2014. Wi-Fi gesture recognition on existing devices. *Eprint Arxiv* 3, 2 (2014), 17–17.
[21] Taiwoo Park, Jinwon Lee, Inseok Hwang, Chungkuk Yoo, Lama Nachman, and Junehwa Song. 2011. E-Gesture: A collaborative architecture for energy-efficient gesture recognition with hand-worn sensor and mobile devices. In *International Conference on Mobile Systems, Applications, and Services*.
[22] Lionel Pigou, Sander Dieleman, Pieter Jan Kindermans, and Benjamin Schrauwen. 2014. *Sign Language Recognition Using Convolutional Neural Networks*.
[23] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. 2013. Whole-home gesture recognition using wireless signals. In *19th Annual International Conference on Mobile Computing & Networking*.
[24] Jiacheng Shang and Jie Wu. 2017. A robust sign language recognition system with multiple Wi-Fi devices. In *The Workshop on Mobility in the Evolving Internet Architecture*.
[25] Yang Si, Song Ren, Qinkun Xiao, Yuan Ma, Kun Zhong, and Yang Xuemeng. 2018. Sign language recognition algorithm based on color and depth image. *Science Technology and Engineering* (2018).
[26] Thad Starner, Joshua Weaver, and Alex Pentland. 1998. Real-time American sign language recognition using desk and wearable computer based video. *IEEE Transactions on Pattern Analysis & Machine Intelligence* 20, 12 (1998), 1371–1375.
[27] Gordon L. Stuber, John R. Barry, Steve W. Mclaughlin, Ye (Geoffrey) Li, Mary Ann Ingram, and Thomas G. Pratt. 2004. Broadband MIMO-OFDM wireless communications. In *IEEE* 92, 2 (2004), 271–294.
[28] Chao Sun, Tianzhu Zhang, Bingkun Bao, Changsheng Xu, and Tao Mei. 2013. Discriminative exemplar coding for sign language recognition with Kinect. *IEEE Transactions on Cybernetics* 43, 5 (2013), 1418–1428.
[29] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu Han Kim. 2015. 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*.

[30] Aditya Virmani and Muhammad Shahzad. 2017. Position and orientation agnostic gesture recognition using WiFi. In *International Conference on Mobile Systems, Applications, and Services*.
[31] Christian Vogler and Dimitris Metaxas. 1998. ASL recognition based on a coupling between HMMs and 3D motion analysis. In *International Conference on Computer Vision*.
[32] Jue Wang, Deepak Vasisht, and Dina Katabi. 2014. RF-IDraw: Virtual touch screen in the air using RF signals. In *ACM Conference on SIGCOMM*.
[33] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. 2015. Understanding and modeling of WiFi signal based human activity recognition. In *International Conference on Mobile Computing and NETWORKING*.
[34] Xuyu Wang, Lingjun Gao, and Shiwen Mao. 2016. PhaseFi: Phase fingerprinting for indoor localization with a deep learning approach. In *Global Communications Conference*.
[35] Kaishun Wu, Xiao Jiang, Youwen Yi, Gao Min, and Lionel M. Ni. 2012. FILA: Fine-grained indoor localization. In *IEEE INFOCOM*.
[36] Jie Yang, Yong Ge, Hui Xiong, Yingying Chen, and Hongbo Liu. 2010. Performing joint learning for passive intrusion detection in pervasive wireless environments. In *IEEE INFOCOM*.
[37] Chengwei Yao and Gencai Chen. 2016. Hyperparameters adaptation for restricted Boltzmann machines based on free energy. In *International Conference on Intelligent Human-Machine Systems and Cybernetics*.
[38] Koji Yatani and Khai N. Truong. 2012. BodyScope: A wearable acoustic sensor for activity recognition. In *ACM Conference on Ubiquitous Computing*.
[39] Nan Yu, Wei Wang, Alex X. Liu, and Lingtao Kong. 2018. QGesture: Quantifying gesture distance and direction with WiFi signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 2, 1 (2018), 23–73.
[40] Jie Zhang, Xiaolong Zheng, yong Tang, Zhan, Tianzhang Xing, Xiaojiang Chen, Dingyi Fang, Rong Li, Xiaoqing Gong, and Feng Chen. 2016. Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality. *Mobile Information Systems* 2016 (2016), 1–14.
[41] Ouyang Zhang and Kannan Srinivasan. 2016. Mudra: User-friendly fine-grained gesture recognition using WiFi signals. In *12th International on Conference on Emerging Networking EXperiments and Technologies*.
[42] Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. 2016. Smokey: Ubiquitous smoking detection with commercial WiFi infrastructures. In *IEEE INFOCOM*.
[43] Xiaolong Zheng, Jiliang Wang, Longfei Shangguan, Zimu Zhou, and Yunhao Liu. 2017. Design and implementation of a CSI-based ubiquitous smoking detection system. *IEEE/ACM Transactions on Networking* 25, 6, pp (2017), 3781–3793.
[44] Zimu Zhou, Longfei Shangguan, Xiaolong Zheng, Lei Yang, and Yunhao Liu. 2017. Design and implementation of an RFID-based customer shopping behavior mining system. *IEEE/ACM Transactions on Networking* 25 (2017), 2405–2418.

Received April 2019; revised November 2019; accepted December 2019