

# Air-writing via Receiver Array Based Ultrasonic Source Localization

Hui Chen, *Student Member, IEEE*, Tarig Ballal, *Member, IEEE*, Ali H. Muqaibel, *Senior Member, IEEE*, Xiangliang Zhang, *Member, IEEE*, and Tareq Y. Al-Naffouri *Senior Member, IEEE*

**Abstract**—Air-writing systems have recently been proposed as tools for human-machine interaction where instructions can be represented using letters or digits written in the air. Different technologies have been used to realize air-writing systems. In this paper, we propose an air-writing system using acoustic waves. The proposed system consists of two components: a motion tracking component, and a text recognition component. For motion tracking, we utilize direction-of-arrival (DOA) information. An ultrasonic receiver array tracks the motion of a wearable ultrasonic transmitter by observing the change in the DOA of the signals. We propose a novel 2-D DOA estimation algorithm that can track the change in the direction of the transmitter using measured phase-differences between the receiver array elements. The proposed phase-difference projection (PDP) algorithm can provide accurate tracking with a 3-sensor receiver array. The motion tracking information is passed next for text recognition. To this end, and in order to strike the desired balance between flexibility, processing speed, and accuracy, a training-free order-restricted matching (ORM) classifier is designed. The proposed air-writing system, which combines the proposed DOA estimation and text recognition algorithms, achieves a letter classification accuracy of 96.31%. The utility, processing time, and classification accuracy are compared with four training-free classifiers and two machine learning classifiers to demonstrate the efficiency of the proposed system.

**Index Terms**—Acoustic localization, tracking, direction-of-arrival, phase-difference, receiver arrays, air-writing, sequence classification, human-machine interaction

## I. INTRODUCTION

Human gestures are indispensable tools for expressing emotions and conveying information to the environment. As alternatives to traditional keyboards, touchpads or other pressing and touching tools, new technologies based on cameras, acceleration sensors, photosensors, electromagnetic and acoustic signals are emerging as new mediums of interaction using gestures [1]. These systems have offered unorthodox ways for human-machine interaction. In the same vein, the concept of *air-writing* has been introduced as a flexible means of touchless human-machine interaction. This new technology has huge potentials in education, entertainment, smart home, and virtual reality applications [2], [3].

In recent years, a few air-writing systems have been developed. Xin et al. [4] realized a finger-writing-in-the-air with a

H. Chen, T. Ballal, X. Zhang and T. Al-Naffouri are with the Division of Computer, Electrical and Mathematical Sciences and Engineering (CEMSE), King Abdullah University of Science and Technology (KAUST), Thuwal, 23955-6900, KSA (email: {hui.chen; tarig.ahmed; xiangliang.zhang; tareq.alnaffouri}@kaust.edu.sa).

A. Muqaibel is with the Electrical Engineering Department, King Fahd University of Petroleum and Minerals (KFUPM), Dhahran, 31261, KSA (email: muqaibel@kfupm.edu.sa).

Kinect Sensor using artificial neural network based depth-skin-background mixture model (DSB-MM). The system was tested on digits, uppercase English letters, lowercase English letters, as well as Chinese characters, with a reported classification accuracy of 92.00%, 94.62%, 86.15%, 78.46%, respectively. By utilizing inertial sensors, an air-writing system is proposed with a two-stage classification approach using support vector machines (SVM) and hidden Markov model (HMM) [5]. For a user-independent setup, classification accuracy of 89% was reported using pen-up segmentation combined with a statistical language model. Chen et al. [2], [6] collected data from an infra-red camera and inertial measurement unit (IMU) sensors to complete letter and word classification. By using the HMM algorithm, accuracy of 99.2% for word recognition and 98.1% for letter recognition were reported. A two-antenna RFID-tagged pen fulfilled the needs of air-writing by using the online handwriting recognition toolkit LipiTk for classification [7]. This pen offered another flexible interaction experience and reached an accuracy of 93.6%.

Generally speaking, air-writing is carried out in two steps. Motion is first tracked by measuring absolute or relative positions. Next, classification methods are applied to recognize the written text. Usually, normalization and feature extraction are performed on the data before sending it to the classifiers.

### A. Motion Tracking Techniques

In order to track the motion of an object, vision-based methods [8], [9] separate the target from the background and then extract its location information from the frames. Even though current commercial depth sensors [10], [11] improve against sensitivity to surrounding illumination conditions, the high computational complexity remains a challenge.

Acoustic positioning systems can estimate the location based on time-difference-of-arrival (TDOA) [12] or a combination of direction-of-arrival (DOA) and time-of-arrival (ToA) [13] information. Another solution is based on the Doppler effect. Relative movement is estimated from the Doppler frequency shifts suffered by the signal reflecting from the moving object [14], [15]. Similarly, WiFi-based systems utilize the phase shift of a radio signal to estimate relative movement [16], [17], or use TDOA to estimate the 2-D object location [18]. These systems work well for simple recognition tasks while taking advantage of off-the-shelf products, such as mobile phones or laptops. However, a subtle movement is hard to capture using the above-mentioned systems.

IMU-based systems such as data gloves [19], [20] are able to detect even fingertip movements. Lighthouse utilizes photo-

sensors and IMUs to provide a highly accurate six degree-of-freedom (DoF) tracking solution for virtual reality (VR) [21]. In exchange of high accuracy, these systems are expensive in price, bulky to wear and require more computational resources, which excludes a majority of ordinary users.

All these technologies offer various distinct features and can be compared based on various metrics such as accuracy, resolution, latency, motion range, user comfort, and cost [22]. In this work, we attempt to realize an accurate motion tracking and text recognition system with low cost and low latency such that the system can operate in real-time.

We propose a low cost, privacy-protection, fast yet accurate air-writing system, which is based on tracking the DOA of acoustic signals transmitted by a handheld device. Acoustic waves provide high accuracy in localization and tracking due to their low propagation velocity [23], [24]. Estimating the location of a target can be realized by utilizing multiple anchor nodes with trilateration, triangulation, or multilateration algorithms [25]. However, tracking a target with its DOA information can lower the complexity and the cost of the system [23], [26].

DOA estimation algorithms can be classified into spectral-based and parametric approaches [27]. The former category creates some spectrum-like functions to calculate DOA in a computationally efficient way with representative methods such as beamforming [28]–[30], and MUSIC [31]. The latter methods, however, offer more accurate and robust estimations in exchange for high computational complexity. This group includes deterministic maximum likelihood (DML) [32]–[34] methods, and uniform linear arrays (ULA) based approaches such as root-MUSIC [35] and ESPRIT [36], [37]. Spectral-based and ULA-based methods, generally speaking, do not perform well when applied to arrays of small number of elements, especially in realistic signal and noise levels. On the other hand, DML methods are capable of handling such situations. Particle swarm optimization (PSO) [33] is proposed as an accurate method for DML DOA estimation; unfortunately, the high accuracy is associated with high processing cost. Phase-difference and time-delay methods, another group in the DML category, offer accurate source localization using small receiver configurations with moderate computational cost [38]–[41]. A basic system requirement for the air-writing application dictates that a low-cost DOA estimation algorithm be devised using a minimal number of sensors

In a previous work [42], a 1-D phase-difference based DOA estimation method was proposed by the authors. An exhaustive grid search is applied to find the DOA that best matches the wrapped phase-difference observations. Even though this algorithm shows remarkable performance, its accuracy is naturally limited by the search step. To achieve high resolution, a computationally expensive search is required.

To improve both the accuracy and computational speed, we propose a novel phase-difference projection (PDP) based algorithm for DOA estimation. This algorithm offers high estimation accuracy with low processing time. The proposed PDP algorithm makes up the motion tracking part of the proposed air-writing system.

## B. Classification Methods

As has been alluded to earlier, the second component of an air-writing system is text recognition that is achieved by processing the localization/motion-tracking results. For air-writing recognition, identifying the letters is the first task for the system since letters are the very elementary composition of words and sentences. Classifiers for letters can be divided into two groups depending on the need for training or not. By creating templates for all the possible alphabets, training-free classifiers can recognize letters based on distance or similarity measures (template matching) between an observed pattern and a set of templates [8]. Dynamic time warping (DTW) [43], [44] is a classical algorithm to calculate the distance between an observed sequence of data and a template. Cross-correlation [42] also indicates similarity between two sequences. On the other hand, machine learning tools, such as artificial neural network (ANN) [45] and hidden Markov models (HMM) [2], are training-based methods, where labeled observations are needed to learn the underlying input-output model and set up the algorithm for the classification of the future incoming data. An adequate amount of (training) data needs to be collected to make the model generalizable to diverse writing styles or patterns.

Each of the above-mentioned methods has its limitations. Dictionary matching based algorithms provide limited accuracy, and their performance degrades with the increase of the number of classes (e.g., extended alphabet system). Machine learning-based methods such as ANN and HMM offer high classification accuracy with low processing time; however, the models are built on a large amount of training data. These methods cannot recognize a newly defined or user-customized patterns. DTW provides a training-free and accurate solution, but the processing time is too high for real-time operation. In this paper, we propose a training-free order-restricted matching (ORM) algorithm, which requires low processing time while maintaining high classification accuracy. The proposed classification algorithm will be compared with a set of widely-used benchmark algorithms.

## C. Paper Contributions

In this work we: 1) propose a novel phase-difference projection (PDP) based 2-D DOA estimation algorithm for multi-frequency signals; 2) propose a training-free order-restricted matching (ORM) classifier for the non-binary (multi-class) case; 3) Consolidate the proposed PDP and ORM methods to implement an air-writing system.

This is an extended version of our previous work [42], [46]. Part of this work was demonstrated in ICASSP2018 demo session as DEMO-3-1: “UBAS: An Ultrasound Based Air-writing System” [47]. This system can recognize English letters written in a predefined style. However, air-writing is only one possible application of the proposed ultrasonic motion tracking system. Other applications of the proposed system include virtual/augmented reality, human activity recognition, and so on. A video demonstration of the proposed system’s in real-time applications such as mouse/keyboard functions and air-writing can be found on YouTube [48].

## D. Paper Organization

This paper is organized as follows. Section II provides a high-level description of the proposed air-writing system. The proposed PDP algorithm for DOA estimation is detailed in Section III. Motion data translation and orientation feature extraction are discussed in Section IV. The proposed text recognition ORM algorithm is explained in Section V. Section VI presents the simulation and the experimental results of the proposed air-writing system. Section VII concludes the whole work and highlights the most prominent future directions.

## II. THE PROPOSED AIR-WRITING SYSTEM

In this section, we present a high-level overview of the proposed air-writing system. Detailed descriptions of various system components will be given in subsequent sections. A block diagram of the proposed ultrasonic air-writing system is shown in Fig. 1. The system function is made up of two main phases: motion tracking and text recognition.

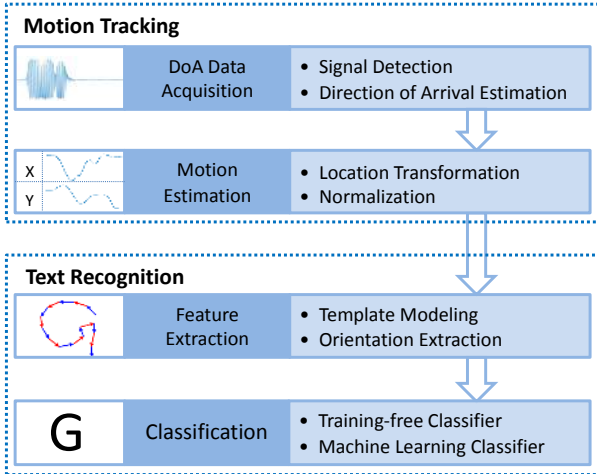


Fig. 1. Block diagram of the proposed air-writing system with two main modules: motion tracking and text recognition.

In the motion tracking phase, a multi-frequency ultrasonic signal is transmitted from a handheld device. A receiver array detects and labels the hand status as idle or active by thresholding the received signal strength. If the status is marked as active, a proposed PDP DOA estimation algorithm is applied to acquire the 2-D angle information of the moving transmitter.

An illustration of a gesture tracking is shown in Fig. 2. The receiver array center point  $O$  is located at  $[0, 0, 0]$ , which represents the origin of the local coordinate system of the array. The local  $x$  and  $y$  axes form a plane that is approximately parallel to the plane where the movement occurs, while the  $z$  axis (not plotted in the figure) points towards the target forming a local 3-D coordinate system. All tracking information is measured relative to this coordinate system.

For any receiver array with more than two non-collinear receivers, the direction of a signal originating from a target can be represented using a pair of angles, a horizontal angle

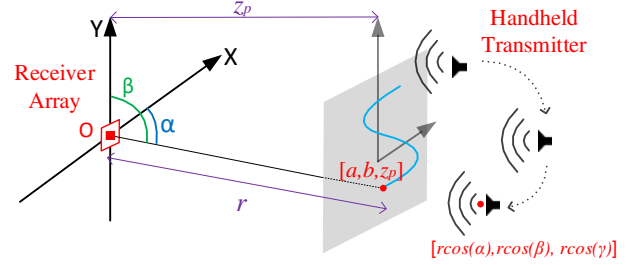


Fig. 2. Gesture motion tracking using DOA information.

$\alpha$  and a vertical angle  $\beta$ , as in Fig. 2. These angles represent the 3-D location of the target up to an unknown range, which is deemed irrelevant for the application of interest.

After acquiring the angle information, the position of the transmitter will be projected on a virtual 2-D plane at a point  $[a, b, d_p]$ , as in Fig. 2. The virtual 2-D plane is parallel to the receiver local coordinate system, and lies at a distance  $r_p$  from the origin. In other words, air-writing is recorded continuously in one stroke in a 3-D space and projected on a 2-D plane. This step is followed by a normalization operation aiming to unify the writing scale and get the data ready for classification.

Text recognition is performed in two steps: feature extraction and classification. Instead of utilizing only the motion pattern, features such as motion direction are also considered to provide additional information to our classifiers. A detailed description of the proposed system is discussed in the following sections.

## III. DOA ESTIMATION

This section presents the proposed method for 2-D DOA estimation using phase-difference information of signals observed at several pairs of receivers. We start by highlighting the signal and observation models needed for developing the 1-D DOA estimation algorithm. Subsequently, the concept of wrapped phase-difference pattern (WPDP) is introduced and utilized to develop the proposed PDP algorithm for DOA estimation. Next, how to estimate the 2-D DOA from several 1-D measurements will be explained.

### A. Signal and Observation Models

1) *Time-domain Model*: A handheld ultrasound transducer is used as a transmitter sending a series of signal blocks separated by silence periods. Each block consists of sinusoidal pulses with frequencies of  $f_1, f_2, \dots, f_F$ , where  $F$  is the total number of frequency components. These hops have the same duration  $T_h$  and are transmitted sequentially inside each block. The transmitted signal has a duty of  $(FT_h + T_p)$ , where  $T_p$  is the silence period duration at the end of each block. The purpose of the silence period is to minimize inter-block interference and its duration  $T_p$  should be chosen based on the acoustic reverberation time of the ambient environment. A transmitted signal block can be expressed in the continuous-time domain as

$$x(t) = \sum_{i=1}^F a_i(t) \cos(2\pi f_i t + \rho_i), \quad (1)$$

where  $a_i(t) = A_i$  for  $t \in \{(i-1)T_h, iT_h\}$  and zero otherwise;  $A_i$  and  $\rho_i$  are the amplitude and phase parameters of a transmitted pulse  $i$ .

At the receiver array side, receiver  $j \in \{1, 2\}$  acquires a signal  $y_j[n]$  which is discretized according to a sampling frequency  $F_s$ . The receiver continuously evaluates the received signal strength to detect the arrival of signal blocks (as opposed to pure background noise). If the detected signal strength exceeds a preset threshold, the receiver calculates the unsigned cross-correlation  $z_j[n]$  between the reference signal  $x[n]$ , which is a discrete-time version of the transmitted signal in (1), and the received signal  $y[n]$  as [24]

$$z_j[n] = \left| \sum_{m=1}^{M=FN_p} x[m] y_j[m+n] \right|, \quad (2)$$

where  $M$  is the number of signal samples per block, and  $N_p = \text{round}(T_h F_s)$ .

The received signal  $y_j[n]$ , is the convolution of the transmitted signal  $x[n]$  with the acoustic channel impulse response of the environment. We assume that, by design, the pulse duration is sufficiently small such that the multipath arrivals pertaining to each pulse interfere minimally with that same pulse. Based on this assumption, the presence of multiple signal arrivals due to acoustic reverberation results in multiple successive peaks of  $z_j[n]$  in (2) that are separated by at least one pulse duration. The location of the earliest significant peak of  $z_j[n]$  is used to indicate the start of the signal block. This decision can be made by combining the results from all the available receivers.

2) *Frequency-domain Model*: After the signal has been identified, the signal can be transformed to the frequency domain. For simplicity, let us assume that we have  $M = FN_p$  samples of  $y_j[n]$  that represents a received signal block. To guard against the effect of reverberation (that arrives later than the pulse), we compute the Discrete Fourier transform (DFT) for each pulse separately. That is

$$Y_{ji}(f_i) = \text{DFT}([y_j((i-1)N_p + 1, \dots, iN_p)]), \quad (3)$$

where the subscripts  $i$  and  $j$  indicates the pulse index (or frequency) in the received signal block, and the receiver number, respectively.

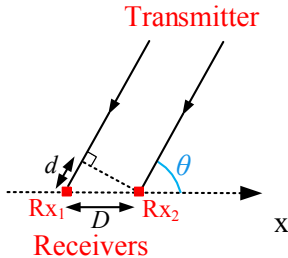


Fig. 3. The far-field model.

3) *Phase-difference Observations and DOA*: Denoting the distance to the target by  $r$ , and assuming that  $r \gg D$ , where  $D$  is the inter-sensor spacing as in Fig. 3, the situation simplifies to the *far-field* scenario [49]. Based on Fig. 3, for a certain DOA  $\theta$  and  $F$  frequency components, the elements of the noise-free phase-difference vector  $\phi = [\phi_1, \phi_2, \dots, \phi_F]^T$  satisfy the following relationship

$$\phi_i(\theta) = \frac{2\pi d(\theta)}{\lambda_i} = \frac{2\pi f_i \cos(\theta) D}{c}, 0 \leq \theta \leq \pi, \quad (4)$$

where  $\phi_i$  is the phase difference observed between the two receivers at frequency  $f_i$  which coincides with wavelength  $\lambda_i$ ,  $d$  is the difference in distance between the transmitter and each of the two receivers, and  $c$  is the speed of signal propagation. The angle  $\theta$  can be calculated if we know the value of the phase-difference  $\phi_i(\theta)$ . However, the true phase-difference cannot always be observed unless  $D < \lambda_i/2$ . In the general case, the observed phase-difference vector, call it  $\psi$ , is actually a wrapped version of the actual phase-difference vector  $\phi$ . The elements of the wrapped phase-difference vector are given by

$$\begin{aligned} \psi_i(\theta) &= \text{wrap}(\phi_i(\theta)) \\ &= \text{mod}(\phi_i(\theta) + \pi, 2\pi) - \pi \\ &= \phi_i(\theta) - 2\pi Z(\theta), \end{aligned} \quad (5)$$

where  $\text{mod}(\cdot, \cdot)$  returns the remainder of the division of the first argument by the second argument,  $Z(\theta)$  is an integer so that  $\psi_i(\theta) \in [-\pi, \pi)$ . In practice, we always have a noisy version of (5) since the observed (wrapped) phase difference is obtained through an estimation process.

Phase-difference can be estimated from the received signals using different techniques [38], [39], [50]. A simple technique is the frequency-domain approach where the phase-difference at frequency  $f_i$  is estimated using [39], [42], [51]

$$\hat{\psi}_i = \text{ang}(Y_{1i}(f_i) \cdot Y_{2i}^*(f_i)). \quad (6)$$

Here  $Y_{1i}$  and  $Y_{2i}$  are defined based on (3),  $(\cdot)^*$  is the complex conjugate operation and  $\text{ang}(\cdot)$  is the operation that returns the angle of a complex number. Since (3) is discrete in frequency,  $\hat{\psi}_i$  can be estimated from the closest available frequency to  $f_i$ .

4) *The Grid-Search Method*: With the observed phase-difference vector  $\hat{\psi}$ , DOA estimation can be obtained using a grid search method [38], [42]. This method starts with a hypothesized value of the DOA, say  $\theta = \omega$ . For that value, we calculate the corresponding wrapped phase-difference vector  $\psi(\omega)$  from (4) and (5) by setting  $\theta = \omega$ . By comparing  $\psi(\omega)$  with the observed phase-difference vector  $\hat{\psi} = [\hat{\psi}_1, \dots, \hat{\psi}_F]^T$  obtained from (6), the mismatch between the observed DOA and the hypothesized DOA can be conceived. The same procedure is repeated for different choices of  $\omega$  that cover the entire field of view of the receiver array. A DOA estimate can be obtained by picking  $\omega$  with the lowest mismatch as i.e.,

$$\hat{\theta} = \underset{\omega}{\text{argmin}} \sum_{i=1}^F |\hat{\psi}_i - \psi_i(\omega)|. \quad (7)$$

### B. The Wrapped Phase-Difference Pattern (WPDP)

The DOA estimation (7) suffers from two main drawbacks: a) Searching the entire angle space can be time consuming; and b) the accuracy is limited by the search resolution. Before discussing the proposed PDP algorithm, which overcomes these issues, we introduce the concept of wrapped phase-difference pattern, a key concept to the development of the proposed DOA estimation method.

To facilitate visualization, and without loss of generality, we start with the case where  $F = 2$ , i.e., only two frequency components are available. In the next section, we extend the algorithm to any number of frequencies. Let us take an example of a signal with acoustic frequencies  $\mathbf{f} = [f_1, f_2]^T = [20, 23]^T$  kHz received from a certain direction angle by a pair of receivers separated by a distance  $D$ . We assume a true DOA  $\theta \in [0^\circ, 180^\circ]$ .

In the noise-free case, for a signal impinging from an arbitrary direction  $\omega$ , the elements of the phase-difference vector  $\phi(\omega) = [\phi_1(\omega), \phi_2(\omega)]^T$  satisfy (4), while the corresponding wrapped phase-difference vector elements  $\psi(\omega)$  are given by (5). By using a combination of (4) and (5), we can calculate the wrapped phase-difference vector  $\psi(\omega)$  for any  $\omega \in [0^\circ, 180^\circ]$ . By discretizing the interval  $[0^\circ, 180^\circ]$ , e.g., by considering  $\omega_k = 0^\circ, 1^\circ, \dots, 180^\circ$ , we can obtain the corresponding sequence of vectors  $\psi_k = \psi(\omega_k)$ . We refer to the scatter plot of the sequence  $\psi_k$  as the *wrapped phase-difference pattern (WPDP)*. Examples of such display are shown in Fig. 5 (a) and (b) for different cases.

For  $D \leq \min(\lambda_1, \lambda_2)/2$  (e.g.,  $D = 0.74$  cm), phase-wrapping does not occur for all DOA angles resulting in  $\psi(\omega) = \phi(\omega), \forall \omega \in [0^\circ, 180^\circ]$ . A WPDP for such case is shown in Fig. 4 and is represented by the blue dotted line (WPD line). On this WPD line, a few points corresponding to selected values of  $\omega$  are highlighted with red circles. From Fig. 4, we observe that phase-difference pairs form a straight line that passes through the origin at  $\omega = 90^\circ$ , which conforms with (5). The whole WPDP, in this case, is represented by this single line. The boundaries of the display are set to be  $\pm\pi$ , as indicated by the green dashed box.

Now, let us consider the case where  $D > \min(\lambda_1, \lambda_2)/2$  (e.g.,  $D = 2.5$  cm). In this case,  $|\phi(\omega)|$  exceeds  $\pi$  and  $\psi(\omega)$  is wrapped into  $[-\pi, \pi)$  as per (5). This leads to breaking of the straight line representing the WPDP at the edges of the boundary box ( $\pm\pi$ ), resulting in several WPD lines parallel to the line that passes through the origin. The WPDP for  $D = 2.5$  cm is shown in Fig. 5 (a).

### C. The Proposed PDP DOA Estimation Algorithm

For a signal with  $F$  frequencies, the WPD lines share the same unit direction vector  $\mathbf{f}/\|\mathbf{f}\|$ , where  $\|\cdot\|$  is the  $\ell_2$  norm. We can form a projection hyperplane (this will be a line for  $F = 2$  as shown in Fig. 5 (b)) crossing the origin point that is perpendicular to all the WPD lines. It is easy to obtain the expression of this hyperplane as

$$\mathbf{f}^T \boldsymbol{\psi} = \psi_1 f_1 + \psi_2 f_2 + \dots + \psi_F f_F = 0. \quad (8)$$

The unit vector  $\mathbf{f}/\|\mathbf{f}\|$  is normal to the projection hyperplane. The distance between a point  $\boldsymbol{\psi}(\omega) =$

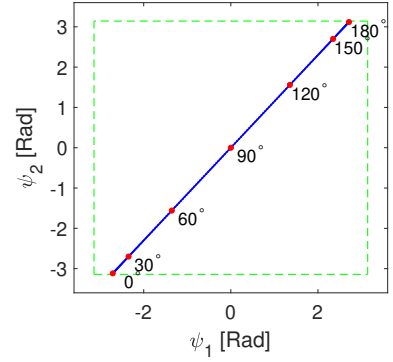


Fig. 4. Illustration of the WPDP for  $D = 0.74$  cm.

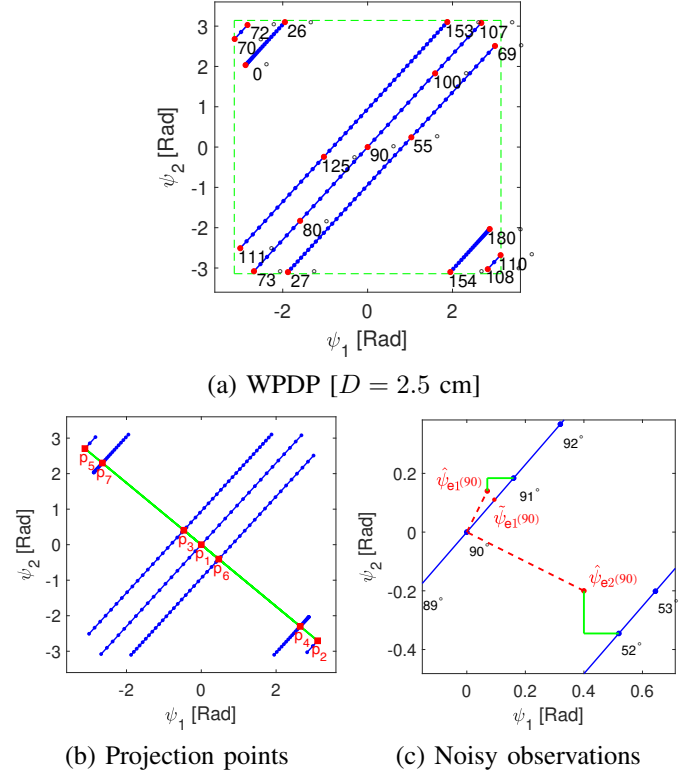


Fig. 5. Illustration of the WPDP for  $D = 2.5$  cm.

$[\psi_1(\omega), \psi_2(\omega), \dots, \psi_F(\omega)]^T$  and the projection hyperplane can be calculated as

$$\text{dis}[\boldsymbol{\psi}(\omega), \mathbf{f}^T \boldsymbol{\psi} = 0] = \frac{\mathbf{f}^T \boldsymbol{\psi}(\omega)}{\|\mathbf{f}\|}. \quad (9)$$

The projection point  $\mathbf{p}(\omega)$  of  $\boldsymbol{\psi}(\omega)$  on the projection hyperplane is given by

$$\mathbf{p}(\omega) = \boldsymbol{\psi}(\omega) - \frac{\mathbf{f}^T \boldsymbol{\psi}(\omega)}{\|\mathbf{f}\|} \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|}. \quad (10)$$

In the noise free case, all  $\boldsymbol{\psi}(\omega), \forall \omega \in [0^\circ, 180^\circ]$  project on a limited number of points. As an example in Fig. 5 (b), the projection points are marked using red squares. All the wrapped phase-difference corresponding to values of  $\omega$  from  $111^\circ$  to  $153^\circ$  are projected on  $\mathbf{p}_3$  (compare Fig. 5 (b) to Fig. 5 (a)). The total number of projection points  $\mathbf{p}_k$  ( $k$

$= 1, 2, \dots, K$ ) is fixed for a certain system setup and DOA estimation range. For simplicity, we assume that the DOA range for the target of interest is restricted to the interval  $[\omega_m, \pi - \omega_m]$ . In this case, the number of projection points is given by

$$K = 2 \sum_{i=1}^N \text{ceil} \left( \frac{\psi_i(\pi - \omega_m) - \pi}{2\pi} \right) + 1, \quad (11)$$

where  $\text{ceil}(\cdot)$  rounds the input to the nearest integer greater than or equal to that input.

To perform phase-difference unwrapping, we notice that for all points  $\psi(\omega_k)$  with the same projection point  $\mathbf{p}_k$  (i.e., points on the same line), the following relationship holds:

$$\phi(\omega_k) - \psi(\omega_k) = \mathbf{u}_k, \quad (12)$$

where  $\mathbf{u}_k$  is the *unwrapping vector* for a projection point  $\mathbf{p}_k$ , which is fixed for all points with the same projection point. We can use this vector to compensate an observed (wrapped) phase-difference  $\psi(\omega_k)$  to obtain the unwrapped phase-difference  $\phi(\omega_k)$ , as can be seen from (12). In the following discussion, we explain how the projection points  $\mathbf{p}_k$  and the corresponding unwrapping vector  $\mathbf{u}_k$  are computed.

The idea of obtaining projection points  $\mathbf{p}_k$  and the corresponding unwrapping vector  $\mathbf{u}_k$  is to find the border of different projection lines. Then the projection point and the unwrapping vector can be obtained from these border points using (10) and (12), respectively. The pseudocode for initializing  $\mathbf{u}$  and  $\mathbf{p}$  is shown in Algorithm 1. This algorithm searches the border point  $\psi_{b,k}$  of the  $k$ -th PD line based on the variable *ind* (except for  $b = 0$  which is the origin). By accumulating the *change* of the border transition and calculating their projection points of  $\psi_{b,k}$ ,  $\mathbf{u}_k$  and  $\mathbf{p}_k$  can be obtained.

---

**Algorithm 1** Initialize  $\mathbf{u}_k, \mathbf{p}_k$ 


---

```

1:  $j \leftarrow 1$ 
2:  $\psi_c, \phi_c, \text{change} \leftarrow \text{zeros}(1, N)$ 
3:  $\phi_{max} \leftarrow 2\pi df \cdot \sin(\pi - \omega_b)/v$ 
4: while  $\phi_c(i) < \phi_{max}(i)$  for all  $i = [1, 2, \dots, N]$  do
5:    $\psi_{b,j} \leftarrow \psi_c$ 
6:    $\mathbf{u}_j \leftarrow \text{change}$ 
7:    $j \leftarrow j + 1$ 
8:    $\text{ind} \leftarrow \arg \min_i [(\pi - \psi_c(i))/f_i]$ 
9:    $\psi_c \leftarrow \mathbf{f} \cdot (\pi - \psi_v(\text{ind}))/f_i + \psi_c$ 
10:   $\psi_c(\text{ind}) \leftarrow \psi_c(\text{ind}) - 2\pi$ 
11:   $\text{change}(\text{ind}) \leftarrow \text{change}(\text{ind}) + 2\pi$ 
12:   $\phi_c = \psi_c + \text{change}$ 
13:  $\mathbf{p}_k \leftarrow \text{project}(\psi_{b,k}), k \in \{1, 2, \dots, j-1\}$ 
14:  $\mathbf{p}_k \leftarrow -\mathbf{p}_{k-j+1}, k \in \{j, j+1, \dots, 2j-3\}$ 
15:  $\mathbf{u}_k \leftarrow -\mathbf{u}_{k-j+1}, k \in \{j, j+1, \dots, 2j-3\}$ 
16: return  $\mathbf{u}_k, \mathbf{p}_k$ 

```

---

Consider the pattern in Fig. 5 (d) where  $K = 7$ , we have  $\mathbf{u}_1 = [0, 0]^T$ ,  $\mathbf{u}_2 = [0, 2\pi]^T$ ,  $\mathbf{u}_3 = [2\pi, 2\pi]^T$ ,  $\mathbf{u}_4 = [2\pi, 4\pi]^T$ ,  $\mathbf{u}_5 = [0, -2\pi]^T$ ,  $\mathbf{u}_6 = [-2\pi, -2\pi]^T$  and  $\mathbf{u}_7 = [-2\pi, -4\pi]^T$ . In the noise-free case, for a given wrapped phase-difference pair  $\psi = [\psi_1, \psi_2]^T$ , we can find its projected point  $\mathbf{p}_k$

using (10) and then obtain the unwrapped phase-difference vector  $\phi$  as

$$\phi = \psi + \mathbf{u}_k. \quad (13)$$

For an observed noisy wrapped phase-difference vector  $\hat{\psi}$ , (10) returns only a perturbed projection point  $\hat{\mathbf{p}}$ . In this case, we choose the nearest projection point  $\mathbf{p}_k$  to  $\hat{\mathbf{p}}$  to compute the unwrapped phase-difference vector  $\hat{\phi}$  as

$$\hat{\phi} = \tilde{\psi} + \mathbf{u}_k = \mathbf{p}_k + \frac{\mathbf{f}^T \tilde{\psi}}{\|\mathbf{f}\|} \cdot \frac{\mathbf{f}}{\|\mathbf{f}\|} + \mathbf{u}_k, \quad (14)$$

where  $\tilde{\psi}$  is the nearest point on the WPD line with the projection point  $\mathbf{p}_k$ .

Take the WPD in Fig. 5 (d) for example. Assume  $\hat{\psi}_{e1}(90)$  is a noisy observed phase vector (the error is shown as red dashed line) of a target at  $\omega = 90^\circ$ . Grid search method estimates the DOA as  $91^\circ$  because of the minimum distance (the length of the solid green lines from the candidate to the noisy observation). The grid search method of equation (7) is actually calculating the Manhattan distance between the observed phase-difference vector with all the candidate PD points. Instead, the proposed PDP algorithm will find the nearest point  $\tilde{\psi}_{e1}(90)$  on the blue (phase-difference) line and estimate the DOA as  $90.6^\circ$ . However, if the noisy observation is  $\hat{\psi}_{e2}(90)$ , both algorithm will fail and have an erroneous estimation. We will use an indicator  $L_{min}$  to evaluate the performance of the system which will be explained in the next section.

With estimated unwrapped phase difference  $\hat{\phi}$ , the DOA can be estimated using any frequency component based on (4) as

$$\hat{\theta} = \cos^{-1} \left( \frac{c\hat{\phi}_i}{2\pi D f_i} \right). \quad (15)$$

#### D. 2-D DOA Estimation

The PDP algorithm described in the previous subsection is an effective way of estimating the 1-D DOA. However, this algorithm can also be used for 2-D DOA estimation by using at least three non-linearly placed sensors. In the following, we show how to obtain the 2-D DOA estimation of  $\alpha$  and  $\beta$  in Fig. 2. To this end, consider a 3-D space with a  $3 \times 1$  target direction unit vector  $\mathbf{t}$  defined as

$$\mathbf{t} = [\cos(\alpha), \cos(\beta), \cos(\gamma)]^T, \quad (16)$$

and a sensor vector matrix  $\mathbf{Q} = [\mathbf{q}_1, \dots, \mathbf{q}_R]$  describing all sensor pair baseline directions in unit vectors. Here,  $R$  is the number of pairwise combinations of all the sensors, and  $\mathbf{q}_i$  is the  $3 \times 1$  unit direction vector of the  $i$ -th sensor pair. The relationship between  $\mathbf{t}$  and  $\mathbf{q}_i$  can be stated as

$$\cos(\theta_i) = \frac{\mathbf{q}_i^T \mathbf{t}}{|\mathbf{q}_i| \cdot |\mathbf{t}|} = \mathbf{q}_i^T \mathbf{t}, i = 1, \dots, R, \quad (17)$$

where  $\theta_i$  is the angle between the target direction vector  $\mathbf{t}$  and the  $i$ -th baseline vector  $\mathbf{q}_i$  obtained using the PDP algorithm. For a set of 1-D DOA estimation  $\theta = [\theta_1, \dots, \theta_M]^T$ , we have

$$\cos(\theta) = \mathbf{Q}^T \mathbf{t} = \mathbf{Q}^T [\cos(\alpha), \cos(\beta), \cos(\gamma)]^T, \quad (18)$$

For a receiver array normal to the  $xy$  plane, the third component in  $\mathbf{q}_i$  will be zero. In order to have a unique solution for the unit vector  $\mathbf{t}$ , at least two pairs of sensors that are not collinear (at least three sensors) are required. By taking the first two rows of  $\mathbf{Q}$  and forming  $\tilde{\mathbf{Q}}$ , the first two elements of the unit direction vector  $\mathbf{t}$  of the target can be obtained simply using least square as

$$[\cos(\alpha) \ \cos(\beta)]^T = (\tilde{\mathbf{Q}}\tilde{\mathbf{Q}}^T)^{-1}\tilde{\mathbf{Q}}\cos(\boldsymbol{\theta}), \quad (19)$$

and the third element of  $\mathbf{t}$  can be obtained as

$$\cos(\gamma) = \sqrt{1 - \cos^2(\alpha) - \cos^2(\beta)}. \quad (20)$$

### E. Summary of the Proposed DOA Estimation Algorithm

The proposed DOA estimation algorithm can be performed through the following steps:

- 1) Calculate the  $K$  projection points  $\mathbf{p}_1, \dots, \mathbf{p}_K$  and the corresponding unwrapping vectors  $\mathbf{u}_1, \dots, \mathbf{u}_K$  using Algorithm 1;
- 2) Calculate the observed phase-difference  $\hat{\psi}$  using (6);
- 3) Calculate the projection point  $\hat{\mathbf{p}}$  for the observed (wrapped) phase difference  $\hat{\psi}$  using (10);
- 4) Find the closest projection point  $\mathbf{p}_k$  to  $\hat{\mathbf{p}}$ ;
- 5) Calculate the unwrapped phase vector  $\hat{\phi}$  using (14);
- 6) Calculate the DOA  $\hat{\theta}_i$  using (15) for each sensor pair.
- 7) Calculate 2-D DOA  $\hat{\alpha}$  and  $\hat{\beta}$  using (19)

## IV. MOTION ESTIMATION AND FEATURE EXTRACTION

In this section, we introduce two motion data processing procedures, namely, location transformation and normalization. In addition, orientation feature extraction from the translated motion data will also be described. The location vectors and corresponding orientation feature vector can be obtained from these pre-processing procedures. These vectors will be used for text classifications.

### A. Location Transformation

The location of the transmitter is given by  $r\mathbf{t} = [a, b, z]$  where  $\mathbf{t} = [\cos\alpha, \cos\beta, \cos\gamma]$  is the direction vector defined in (16) and  $r$  is the range (distance) between the transmitter and the receiver array. To obtain the values of the DOA parameters  $r$  and  $\mathbf{t}$ , three non-collinearly placed sensors are sufficient to accomplish this task.

However, estimating the range  $r$  needs synchronization between the transmitting and receiving systems to measure the time of flight of the signal. Our proposed system circumvents the need for range estimation, and the associated hassle of synchronization. Specifically, we assume<sup>1</sup> that the user stands at a fixed location  $z_p$ , which we set to 1 m. We can then write

$$\begin{cases} a = r \cos(\alpha) = z_p \frac{\cos(\alpha)}{\cos(\gamma)} \\ b = r \cos(\beta) = z_p \frac{\cos(\beta)}{\cos(\gamma)}. \end{cases} \quad (21)$$

<sup>1</sup>Our experiments show that even when this assumption is violated to some degree, the system is able to recognize the letters successfully. This is because the transformed 2D location from DOA estimation  $\alpha$  and  $\beta$  is sufficient to represent the target movement for further classification.

Since  $\gamma$  is related to  $\alpha$  and  $\beta$  through (20),  $a$  and  $b$  can be completely determined from  $\alpha$  and  $\beta$ . We collect all the transformed locations in two vectors  $\mathbf{a} = [a_1, a_2, \dots, a_L]$  and  $\mathbf{b} = [b_1, b_2, \dots, b_L]$ , where  $L$  is the length of samples for the written letter.

### B. Location Normalization

For different letters, the center position and writing duration may vary from one user to another. In order to make the collected information consistent, normalization is a necessary procedure. The location vectors  $\mathbf{a}$  and  $\mathbf{b}$  with length  $L$  (which might vary from one user to another) is linearly interpolated into two vectors of fixed length  $N$ . This step is necessary for the classifiers using location information so that the dimensionality of the data is consistent. Following this step, the DC component of each vector needs to be removed to make sure that all the vectors are centered around zero so that  $\sum \mathbf{a} = \sum \mathbf{b} = 0$ . After location transformation and normalization, we use the processed location vectors  $\mathbf{a} = [a_1, a_2, \dots, a_N]^T$  and  $\mathbf{b} = [b_1, b_2, \dots, b_N]^T$  to make inferences about the collected data and to identify the writing pattern present therein. These two vectors are then used to extract motion orientation feature and to recognize letters as we describe next.

### C. Orientation Feature Extraction

In our proposed method, the normalized location vectors  $\mathbf{a}$  and  $\mathbf{b}$  are not enough for text recognition. The motion orientation change is also of great utility. Specifically, we segment the process of writing a letter into several transitions and use the orientation feature of each transition to aid in classification. We define the orientation  $h$  between two locations  $(a_i, b_i)$  and  $(a_j, b_j)$  as

$$h_k = \text{atan2}(b_j - b_i, a_j - a_i), \quad (22)$$

where  $\text{atan2}(\cdot)$  is the four-quadrant inverse tangent function which returns the orientation in  $(-\pi, \pi]$ . Note that orientation  $h$  will only be registered if the distance between  $(a_i, b_i)$  and  $(a_j, b_j)$  is greater than a threshold  $T_{dis}$ . Thus, to calculate  $h_1$ , we first set  $i = 1, j = 2$  and check if the distance between  $(a_i, b_i)$  and  $(a_j, b_j)$  is greater than  $T_{dis}$ . If not, then  $j$  is incremented until this condition is satisfied. When the condition is satisfied,  $h_1$  is calculated using (22) and  $i, j$  are updated to calculate the value of  $h_2$ . The procedure is continued until  $j = N$ . This procedure, whose pseudo-code is shown in Algorithm 2, produces the vector  $\mathbf{h}$  which presents the orientation feature of the written letter.

## V. TEXT RECOGNITION

From previous sections, the estimated 2-D DOA data will be transformed into location data vectors  $\mathbf{a}$ ,  $\mathbf{b}$  and a feature vector  $\mathbf{h}$ . In this section, we first introduce the letter template. Then, the proposed ORM classifier will be detailed.

---

**Algorithm 2** Orientation Feature Extraction
 

---

```

1:  $i, k \leftarrow 1$ 
2: for  $j = 2$  to  $N$  do
3:    $dist \leftarrow \sqrt{(b_j - b_i)^2 + (a_j - a_i)^2}$ 
4:   if  $dist > T_{dis}$  then
5:      $h_k \leftarrow \text{atan2}(b_j - b_i, a_j - a_i)$ 
6:      $i \leftarrow j$ 
7:      $k \leftarrow k + 1$ 
8: return  $\mathbf{h}$ 
  
```

---

### A. Letter Templates

In this paper, we consider only ( $M = 26$ ) English uppercase letters. Example templates of the letters are shown in Fig. 6, where each letter is written with one stroke starting at the blue circle and ending at the red triangle. The figure shows examples of letter writing templates and the temporal progression of the writing, which is indicated by the gradual color change (from blue to red).

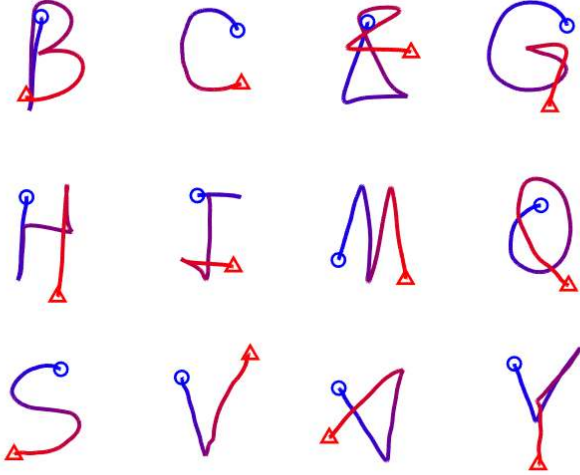


Fig. 6. Some letter templates. The writing starts with a circle and ends with a triangle.

By knowing this general temporal structure of each letter, one can produce the corresponding location vectors  $\bar{\mathbf{a}}_m$ ,  $\bar{\mathbf{b}}_m$  and the feature vector  $\bar{\mathbf{h}}_m$  for the  $m$ -th letter template,  $m = 1, 2, \dots, M$ . Note that  $\bar{\mathbf{a}}_m$  and  $\bar{\mathbf{b}}_m$  have the same length  $N$  for different letter templates (due to the normalization procedure described above), but the length of  $\bar{\mathbf{h}}_m$  may vary for different values of  $m$ . Fig. 7 shows examples of the motion patterns and orientation features of letters ‘D’ and ‘W’. These templates serve two purposes: they unify the writing styles for users and provide a reference for training-free classifiers.

### B. The Order-Restricted Matching Classifier

In this subsection, we propose a light, accurate and flexible classifier that allows the users to add customized letters without training the classifier again. The classifier is called order-restricted matching (ORM) classifier inspired by the finite state machine (FSM) method, which defines a set of states and a set of transitions for the patterns [52]. Here, instead of defining

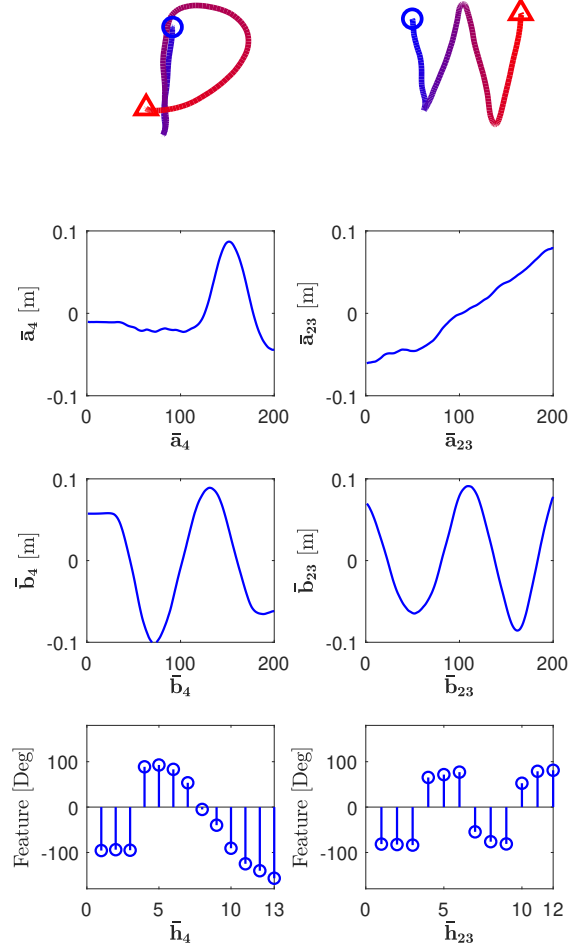


Fig. 7. Examples of the motion pattern for the letters ‘D’ and ‘W’. Note that the length of  $\bar{\mathbf{a}}_m$  and  $\bar{\mathbf{b}}_m$  is fixed ( $N = 200$ ) but the length of  $\bar{\mathbf{h}}_m$  is varied.

fixed states, we use letter-specific states which are defined by the orientation features.

The general idea of this classifier is to compare the observed sequence vector  $\mathbf{h}$  with all the template feature sequences  $\bar{\mathbf{h}}_m$  ( $m = 1, \dots, M$ ) in an order-restricted way. The sequence with the minimum accumulated distance measurement  $Dis$  is chosen as the classification result. Part of the difficulty in comparing  $\mathbf{h}$  with the templates  $\bar{\mathbf{h}}_m$  is that these vectors might not have the same length due to different writing speeds or styles. Our proposed ORM algorithm is able to solve this issue.

To describe the algorithm, let  $L$  be the length of  $\mathbf{h}$  and  $L_m$  be the length of  $\bar{\mathbf{h}}_m$ . As mentioned above, in general  $L \neq L_m$ . The distance between the observed orientation feature  $h_i$  and the template orientation feature  $\bar{h}_{m,j}$  is calculated as

$$dist(h_i, \bar{h}_{m,j}) = |\text{wrap}(h_i - \bar{h}_{m,j})|. \quad (23)$$

where  $\text{wrap}(\cdot)$  is defined in (5) and it is used because the elements of  $\mathbf{h}$  and  $\bar{\mathbf{h}}_m$  are in radians (or degrees). Now, two features  $h_i$  and  $\bar{h}_{m,j}$  are *matched* if the distance between the two features is smaller than a threshold  $T_{ang}$ .

We define the comparison status as *Start*, *Ongoing* and *Stop*. The status of each comparison is initialized as *Start*, and it turns to *Ongoing* if the first template feature  $\bar{h}_{m,1}$  is



matched. The status will be set as *Stop* when the last template feature  $\bar{h}_{m,L_m}$  is matched.

To obtain the accumulated distance measurement  $Dis$  with a given template feature vector  $\bar{\mathbf{h}}_m$ , we set  $i, j = 1$  and  $Dis = 0$ . The index of the observed sequence vector  $i$  increases from 1 to  $L$  to compare with the template feature  $\bar{h}_{m,j}$ . For each comparison between the two features  $h_i$  and  $\bar{h}_{m,j}$ ,  $Dis$  changes depending on current status:

1) *Start*: If features  $h_i$  and  $\bar{h}_{m,j}$  are matched in *Start* status, then current status moves to *Ongoing* and  $j$  is incremented. Otherwise, we remain at *Start* status and  $Dis$  is incremented by 1.

2) *Ongoing*: If  $h_i$  and  $\bar{h}_{m,j}$  are matched in *Ongoing* status,  $j$  is incremented unless it equals to  $L_m$  (if  $j$  reaches  $L_m$ , the status is then changed to *Stop*). If on the other hand  $h_i$  does not match both  $\bar{h}_{m,j}$  and  $\bar{h}_{m,j-1}$ ,  $Dis$  is incremented by 1.

3) *Stop*: In *Stop* status,  $Dis$  is only incremented if  $h_i$  and  $\bar{h}_{m,L_m}$  are not matched. We keep incrementing  $i$  and comparing with  $\bar{h}_{m,L_m}$  until  $i$  reaches its maximum value  $L$ .

The pseudo-code of the ORM distance measurement is given in Algorithm 3. The ORM classifier calculates the distances between the observed sequence and all  $M$  templates. Among the candidates with *Stop* status, the one with the smallest distance will be the classification result.

---

#### Algorithm 3 ORM Distance Measurement

---

```

1:  $j \leftarrow 1$ 
2:  $Dis \leftarrow 0$ 
3:  $status \leftarrow start$ 
4: for  $i = 1$  to  $L$  do
5:    $matched \leftarrow dist(h_i, \bar{h}_{m,j}) < T_{ang}$ 
6:   if  $status = Start$  then
7:     if  $matched$  then
8:        $status \leftarrow Ongoing$ 
9:        $j \leftarrow j + 1$ 
10:    else
11:       $Dis \leftarrow Dis + 1$ 
12:    else if  $status = Ongoing$  then
13:      if  $matched$  then
14:        if  $j = L_m$  then
15:           $status \leftarrow Stop$ 
16:        else
17:           $j \leftarrow j + 1$ 
18:        else if  $dist(h_i, \bar{h}_{m,j-1}) > T_{ang}$  then
19:           $Dis \leftarrow Dis + 1$ 
20:        else if  $matched = false$  then
21:           $Dis \leftarrow Dis + 1$ 
22: return  $Dis$ 

```

---

An example of ORM distance measurement with a threshold  $T_{ang} = 45^\circ$  is shown in Fig. 8 (the orientation feature is represented in degree for simplicity). In this figure, the dashed red arrow indicates failed attempts matching the template sequence, while the solid green arrow means a successful matching. The status is first set as *Start*. The first feature of the observed sequence is a ‘*Start error*’ because  $120^\circ$  does not match current template feature  $10^\circ$  ( $j = 1$ ). The orientation

feature  $-70^\circ$  is an ‘*Ongoing error*’ because it does not match current template feature  $120^\circ$  ( $j = 4$ ) and the previous feature  $90^\circ$ . The feature  $30^\circ$  is a ‘*Stop error*’ because the status is marked as *Stop* and it does not match the last template feature  $120^\circ$ . The detailed information of all the comparisons is shown in Table I. This algorithm provides a training-free classification solution with high accuracy and low computational time.

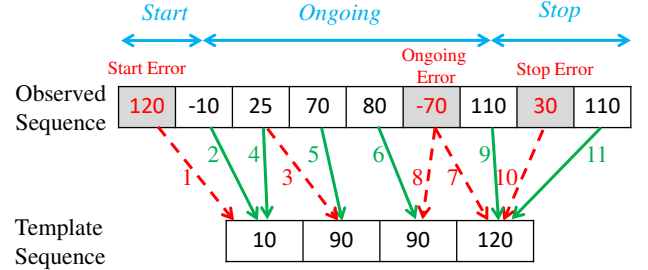


Fig. 8. An illustration of the ORM algorithm.

TABLE I  
DETAILS OF THE ORM ALGORITHM EXAMPLE IN FIG. 8

Comparison	i	j	matched	status	Dis
1	1	1	False	Start	1
2	2	1	True	Ongoing	1
3	3	2	False	Ongoing	1
4	3	1	True	Ongoing	1
5	4	2	True	Ongoing	1
6	5	3	True	Ongoing	1
7	6	4	False	Ongoing	1
8	6	3	False	Ongoing	2
9	7	4	True	Stop	2
10	8	4	False	Stop	3
11	9	4	True	Stop	3

## VI. SIMULATION AND EXPERIMENTAL RESULTS

In this section, we start by evaluating the performance of the proposed PDP algorithm for DOA estimation. Following that, we will present our results for the proposed air-writing system that uses the PDP algorithm for localization.

### A. Benchmark Algorithms

1) *Benchmark DOA Algorithms*: The PDP algorithm is compared with four benchmark algorithms, namely, Dual Freq [38], MUSIC [31], PSO [33] and Grid Search [42]. Both MUSIC and Grid Search use a searching resolution of  $0.5^\circ$ . The swarm size of the PSO algorithm is chosen as 50.

2) *Benchmark Classifiers*: The proposed ORM algorithm is compared with four training-free methods, namely, redundant dictionary (RD) [42], dynamic time warping (DTW) [43], [44] and thier modified version, redundant dictionary with a decision tree (RD-DT), and direction-based dynamic time warping (D-DTW). Two other learning-based classification methods, namely, ANN [53] and HMM [54], are also used as benchmarks. The implementation of these methods is detailed in the Appendix.

## B. DOA Estimation Results

1) *DOA Estimation Simulation Results:* In simulation, the sampling frequency is chosen as  $F_s = 96$  kHz. The signal parameters are chosen as, signal frequency vector  $\mathbf{f} = [20, 23]^T$  kHz, signal hop duration  $T_h = 1$  ms, silent duration  $T_p = 8$  ms. The inter-sensor distances are chosen as  $D = 2.5$  cm.

The simulation is repeated 1000 times for each DOA. The root mean square error (RMSE) of different DOAs at different SNRs is shown in Fig. 9. The average processing time for different algorithms (processed in Matlab R2017b running on a DELL T7500 workstation) is shown in Table II.

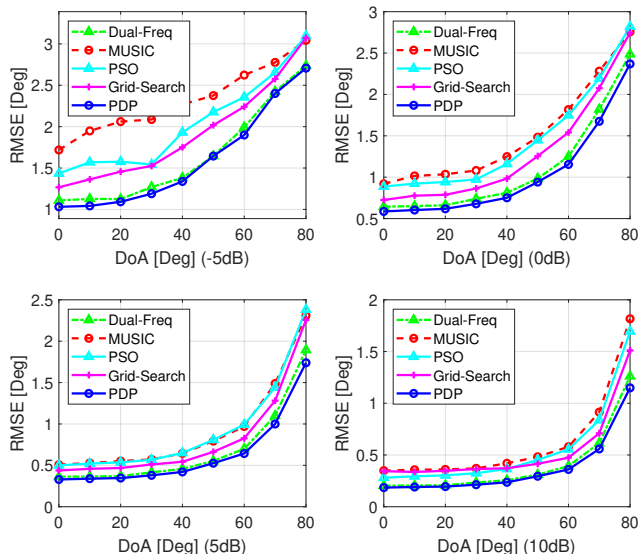


Fig. 9. RMSE for different DOA algorithms at different SNRs.

TABLE II  
AVERAGE PROCESSING TIME OF DIFFERENT ALGORITHMS

Algorithm	Dual Freq	MUSIC	PSO	Grid Search	PDP
Time [ms]	0.02	12.21	48.96	1.04	0.06

Based on the simulation results, MUSIC and PSO do not work well for solving the 1-D DOA problem using a pair of sensors. The Dual Freq method requires the minimum processing time, but its performance is not as good as that of PDP. The PDP algorithm, offers a good trade-off between processing time and performance. In the next section, we will apply PDP to estimate the 2-D DOA of a moving target in an experimental setup.

2) *Moving Target DOA Estimation Experimental Results:* Experiments with a moving target were carried out in an office room with a temperature of around  $24^\circ\text{C}$ . We use *PioneerTS - T110* tweeter [55] as the transmitter and *SPH0611LR5H* microphones [56] as the receivers. A receiver array of two orthogonal pairs of sensors was used and the signal parameters are the same as those used in the simulations.

The DOA estimation accuracy of the system is evaluated by tracking the motion of the handheld transmitter using both ultrasound and an *ART-TRACK5* camera tracking

system [57]. The camera tracking system estimates the ground truth locations of the tracking markers attached to the transmitter with a resolution of 0.1 mm. 50-thousand points were collected while the user was standing in front of the receiver array and moving the transmitter within a square writing area. The setup is shown in Fig. 10.

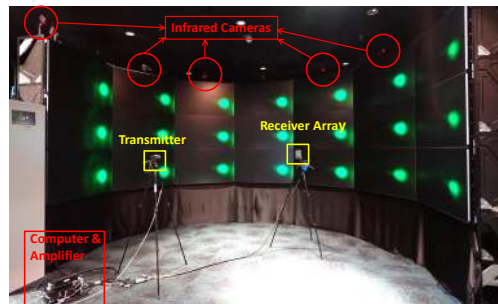


Fig. 10. Infrared camera-based system for ground truth measurement.

In this test, each sample was recorded with ground truth, including DOA and speed information. Part of the results (2000 estimations) is shown in Fig. 11. We can see from the figure that the estimations fit the ground truth well. There exist some obvious outliers which could be removed by using simple thresholding. The overall RMSE of the horizontal DOAs and vertical DOAs estimations are  $2.75^\circ$  and  $3.85^\circ$ , respectively. Several reasons may affect performance: calibration of the tracking markers, time synchronization between the camera and the ultrasonic estimation, multipath, and Doppler effect. In the following subsection, we show that the achieved level of localization accuracy is sufficient to complete the letter recognition task with high success rate, which confirms the effectiveness of the proposed system.

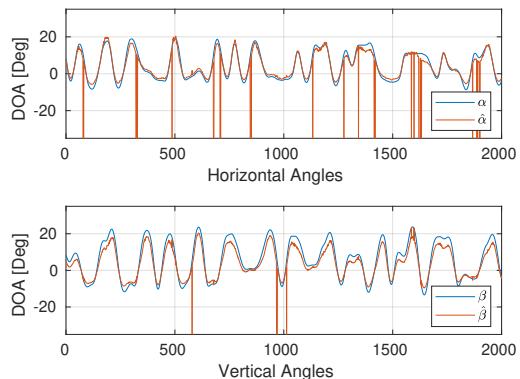


Fig. 11. DOA Estimation RMSE for Different Target DOAs.

## C. Letter Recognition Results

Volunteers were recruited and familiarized with the letter writing templates presented in Fig. 6. They were first-time users of the system with no previous experience or training on a similar system. The volunteers were asked to follow the templates to complete the tests. Each letter was repeated 10 times, and 3120 letter data was collected from 12 different users.

The average writing time for each letter is shown in Fig. 12. We can see that for each letter, the number of stable features and its average writing time is positively correlated. Considering that all the volunteers were first-time users of the system, the average writing time is reasonable for interactive input.



Fig. 12. Average writing time for different letters.

All the collected data was used to evaluate the performance of the training-free classifiers RD, DTW, and ORM. For ANN and HMM, performance was evaluated using  $k$ -fold cross-validation with 10 subsets. For the ORM algorithm,  $T_{dis}$  is chosen as 0.008 for the feature extraction of the observed sequences, and 0.032 for the template sequence. Adopting a larger  $T_{dis}$  for the template sequences helps to obtain stable features which improves classification performance.  $T_{ang}$  is chosen as  $45^\circ$  in this case.

The classification results and the processing time for the six classifiers are shown in Table III. From this table, it can be seen that D-DTW has the best classification accuracy of 97.12%. However, RD-DT, ANN, HMM, and ORM have reasonable accuracy (96.31%) and much reduced recognition time  $T_{avg}$ . Considering the fact that the design of a decision tree and the training of the machine learning classifiers make the system non-extendable, ORM is more practical for real-time applications.

TABLE III  
CLASSIFIER EVALUATION

Method	Data Type	Trained	$T_{avg}$ (ms)	Accuracy (%)
ORM	<b>h</b>	No	1.8	96.31
RD	<b>a,b</b>	No	0.1	90.83
RD-DT	<b>a,b</b>	Yes	1.2	95.83
HMM	<b>h</b>	Yes	6.9	97.05
ANN	<b>a,b</b>	Yes	15.4	96.12
D-DTW	<b>h</b>	No	81.5	97.12
DTW	<b>a,b</b>	No	154.3	94.42

The letters with the lowest classification accuracy and their most likely confused targets using the ORM classifier are shown in Table IV. The letter ‘Y’ has the lowest classification accuracy of 75.83%, and it is most likely to be detected as the letter ‘X’.

#### D. Utility Test

The utility test is conducted to verify the practicability of the DOA estimation algorithm and the ORM classifier. A real-time air-writing system is realized by using an ultrasound

TABLE IV  
CLASSIFICATION CONFUSING CASES

Letter	Accuracy(%)	Confused Targets (%)
Y	75.83	X (21.67)
K	81.67	F (18.33)
C	90.00	L ( 5.83)
J	90.00	I ( 5.83)
F	95.00	K ( 3.33)
O	95.00	X ( 5.00)
G	95.83	O ( 4.17)
X	96.67	Y ( 2.50)

transducer driven by an LM386 audio amplifier controlled by an ESP32-WROOM micro-controller, as shown in Fig. 13. The processing task is completed on a computer using a python code. This air-writing system is capable of completing mouse and keyboard tasks such as moving cursor, clicking, dragging, typing, and air-writing. The demonstration video can be found on YouTube [48].



(a) Ultrasound transmitter



(b) Video screenshot

Fig. 13. Air-writing prototype and video screenshot.

## VII. CONCLUSION

This work proposed an ultrasonic air-writing system based on acoustic source motion tracking. Motion information is extracted from multi-frequency ultrasonic signals using a proposed phase-difference projection (PDP) based DOA estimation algorithm. A training-free order-restricted matching (ORM) classifier, intended to meet the requirement of low processing time and high accuracy, achieves a 96.31% letter recognition accuracy and a 1.8 ms average processing time.

Several aspects of the proposed system can be improved as part of future extensions. To this end, extending the proposed DOA estimation algorithm to the multi-source case will be considered to enable multi-user tracking. On the other hand, improving text recognition represents another possible futuristic improvement. In this regard, integrating natural language processing (NLP) tools is a promising direction.

## APPENDIX

### IMPLEMENTATION OF BENCHMARK CLASSIFIERS

#### A. Redundant Dictionary (RD)

The variability of writing dictates that there are many candidate vectors  $\mathbf{a}$  and  $\mathbf{b}$  that represent a certain letter, even when the user writes following the same writing approach. A redundant dictionary extends the template matrices  $\mathbf{A} = [\mathbf{a}_1, \dots, \mathbf{a}_M]$ ,  $\mathbf{B} = [\mathbf{b}_1, \dots, \mathbf{b}_M]$  to  $\mathbf{A}_R$ ,  $\mathbf{B}_R$  using several transform methods [42]. In this way, letters with different writing styles and writing speed can be recognized.

Classification can be done by a simple matrix multiplication as

$$\mathbf{v} = \bar{\mathbf{A}}_R^T \mathbf{a} + \bar{\mathbf{B}}_R^T \mathbf{b}. \quad (24)$$

The maximum value in the vector  $\mathbf{v}$  indicates the RD column with the highest similarity to the observed angle vector. In the experiments, the size of matrices  $\bar{\mathbf{A}}_R$  and  $\bar{\mathbf{B}}_R$  was  $546 \times 200$  (546 redundant templates with a length of 200 for horizontal and vertical locations).

### B. Redundant Dictionary with Decision Tree (RD-DT)

The RD provides a simple way of performing training-free classification. However, the performance of this method may suffer in some cases. For example, if the user does not emphasize the last orientation (going down) in the letters ‘M’, this letter might be confused with the letter ‘N’. This issue can be solved by utilizing the orientation sequence from (22). If the detected letter is ‘M’ or ‘N’, ‘M’ will be chosen if the last element of the direction vector is pointing down and vice versa. Seven similarity branches  $(B, F, K)$ ,  $(I, J, T)$ ,  $(V, W, U)$ ,  $(H, B)$ ,  $(M, N)$ ,  $(P, R)$ , and  $(T, Y)$  are included to form a decision tree after acquiring the classification results from the RD.

### C. Dynamic Time Warping (DTW)

DTW is an algorithm designed to measure the similarity between two sequences [43], [44]. This algorithm works well for sequences with variable lengths. The observed location sequence is compared with the 26 letter templates in  $\bar{\mathbf{A}}$ . The distance between the observed sequence and the template sequence is calculated, and the letter with the minimum DTW distance is chosen. The element distance  $s$  and cost matrix  $\mathbf{C}^m$  are defined as

$$s_{i,j}^m = \sqrt{(a_i - \bar{a}_{m,j})^2 + (b_i - \bar{b}_{m,j})^2}, \quad (25)$$

$$c_{i,j}^m = \begin{cases} \sum_{w=1}^j s_{i,w}^m & i = 1, \\ \sum_{w=1}^i s_{w,j}^m & j = 1, \\ u_{i,j}^m + s_{i,j}^m & \text{otherwise.} \end{cases} \quad (26)$$

Here,  $s_{i,j}^m$  is the distance between the  $i$ -th sample of the observed sequence and the  $j$ -th sample of the  $m$ -th template sequence,  $u_{i,j}^m = \min\{c_{i-1,j-1}^m, c_{i-1,j}^m, c_{i,j-1}^m\}$ . The value of  $c_{N,N}^m$  is the distance between the observed data and the  $m$ -th letter template and the letter with minimum distance is chosen as the classified result.

### D. Direction-based Dynamic Time Warping (D-DTW)

In real applications, there is a variety of writing styles, and thus, the angle data may not match well. Also, a large number of locations affect the processing time. In this case, orientation features are used to measure the distance between the observed vector and the template vectors. As a consequence, elements in the distance matrix  $\mathbf{S}^m$  from (25) can be calculated as  $s_{i,j}^m = |\text{wrap}(h_i - \bar{h}_{m,j})|$ , which is the angle distance as (23). This method provides a more robust way of classification and dramatically reduces the computational complexity for calculating the cost matrix  $\mathbf{C}^m$ .

### E. Artificial Neural Network (ANN)

In our ANN model, the raw angle data is directly used to train the neural network model. We have built a network with two autoencoders connected with a softmax layer. The parameters for the two autoencoders are chosen as ‘Hidden Size’=(100, 50), ‘Max Epochs’=(400, 100), ‘L2 Weight Regularization’=(0.004, 0.002), ‘Sparsity Regularization’=(4, 4) and ‘Sparsity Proportion’=(0.15, 0.1). The input of the model is the concatenation of the normalized target horizontal and vertical locations of size  $400 \times 1$ , and the output are the probabilities of all the  $M = 26$  classes. This model is trained with collected transformed location vectors and realized in MATLAB 2017b using Matlab Neural Network Toolbox [53].

### F. Hidden Markov Model (HMM)

As opposed to the previously discussed classifiers, an HMM models the writing process as a sequence of interconnected states. We partition the angular space of the extracted features into eight sections, each section represents one state, for example, the angle interval  $(-\pi, 3\pi/4]$  represents state 1,  $(-3\pi/4, -\pi/2]$  represents state 2, and continue till state 8,  $(3\pi/4, \pi]$ . One example of the state feature extraction procedure is shown in Fig. 14, where the letter ‘G’ is discretized as a sequence  $[7, 8, 8, 1, 2, 3, 3, 4, 5, 5, 7, 5, 3, 2, 3]^T$ .

A left-right model [54] with a max jump step equal to 2 is used in this work to form an HMM classifier. The initial guess of the model states is based on the complexity of each letter. We use the following initialization for different letters: ‘A-G’ (8 9 4 6 7 7 8), ‘H-N’ (7 6 5 7 3 7 6), ‘O-T’ (7 7 9 8 7 4), and ‘U-Z’ (5 4 7 4 6 4). The classifier initializes the transition probability distribution and the observed symbol probability distribution randomly. Then the training data of each letter is used to train the corresponding HMM model. In total,  $M = 26$  models are trained for all the letters. To recognize an observed sequence, we pass it to each of the  $M$  trained models. Each model produces a probability indicating the level of fitness, and the observed sequence is assigned to the model that gives the maximum probability.

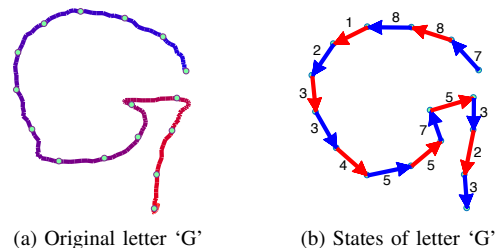


Fig. 14. State representation of the letter ‘G’.

### ACKNOWLEDGMENT

The authors would like to thank KAUST Visualization Core Lab for facilitating part of the experimental tests.

## REFERENCES

- [1] X. Zabulis, H. Baltzakis, and A. A. Argyros, "Vision-based hand gesture recognition for human-computer interaction." *The universal access handbook*, vol. 34, p. 30, 2009.
- [2] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—part I: Modeling and recognition of characters, words, and connecting motions," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 403–413, 2016.
- [3] M. R. Abid, E. M. Petriu, and E. Amjadian, "Dynamic sign language recognition for smart home interactive application using stochastic linear formal grammar," *IEEE Transactions on Instrumentation and Measurement*, vol. 64, no. 3, pp. 596–605, 2014.
- [4] X. Zhang, Z. Ye, L. Jin, Z. Feng, and S. Xu, "A new writing experience: Finger writing in the air using a kinect sensor," *IEEE MultiMedia*, vol. 20, no. 4, pp. 85–93, 2013.
- [5] C. Amma, M. Georgi, and T. Schultz, "Airwriting: Hands-free mobile text input by spotting and continuous recognition of 3D-space handwriting with inertial sensors," in *Wearable Computers (ISWC), 2012 16th International Symposium on*. IEEE, 2012, pp. 52–59.
- [6] M. Chen, G. AlRegib, and B.-H. Juang, "Air-writing recognition—part II: Detection and recognition of writing activity in continuous stream of motion data," *IEEE Transactions on Human-Machine Systems*, vol. 46, no. 3, pp. 436–444, 2016.
- [7] L. Shangguan and K. Jamieson, "Leveraging Electromagnetic Polarization in a Two-Antenna Whiteboard in the Air," in *Proceedings of the 12th International on Conference on emerging Networking EXperiments and Technologies*. ACM, 2016, pp. 443–456.
- [8] Z. Ren, J. Yuan, J. Meng, and Z. Zhang, "Robust part-based hand gesture recognition using Kinect sensor," *IEEE Transactions on Multimedia*, vol. 15, no. 5, pp. 1110–1120, 2013.
- [9] K. K. Biswas and S. K. Basu, "Gesture recognition using Microsoft Kinect," in *Automation, Robotics and Applications (ICARA), 2011 5th International Conference on*. IEEE, 2011, pp. 100–103.
- [10] RealSense Depth Camera. <https://www.intelrealsense.com/stereo-depth>. Accessed: 2020-03-15.
- [11] Leap Motion Controller. <https://www.ultraleap.com/>. Accessed: 2020-03-15.
- [12] A. Kumar and J. McNames, "Wideband acoustic positioning with precision calibration and joint parameter estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 66, no. 8, pp. 1946–1953, 2017.
- [13] M. M. Saad, C. J. Bleakley, T. Ballal, and S. Dobson, "High-accuracy reference-free ultrasonic location estimation," *IEEE Transactions on Instrumentation and Measurement*, vol. 61, no. 6, pp. 1561–1570, 2012.
- [14] S. Gupta, D. Morris, S. Patel, and D. Tan, "Soundwave: using the doppler effect to sense gestures," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM, 2012, pp. 1911–1914.
- [15] Y. Qifan, T. Hao, Z. Xuebing, L. Yin, and Z. Sanfeng, "Dolphin: Ultrasonic-based gesture recognition on smartphone platform," in *Computational Science and Engineering (CSE), 2014 IEEE 17th International Conference on*. IEEE, 2014, pp. 1461–1468.
- [16] H. Abdelnasser, M. Youssef, and K. A. Harras, "Wigest: A ubiquitous wifi-based gesture recognition system," in *Computer Communications (INFOCOM), 2015 IEEE Conference on*. IEEE, 2015, pp. 1472–1480.
- [17] Z. Fu, J. Xu, Z. Zhu, A. X. Liu, and X. Sun, "Writing in the air with WiFi signals for virtual reality devices," *IEEE Transactions on Mobile Computing*, vol. 18, no. 2, pp. 473–484, 2018.
- [18] A. Makki, A. Siddig, M. Saad, J. R. Cavallaro, and C. J. Bleakley, "Indoor localization using 802.11 time differences of arrival," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 3, pp. 614–623, 2016.
- [19] R.-H. Liang and M. Ouhyoung, "A real-time continuous gesture recognition system for sign language," in *Automatic Face and Gesture Recognition, 1998. Proceedings. Third IEEE International Conference on*. IEEE, 1998, pp. 558–567.
- [20] Noitom Motion Capture System. <https://www.noitom.com>. Accessed: 2020-03-15.
- [21] HTC VIVE. <https://www.vive.com/eu/product>. Accessed: 2020-03-15.
- [22] S. Mitra and T. Acharya, "Gesture recognition: A survey," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 3, pp. 311–324, 2007.
- [23] T. Ballal, *Angle-of-arrival Based Ultrasonic 3-D Location for Ubiquitous Computing*. Ph. D. Thesis, University College Dublin, 2010.
- [24] M. M. Saad, C. J. Bleakley, and S. Dobson, "Robust high-accuracy ultrasonic range measurement system," *IEEE Transactions on Instrumentation and Measurement*, vol. 60, no. 10, pp. 3334–3341, 2011.
- [25] A. Lindo, E. Garcia, J. Ureña, M. del Carmen Perez, and A. Hernandez, "Multiband waveform design for an ultrasonic indoor positioning system," *IEEE Sensors Journal*, vol. 15, no. 12, pp. 7190–7199, 2015.
- [26] T. Ballal and C. J. Bleakley, "3D location and orientation estimation using Angle of Arrival," in *2009 IEEE International Symposium on Intelligent Signal Processing*. IEEE, 2009, pp. 21–26.
- [27] H. Krim and M. Viberg, "Two decades of array signal processing research: the parametric approach," *IEEE Signal Processing Magazine*, vol. 13, no. 4, pp. 67–94, 1996.
- [28] V. Krishnaveni, T. Kesavamurthy, and B. Aparna, "Beamforming for direction-of-arrival (DOA) estimation—a survey," *International Journal of Computer Applications*, vol. 61, no. 11, 2013.
- [29] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.
- [30] B. D. Van Veen and K. M. Buckley, "Beamforming: A versatile approach to spatial filtering," *IEEE ASSP Magazine*, vol. 5, no. 2, pp. 4–24, 1988.
- [31] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.
- [32] J. A. Cadzow, "Multiple source location—the signal subspace approach," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 38, no. 7, pp. 1110–1125, 1990.
- [33] M. Li and Y. Lu, "Angle-of-arrival estimation for localization and communication in wireless networks," in *Signal Processing Conference, 2008 16th European*. IEEE, 2008, pp. 1–5.
- [34] A. Quazi, "An overview on the time delay estimate in active and passive systems for target localization," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 29, no. 3, pp. 527–533, 1981.
- [35] A. Barabell, "Improving the resolution performance of eigenstructure-based direction-finding algorithms," in *Acoustics, Speech, and Signal Processing, IEEE International Conference on ICASSP'83*, vol. 8. Citeseer, 1983, pp. 336–339.
- [36] A. Paulraj, R. Roy, and T. Kailath, "Estimation of signal parameters via rotational invariance techniques-ESPRIT," in *Circuits, Systems and Computers, 1985. Nineteenth Asilomar Conference on*. IEEE, 1985, pp. 83–89.
- [37] R. Roy and T. Kailath, "ESPRIT-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.
- [38] T. Ballal and C. J. Bleakley, "DOA estimation for a multi-frequency signal using widely-spaced sensors," in *Signal Processing Conference, 2010 18th European*. IEEE, 2010, pp. 691–695.
- [39] —, "Phase-difference ambiguity resolution for a single-frequency signal," *IEEE Signal Processing Letters*, vol. 15, pp. 853–856, 2008.
- [40] C. Hekimian-Williams, B. Grant, X. Liu, Z. Zhang, and P. Kumar, "Accurate localization of RFID tags using phase difference," in *RFID, 2010 IEEE International Conference on*. IEEE, 2010, pp. 89–96.
- [41] J.-F. Gu, W.-P. Zhu, and M. Swamy, "Joint 2-D DOA estimation via sparse l-shaped array," *IEEE Transactions on Signal Processing*, vol. 63, no. 5, pp. 1171–1182, 2015.
- [42] H. Chen, T. Ballal, M. Saad, and T. Y. Al-Naffouri, "Angle-of-arrival-based gesture recognition using ultrasonic multi-frequency signals," in *Signal Processing Conference (EUSIPCO), 2017 25th European*. IEEE, 2017, pp. 16–20.
- [43] L. R. Rabiner and B.-H. Juang, *Fundamentals of speech recognition*. PTR Prentice Hall Englewood Cliffs, 1993, vol. 14.
- [44] P. Guillaume and C. Ana-Maria, "Static and dynamic hand gesture recognition in depth data using dynamic time warping," *IEEE Transactions on Instrumentation and Measurement*, vol. 65, no. 2, pp. 305–316, 2015.
- [45] E. Stergiopoulou and N. Papamarkos, "Hand gesture recognition using a neural network shape fitting technique," *Engineering Applications of Artificial Intelligence*, vol. 22, no. 8, pp. 1141–1158, 2009.
- [46] H. Chen, T. Ballal, X. Liu, and T. Y. Al-Naffouri, "Realtime 2-D DOA Estimation using Phase-Difference Projection (PDP)," in *2019 27th European Signal Processing Conference (EUSIPCO)*. IEEE, 2019, pp. 1–5.
- [47] ICASSP 2018 demo session. <https://2018.ieeeicassp.org/Demos.asp>. Accessed: 2020-03-15.
- [48] Demo Video. <https://www.youtube.com/watch?v=XRi2iezsG4Q&t=2s>. Accessed: 2020-03-15.
- [49] J. R. Gonzalez and C. J. Bleakley, "High-precision robust broadband ultrasonic location and orientation estimation," *IEEE Journal of selected topics in Signal Processing*, vol. 3, no. 5, pp. 832–844, 2009.
- [50] T. Ballal and C. J. Bleakley, "DOA estimation of multiple sparse sources using three widely-spaced sensors," in *Signal Processing Conference, 2009 17th European*. IEEE, 2009, pp. 1978–1982.

- [51] —, “Phase-difference ambiguity resolution for a single-frequency signal in the near-field using a receiver triplet,” *IEEE Transactions on Signal Processing*, vol. 58, no. 11, pp. 5920–5926, 2010.
- [52] S. R. Siddharth and A. Anupam, “Vision based hand gesture recognition for human computer interaction: a survey,” *Artificial Intelligence Review*, vol. 43, no. 1, pp. 1–54, 2014.
- [53] Matlab Deep Learning Toolbox-Examples. <https://www.mathworks.com/help/deeplearning/examples.html>. Accessed: 2020-03-15.
- [54] L. R. Rabiner, “A tutorial on hidden Markov models and selected applications in speech recognition,” in *Readings in speech recognition*. Elsevier, 1990, pp. 267–296.
- [55] PioneerTS-T110 Tweeter. <https://www.pioneerelectronics.com/PUSA/Car/Speakers/Special-Application/TS-T110>. Accessed: 2020-03-15.
- [56] SPH0611LR5H Microphone. <https://www.mouser.com/ds/2/218/SPH0611LR5H-1-1173858.pdf>. Accessed: 2020-03-15.
- [57] ARTTRACK5 (advanced realtime tracking) technical data. <https://ar-tracking.com/products/tracking-systems/arttrack5>. Accessed: 2020-03-15.