IEEE Transactions on Multimedia

An FCNN-Based Super-Resolution mmWave Radar Framework for Contactless Musical Instrument Interface

Journal:	*IEEE Transactions on Multimedia*
Manuscript ID	MM-011628.R2
Suggested Category:	Regular Paper
Date Submitted by the Author:	n/a
Complete List of Authors:	Smith, Josiah; The University of Texas at Dallas, Electrical Computer Engineering Furxhi, Orges; IMEC, Computational Imaging Torlak, Murat; Dept. of Electrical Engineering, The University of Texas at Dallas
EDICS:	3-MHMI Multimodal Human-machine Interfaces and Interaction < 3 HUMAN CENTRIC MULTIMEDIA, 9-MWIM Multimedia for personal applications (mobile, wearables, interactive) < 9 EMERGING TOPICS IN MULTIMEDIA, 9-ARVL Multimedia for Augmented Experience in Real and Virtual Life < 9 EMERGING TOPICS IN MULTIMEDIA, 9-SLAM Speech, Language, and Audio in Video Analysis; Music in Multimedia < 9 EMERGING TOPICS IN MULTIMEDIA, 9-DLMP Deep Learning for Multimedia Processing < 9 EMERGING TOPICS IN MULTIMEDIA

Note: The following files were submitted by the author for peer review, but cannot be converted to PDF. You must view these files (e.g. movies) online.

Simple Tracking.mp4
FCNN-DPF Tracking.mp4
Doppler Tracking.mp4
Music Example.mp4
Cross-Range Oscillation Tracking.mp4

An FCNN-Based Super-Resolution mmWave Radar Framework for Contactless Musical Instrument Interface

Josiah W. Smith[1], Orges Furxhi[2], and Murat Torlak[1]

[1]Dept. of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX, United States
[2]Camera Systems and Computational Imaging, Imec, Kissimmee, FL, United States

***Abstract*—In this paper, we propose a framework for contactless human-computer interaction (HCI) using novel tracking techniques based on deep learning-based super-resolution and tracking algorithms. Our system offers unprecedented high-resolution tracking of hand position and motion characteristics by leveraging spatial and temporal features embedded in the reflected radar waveform. Rather than classifying sample from a predefined a set of hand gestures, as common in existing work on deep learning with mmWave radar, our proposed imager employs a regressive full convolutional neural network (FCNN) approach to achieve spatial super-resolution improving localization. While the proposed techniques are suitable for a host of tracking applications, this article focuses on their application as a musical interface to demonstrate the robustness of the gesture sensing pipeline and deep learning signal processing chain. The user can control the instrument by varying the position and velocity of their hand above the vertically-facing sensor. By employing a commercially-available multiple-input-multiple-output (MIMO) radar rather than a traditional optical sensor, our framework demonstrates the efficacy of the mmWave sensing modality for fine motion tracking and offers an elegant solution to a host of HCI tasks. Additionally, we provide a freely-available software package and user interface for controlling the device, streaming the data to MATLAB in real-time, and increasing accessibility to the signal processing and device interface functionality utilized in this article.**

***Index Terms*—deep learning, human-computer interaction (HCI), fully-convolutional neural network (FCNN), millimeter-wave (mmWave), multiple-input multiple-output (MIMO), radar perception, super-resolution**

I. Introduction

Radar perception for human-computer interaction (HCI) on multiple-input-multiple-output (MIMO) millimeter-wave (mmWave) radars has emerged as a promising solution to a variety of sensing problems. The physical nature of millimeter-waves offers a safe method for high-resolution imaging where optical sensors may fail due to insufficient lighting, fog, or other line-of-sight interference. Additionally, mmWave sensors are considered less-invasive than optical counterparts and promote user privacy. Ultra-wideband MIMO devices enable centimeter-level spatial resolution with a small profile device. As a result, precise spatial information of a target scene can be easily acquired from such imaging devices at a low cost.

mmWave sensors are relatively modern technology, but some of the earliest electronic interfaces were contactless devices for physically expressive musical control including the Radio Drum and Theremin [1]. Russian physicist Leon Theremin demonstrated his noncontact musical instrument in 1921, an interface controlled by the proximity of the musician's hand to an antenna using beat-frequency oscillators and a capacitive sensing apparatus [2]. More recently, computer vision approaches have been adopted for the innovation of contactless new musical interfaces (NMIs), most of which rely on optical camera solutions. Extensive prior work exists on optical-based NMIs using popular sensors such as the Microsoft Kinect and Leap Motion.

In [3], Polfreman uses the Kinect to track the 3-D position of both hands of a standing performer to construct a multi-modal instrument. Trail *et al.* present a pitched percussion hyper-instrument to track the tips of two mallets simultaneously with the Kinect [4]. Crossole, designed by Senturk *et al.*, is a Kinect-based metainstrument that visualizes chord progressions as virtual blocks resembling a crossword puzzle [5]. Schramm *et al.* use the Kinect to analyze and classify motions of a orchestral conductor [6]. In [7], the Kinect is used to track hand motion across time and then translated to music using the inverse Fourier transform of the physical pattern using a sonification technique called sonomotiongram.

Alternatively, the popular Leap Motion controller is capable of modeling the entire hand, including the fingers, which allows for even more detailed hand posture-based gesture control to be explored for musical interface development. Using the Leap Motion sensor, Han *et al.* developed two NMIs, *Air Keys* and *Air Pad*. *Air Keys* tracks the motion and position of each finger to recognize when and which keys the musician is pressing and playing the desired notes. Similarly, *Air Pad* tracks the hand position to create a 2-D virtual drum pad played by pressing specific regions in a 2-D horizontal plane, thus requiring accurate 3-D hand-tracking [8]. Hantrakul and Kaczmarek use the Leap Motion controller to track both hands for controlling MIDI (Musical Instrument Digital Interface) instruments and virtual effects [9]. Similarly, Leimu pairs the Leap Motion with an inertial measurement unit (IMU) demonstrating improved performance over the Leap Motion controller alone for musical interface [10]. Other solutions have been attempted, such as employing non-invasive force sensing resistors to enhance "traditional" instruments by learning and monitoring for gestures performed by the musician [11].

In the optical HCI domain, [12] proposes a musical interface using only a portable RGB camera to recognize hand gestures using a gesture classification technique. Akbari and Cheng developed a system to transcribe music played on a piano in real-time using optical cameras positioned to view the keys [13]. These projects have yielded high-performing real-time musical interfaces capable of consistent high-accuracy motion tracking but require several key design constraints, namely specific lighting conditions and line-of-sight. As shown in this article, mmWave sensors overcome these major obstacles while providing superior privacy through the means of advanced spatiotemporal algorithms. However, little work has been done towards gestural musical interfaces on mmWave radar sensors using hand-tracking techniques. Even though extensive research exists on static and dynamic gesture recognition using deep learning models and mmWave radars [14], [15], Google ATAP's Project Soli is the only effort using mmWave radar for musical interface, using gesture recognition and 1-D position estimation to control the parameters of audio synthesizers [16].

The novel framework presented in this article offers a major advancement for near-field mmWave hand-tracking and an accessible MATLAB software platform for further investigation into real-time mmWave HCI and algorithm innovation. 2-D localization performance is considerably improved from past work [17] by employing a novel deep learning-based technique to improve the resolution beyond the theoretical limitations.

It is important to note our approach is contrary to gesture classification, wherein the objective is to determine the class of a sample from a set of predefined classes as in [14], [15]; rather, we apply a novel fully convolutional neural network (FCNN) to preserve the geometry of the image and perform super-resolution for improved localization. Prior work on resolution improvement using FCNNs has been limited to the far-field domain with large apertures [18]; however, our novel approach unifies FCNN-based super-resolution with near-field imaging on a small (8-channel) array and is shown to improve hand-tracking performance significantly. Additionally, a particle filter tracking algorithm is presented to further improve tracking robustness by employing the Doppler effect. Compared to prior work on gesture tracking using optical solutions [3], [7]–[9], [12], [19], our approach offers fine hand-tracking using a single mmWave sensor offering higher depth resolution with superior privacy. This article proposes a novel hand-tracking method for musical interface by fusing spatiotemporal algorithms, deep learning-enhanced feature extraction, and robust position tracking algorithms. To aid further development and prototyping for real-time mmWave gesture applications, the entire software implementation is available by request to the corresponding author. To our knowledge, this proposed framework is the first openly available software package supporting real-time data streaming from a mmWave radar into MATLAB for streamlined signal processing and deep learning algorithm development.

The rest of this paper is formatted as follows. Section II provides an overview of the frequency modulated continuous wave (FMCW) radar signal model and feature extraction methods. In Section III, two robust tracking algorithms and estimation techniques are presented. The system implementation is discussed in Section IV and results are shown in Section V. Section VI provides a discussion of the performance, design constraints, and distinct advantages of the two tracking methods in Sections III-A and III-B, followed finally by conclusions.

Notation: Throughout this paper, vectors and matrices are set in boldface, using lowercase letters for vectors and uppercase letters for matrices. The superscripts ^{T} and * denote the transpose and conjugation operations, respectively. The identity matrix of size $N \times N$ is expressed as $\boldsymbol{I}_N$ and $\mathbf{1}_N$ is the all-ones vector of size $N \times 1$. Spatial coordinates are treated as continuous to support continuously distributed target scenes and all time variables are modeled in discrete-time.

II. Preliminaries of MIMO-FMCW Radar Signaling

In this section, we overview the propagation model for the FMCW radar chirp signal and examine the spatiotemporal features of a target in motion. The imaging scenario, as shown in Fig. 1, consists of a multistatic linear MIMO array facing vertically. Orthogonality is leveraged in time by employing time-division-multiplexing MIMO (TDM-MIMO), wherein the transmitters are activated at separate time instances. Throughout this paper, the musician's hand is modeled as a point reflector located at the point (y, z), an assumption that holds given the physical limitations of the device and scenario examined in this article and has been verified empirically.

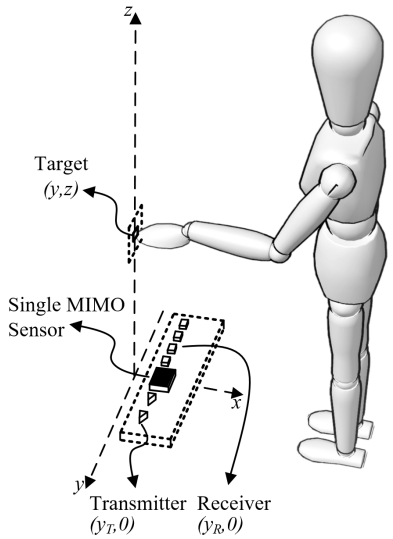


Fig. 1. The imaging geometry, where the linear MIMO array faces vertically and the musician moves their hand throughout the y-z plane.

A. MIMO-FMCW Signal Model

The FMCW chirp signal model is well documented in literature [20]–[22] and is discussed in this section for reference and continuity throughout this paper. Considering a single transmitter/receiver pair located at $(y_T, 0)$ and $(y_R, 0)$ in the y-z plane, respectively, and an ideal point target with reflectivity

p located at (y, z), the time sampled FMCW beat signal can be expressed in discrete-time as

$$s(y_T, y_R, n_k) = \frac{p}{R_T R_R} e^{j(k_0+\Delta n_k)(R_T+R_R)}, \quad (1)$$

where R_T, R_R are the distances from the transmitter and receiver to the point target, respectively, n_k is the wavenumber index, $k_0 = 2\pi f_0/c$ is the starting wavenumber corresponding to the starting frequency f_0, and $\Delta = 2\pi K/(cf_S)$ is the wavenumber step size with K being the chirp slope, f_S being the sampling frequency, and c is the speed of light.

To ease the subsequent signal processing, it is desirable to approximate the multistatic MIMO beat signal, represented in (1) as its corresponding monostatic equivalent using the approximation developed in [23] as

$$\hat{s}(y', n_k) = s(y_T, y_R, n_k) e^{-j(k_0+\Delta n_k)\frac{d_y^2}{4z_0}}, \quad (2)$$

valid only for small d_y, the distance between the transmitter and receiver elements, where z_0 is a reference plane typically given as the center of the target scene. Taking y' as the locations of the virtual elements located at the midpoints between each transceiver pair and R as the corresponding distance from each virtual element to the point reflector, the resulting monostatic beat signal approximates to

$$\hat{s}(y', n_k) \approx \frac{p}{R^2} e^{j2(k_0+\Delta n_k)R}. \quad (3)$$

From (3), the spatial location, (y, z), of the target is embedded in the radar beat signal, in the form of the radial distance R.

B. Doppler Radar Signal Processing

The relative velocity of a target can be extracted from the beat signal expressed in (3) by exploiting the Doppler effect. As discussed in [24], by transmitting a series of chirp waveforms at a known pulse repetition interval (PRI), T_{PRI}, the velocity of a moving target can be identified as the frequency component along the chirp index dimension given by

$$\hat{s}(y', n_k, n_c) = \frac{p}{R^2} e^{j(2(k_0+\Delta n_k)R + \frac{4\pi v T_{PRI}}{\lambda_0} n_c)}, \quad (4)$$

where R is the initial range of the target, v is the velocity of the target, λ_0 is the wavelength corresponding to f_0, and n_c is the chirp index,

Thus, the beat signal sampled across time is a 2-D complex sinusoidal with frequencies corresponding to the range and velocity of the target on the first and second dimensions, respectively. Subsequently, to extract the range and velocity, traditional methods perform a 2-D fast Fourier transform (FFT) over a matrix whose rows or columns consist of subsequent chirps.

C. Range Migration Algorithm Image Reconstruction

To achieve high-fidelity 2-D localization, we employ the range migration algorithm (RMA) over traditional range-angle FFT methods [20], whose localization accuracy is known to be inferior [25]. The primary goal of the RMA is to reconstruct the target scene's reflectivity function, $p(y, z)$. For a distributed target, the beat signal can be modeled as the superposition of the backscattered signal at every point in the scene, neglecting the amplitude terms, as

$$\hat{s}(y', n_k) = \iint p(y, z) e^{j2(k_0+\Delta n_k)R} dy dz, \quad (5)$$

This target model assumes a spatially distributed target whose reflectivity only depends on spatial location and neglects the any frequency dependence of the reflectivity function. Inverting (5) using the method of stationary phase, the reflectivity function, $p(y, z)$, can be estimated efficiently by

$$\hat{p}(y, z) = \text{IFT}_{\text{2D}}^{(k_y, k_z)} \left[\mathcal{S} \left[\text{IFT}_{\text{1D}}^{(y')} [\hat{s}^*(y', n_k)] \right] \right], \quad (6)$$

where $\mathcal{S}[\bullet]$ is the Stolt interpolation operation [23] and FT[$\bullet$], IFT[$\bullet$] are the forward and inverse Fourier transform operators. To avoid aliasing in the image sampling criteria must be considered [22]. Spatial resolution along the y and z directions are constrained by the physical and device limitations and are expressed as

$$\delta_y = \frac{\lambda_c z_0}{2D_y}, \quad (7)$$

$$\delta_z = \frac{c}{2B}, \quad (8)$$

where λ_c is the wavelength corresponding to the frequency at the center of the chirp sweep, D_y is the aperture size along the y direction, and z_0 is the center of the imaging scene [22].

After the 2-D reflectivity function of the target scene is recovered, the hand position is estimated subsequently as

$$\{\hat{y}, \hat{z}\} = \arg\max_{\{y,z\}} \hat{p}(y, z). \quad (9)$$

Further, the aforementioned Doppler principle can be leveraged to extract the velocity of the target by Fourier analysis over successive chirps. To optimally exploit the deep learning framework discussed in Section III-B2 and reduce the required computation complexity, the velocity is extracted after the RMA is performed and hand location is estimated.

As evident in (4), the velocity is decoupled from the wavenumber index and is the scaled frequency component along the chirp index dimension. As a result, the phase term corresponding to the velocity is preserved in the reconstructed image, $\hat{p}(y, z)$. Therefore, the velocity profile can be obtained by performing an FFT across the chirp index, n_c, dimension of the recent images. Rather than performing the FFT across the 3-D array, $\hat{p}(y, z, n_c)$, we perform the FFT over the slice of the image corresponding to the estimated position, $\hat{y}$, yielding the velocity profile along the z-direction, where n_d is the velocity index, as

$$\hat{d}(z, n_d) = \text{FFT}_{\text{1D}}^{(n_c)} \left[\hat{p}(y, z, n_c) \Big|_{y=\hat{y}} \right]. \quad (10)$$

Finally, the velocity can be estimated from (10) using video pulse integration by

$$\hat{v}_d = \arg\max \sqrt{\int |\tilde{d}(z, n_d)|^2 dz}. \quad (11)$$

The velocity computed by this method is referred to as the Doppler velocity. The recovered velocity using this approach is limited by the timing and physical constraints between $[-\frac{\lambda_0}{4T_{PRI}}, \frac{\lambda_0}{4T_{PRI}}]$. Later, the Doppler velocity is employed to improve the tracking performance using the Doppler corroborated particle filter.

III. Spatiotemporal Imaging on mmWave Radar

In this section, we present the methods for our proposed imager capable of high accuracy hand-tracking for HCI. The contribution of this article is the advancement in algorithm performance for 2-D localization by utilizing both the novel super-resolution FCNN and proposed tracking algorithm. While we will investigate the application of such algorithms as an NMI, our mmWave radar-based sensing algorithms can be applied to a host of HCI problems.

It is important to note that this work is not intended to compete with the computational efficiency of embedded HCI solutions and existing musical interfaces. Rather, the main contributions of this article are novel algorithms for super-resolution spatiotemporal hand-tracking and a freely-downloadable platform to increase accessibility and encourage further research in this arena. As such, we will focus primarily on the development of the algorithms and their localization performance. Discussions on performance and implementation issues are considered secondary and are addressed in Sections IV and VI.

A. Classical Spatiotemporal Feature Extraction Techniques

In this section, we introduce the simple approach to spatiotemporal sensing for contactless musical instrument interface. While our system generally tracks the 2-D position and velocity of the user's hand, we have identified three underlying features to achieve fine control of the musical interface: range, cross-range oscillation, and velocity. By the geometry given in Fig. 1, we define the range as the position of the hand along the z-axis, i.e. the vertical displacement between the sensor and the user's hand. Similarly, cross-range is defined as the position of the hand along the y-axis. Subsequently, cross-range oscillation is the rate at which the hand oscillates in the cross-range direction. Velocity is given by the velocity of the hand with respect to the range z-axis. These parameters are selected such that the output musical interface is controlled primarily by the range of the musician's hand and secondarily by the cross-range oscillation and velocity. However, these parameters can be assigned by the user based on preference using the MIDI interface, as discussed later. Throughout the remainder of this article, we will refer to these parameters as features extracted from the radar beat signal.

Under the simple gesture tracking regime, the 2-D location and velocity $(\hat{y}, \hat{z}, \hat{v}_d)$ are extracted from the reconstructed image and buffer of recent images using (9) and (11). In the next section, the three parameters extracted from the raw data are treated as a vector called the noisy measurement vector $\boldsymbol{r}$. In the optimal scenario, the bandwidth, antenna array size, and the signal-to-noise-ratio (SNR) are quite large, tending towards infinity. For the case of an 8 channel automotive mmWave radar and a human hand, the bandwidth is limited (4 GHz), the antenna array size is small ($D_y = 2\lambda_c$), and the reflectivity of the hand is not high compared to the noise level. As a result, simply extracting the maximum from the reconstructed RMA images yields sporadic location and velocity estimates. Even in the ideal case, the spatial resolution of our system along the y and z directions is $\delta_y = 7.5$ cm and $\delta_z = 3.75$ cm, respectively. Several other factors are not taken into account in the classical, direct tracking method including beam-pattern, residual phase errors, and antenna coupling. All these limitations and non-idealities in the imaging scenario degrade the image and result in noisy location and velocity estimates; however, many of these issues analytical forms and cannot be solved directly classical methods. To address these issues, we present a novel data-driven approach employing an FCNN for super-resolution and image enhancement.

B. FCNN-Based Super-Resolution Feature Extraction and Particle Filter Tracking Methods

In this section, we improve upon the simple tracking techniques to overcome noise and foundational non-idealities in the imaging scenario, yielding a much-improved user experience. The concepts demonstrated in this section are applicable for many tracking and high-resolution imaging applications beyond the scope of musical interfaces.

To improve the tracking robustness of the proposed musical interface, we adopt the well-known particle filter [26] and present a novel modification. While traditional methods such as the extended Kalman filter (EKF) employ a motion model, our implementation of the particle filter bypasses the need for a deterministic motion model. The particle filter is selected for this application as other traditional approaches have demonstrated poor tracking performance in our experimentation, yielding either sporadic localization or overly damped, sluggish estimation. Additionally, the particle filter is advantageous as it can track non-linear dynamics and does not require prior knowledge of the motion model or noise parameters for robust localization.

In our modification of the particle filter, the control input is a weighted movement towards the newest measurement. To demonstrate our proposed algorithm, consider the case of simultaneous location estimation along the y and z directions. The new noisy measurement vector, $\boldsymbol{r}$, has two elements, the newest estimates of location, $\hat{y}$ and $\hat{z}$, which are extracted by the methods described in the prior section. Algorithm 1 details the modified particle filter implementation. For 2-D localization, $\boldsymbol{X}_n$ is a matrix of size $N \times 2$, whose rows are the (y, z) coordinates of each particle at time index n, where N is the number of particles, and $\boldsymbol{w}_n$ is the vector of weights corresponding to each particle. The estimates of the 2-D location (also known as the estimated states) form the vector $\boldsymbol{s}_n$ and the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is denoted as $G(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Before executing the iterative algorithm, the initial particle states matrix, $\boldsymbol{X}_0$, and initial weights vector, $\boldsymbol{w}_0$, are initialized with as random locations throughout the region of interest (ROI) and uniform weights, respectively.

Algorithm 1: Modified Particle Filter Algorithm

input : $\boldsymbol{r} = [\hat{y}, \hat{z}]^T$
output: $\boldsymbol{s}_n = [\tilde{y}, \tilde{z}]^T$
1 $\boldsymbol{X}_n \leftarrow$ rows of $\boldsymbol{X}_{n-1}$ sampled using weights $\boldsymbol{w}_{n-1}$;
2 $\boldsymbol{X}_n \leftarrow \boldsymbol{X}_n + \mathbf{1}_N \boldsymbol{a}^T (\boldsymbol{r} - \boldsymbol{s}_{n-1}) + \boldsymbol{\psi}$;
3 $\boldsymbol{w}_n \leftarrow e^{-\frac{1}{2}(\boldsymbol{X}_n - \boldsymbol{s}_{n-1})^T \boldsymbol{\Sigma}_w^{-1} (\boldsymbol{X}_n - \boldsymbol{s}_{n-1})}$;
4 $\boldsymbol{s}_n \leftarrow \frac{1}{\mathbf{1}_N^T \boldsymbol{w}_n} \boldsymbol{X}_n^T \boldsymbol{w}_n$;

Proper handling of the key steps, (step 2) resampling of the particle states and (step 3) computing new weights, is essential to effectively implement our novel particle filter algorithm.

The particle resampling process involves moving the particles towards the new measurement by a specified weight. $\boldsymbol{a} = [a_y, a_z]^T$ is a vector whose two elements provide weight to the noisy estimates $\hat{y}$ and $\hat{z}$, respectively. The size of $\boldsymbol{a}$, $\boldsymbol{r}$, and $\boldsymbol{s}_n$ can be varied depending on the number of parameters to be tracked by the particle filter. Hence, the new measurements do not dominate the motion tracking but have a weighted influence on the localization procedure. Fig. 2 demonstrates the resampling process with $a_y = a_z = 0.5$. Note that before computing the new weights, particle diffusion is performed by adding the perturbation term $\boldsymbol{\psi}$. The random vector $\boldsymbol{\psi}$ is Gaussian distributed with zero mean and predefined covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}$.

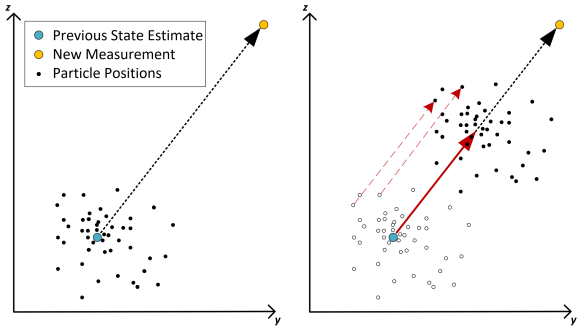


Fig. 2. A visual example of the modified particle filter algorithm resampling process. The particle locations are resampled by a shift transformation towards the new measurement according to the weight vector $\boldsymbol{a}$, where $a_y = a_z = 0.5$.

The new weights are computed from a multivariate Gaussian distribution with the previously estimated states, $\boldsymbol{s}_{n-1}$, as the mean vector and a predefined covariance matrix $\boldsymbol{\Sigma}_w$. Therefore, particles closer to the previously estimated state are assigned a higher weight than those farther away. This results in a tendency towards small changes in the state estimations while monitoring for movement from the current position. For many applications requiring precise and consistent localization and motion tracking on mmWave radar, our modified particle filter algorithm is an ideal fit as it tends to a steady-state estimation of the states but remains active in monitoring the noisy sensor input.

1) Doppler-Corroborated Real-Time Weighting: In this section, we present a dynamic weighting technique for updating $\boldsymbol{a}$ in real-time by exploiting the dependence between position and velocity. Our approach considers corroboration between the Doppler velocity estimate and the velocity estimated from the range samples as a measure of the new measurement's reliability. Thus, the dependability of the Doppler velocity can improve tracking of the target position along the range (z) dimension even in the presence of noisy position estimates. After the Doppler velocity is calculated by (11), the recent range estimates are used to calculate the sample velocity ($\hat{v}_s$) by the least squares estimator as

$$\hat{v}_s = \frac{N_z T_{PRI} \sum_m (\boldsymbol{z}^{(m)} m) - T_{PRI} \sum_m \boldsymbol{z}^{(m)} \sum_m m}{N_z \sum_m (\boldsymbol{z}^{(m)})^2 - \left(\sum_m \boldsymbol{z}^{(m)}\right)^2}, \tag{12}$$

where $\boldsymbol{z}^{(m)}$ is the m^{th} element of the vector of recent $\hat{z}$ estimates, $\boldsymbol{z}$, with $\boldsymbol{z}^{(N_z-1)}$ being the most recent.

The difference between the Doppler estimated velocity and sample estimated velocity is computed as $\Delta_v = |\hat{v}_d - \hat{v}_s|$ and used in the reward function (13) to update the weight placed on the new noisy measurement in real-time.

$$a_z(\Delta_v) = \begin{cases} a_{z,0} \cos\left(\frac{2\pi T_{PRI} \Delta_v}{\lambda_0}\right) & \text{if } \Delta_v \leq \frac{\lambda_0}{4T_{PRI}} \\ 0 & \text{if } \Delta_v > \frac{\lambda_0}{4T_{PRI}} \end{cases} \tag{13}$$

When the sample velocity is close to Doppler velocity, i.e. Δ_v is small, the reward function is close to $a_{z,0}$. Hence, the new measurement is corroborated by the reliable Doppler velocity and weighted accordingly. Outliers and erroneous measurements contradicting the Doppler velocity are given less importance during the particle resampling process. To implement the Doppler corroborated particle filter, $\boldsymbol{a} = [a_y, a_z(\Delta_v)]^T$ is dynamically updated by (13) at each iteration of Algorithm 1.

2) Improved 2-D Position Estimation by Enhancing FCNN: The modified particle filter algorithm improves the tracking consistency and smoothness; however, several issues such as instrumentation delay, ambient/device noise, multistatic effects, and non-spherical beam patterns remain unaddressed and degrade tracking performance. To overcome these non-idealities, we present a novel FCNN-based technique for image enhancement that improves the 2-D position estimation, subsequent tracking accuracy, and Doppler spectrum SNR. Compared to prior FCNN synthetic aperture radar (SAR) techniques employing far-field assumptions and trained on synthetically generated data [18], our enhancement FCNN method operates on near-field images, improves localization even with a small aperture, and is trained using a novel technique allowing the network to learn the environment and device noise, near-field beam pattern, and multistatic effects.

To train the enhancement FCNN, we construct a dataset consisting of both real human hand data and synthetically generated data. Real hand data are collected by capturing frames while the user holds their hand at known locations relative to the device and synthetic data are used to supplement the training set. Each synthetic sample is generated by simulating a MIMO beat signal using (5) with one ideal point target located at a known location and additive real device noise, collected from the radar. The simulated locations are randomized to uniformly cover the ROI. Both the real and synthetic data are

used as features in the FCNN training process, thus enabling the network to fit the non-ideal beam pattern, real multipath and multistatic effects, empirical reflection of a human hand, device and ambient noise, and hand positions throughout the ROI.

To train the image-to-image regression FCNN, each training feature (real or synthetic image) must correspond to a ground truth label. The ground truth label images are synthetically generated by the model

$$\mathcal{I}(y,z) = e^{-(y-y_0)^2/\sigma_y^2-(z-z_0)^2/\sigma_z^2} \tag{14}$$

where the width of the expected target located at (y_0, z_0) is dictated by σ_y and σ_z in the y and z dimensions, respectively, yielding resolutions of $1.18\sigma_y$ and $1.18\sigma_z$ according to the 3 dB beamwidth definition [27]. Each label is generated using the requisite knowledge of the location of the human hand or target of each feature image. During training, the FCNN learns the highly nonlinear relationship between distorted, blurred RMA images and the ideal images generated using (14). Our novel training technique results in a robust and generalizable FCNN that improves image SNR and localization by fitting to the non-ideal imaging constraints. Further, the trained network enables localization precision beyond the physical limitations of the device improving tracking performance significantly. FCNN training is discussed in Section IV-D and results are presented and discussed in Section V-C.

Additionally, by isolating the peak corresponding to the human hand, clutter and phase noise at other positions are mitigated thereby improving the Doppler spectrum SNR and subsequent velocity estimation. Thus, the FCNN enhances both the spatial and temporal features extracted from the radar beat signal before the particle filter. Uniting the proposed particle filter and enhancement FCNN, the range, cross-range oscillation, and velocity are robustly tracked by our novel algorithms and mapped to musical interface controls.

IV. System Design and Implementation

In this section, we present the system implementation for both the classical tracking techniques and our novel super-resolution feature extraction and tracking algorithms discussed in the previous section.

A. Hardware and Software Implementation

The hardware employed in the proposed system consists of a Texas Instruments (TI) AWR1243 automotive radar in conjunction with a DCA1000EVM real-time data capture adapter. The TI radar is a MIMO-FMCW mmWave radar with an operating bandwidth of 4 GHz and a center frequency of 79 GHz. In this research, we utilize the linear MIMO array consisting of 2 transmit antenna (TX) elements, separated by $2\lambda_c$, and 4 receive antenna (RX) elements, separated by $\lambda_c/2$. The resulting virtual array has 8 equally spaced virtual elements separated by $\lambda_c/4$ [20]. The calibration methodology discussed in [23] is adopted to mitigate range bias, constant phase errors, and instrumentation delay. In this process, data are captured from a corner reflector at a known location and used to identify range bias and phase offsets among the antennas. Unlike an optical or infrared calibration, this process is invariant of lighting and temperature constraints as well as user hand sizes, etc. Thus, the one-time calibration applies to a variety of environments and users.

The software platform for signal processing, visualization, and machine learning is written in MATLAB. Despite its inferior computational efficiency compared to other languages, MATLAB is employed to provide an accessible platform for researchers to engage with this work and rapidly prototype custom real-time algorithms using our custom tools. Once the algorithms are validated on a PC, they can be implemented onto such embedded devices for optimized application-specific usage. For a positive user experience as a musical interface, latency and timing issues must be taken into account, and are discussed in Section VI.

B. Real-Time Data Retrieval and Interactive MATLAB User Interface

To stream the data from the device into MATLAB, a custom UDP interface software is written. This routine is implemented efficiently in C++ and is capable of receiving the sequential UDP packets, organizing the packets to form each chirp, and providing the data to MATLAB over shared memory.

A custom interactive MATLAB graphical user interface (GUI), shown in Fig. 3, is written to serve as the single user interface for our framework. The MATLAB GUI interfaces with TI mmWave Studio [28] to control the hardware setup and initializes the UDP interface, bypassing the need for user setup outside our GUI. The radar continuously captures and streams data into MATLAB using the fully-integrated implementation. While MATLAB does not offer the computational speed necessary for real-time system implementation, it is capable of completing the data capture, signal processing, deep learning, visualization, and signal output at around 250 Hz, from our experimentation. In the early prototyping phase, we consider this throughput sufficient for investigating the performance of the super-resolution tracking algorithms and a simple musical interface.

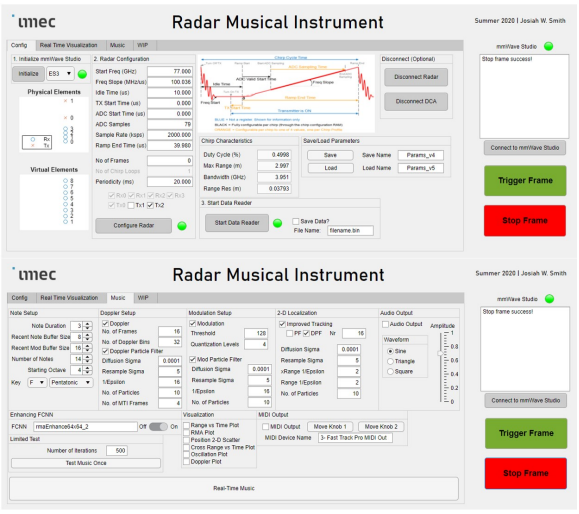


Fig. 3. Interactive MATLAB GUI: device setup and music generation pages.

Using our proposed methods, the software extracts high-resolution spatiotemporal features of the user's hand and maps them to corresponding output using either a built-in audio output tool or the included MIDI output. The custom MATLAB GUI provides an accessible option for investigating and demonstrating our methods as well as an open-source platform to stimulate further collaborative investigation by the multimedia and radar communities.

As previously mentioned, the primary mechanism to control the output of the proposed musical interface is the range (z-position) of the user's hand. Using the built-in audio output tool and the MIDI output, the range of the user's hand controls the note selection directly. Unlike the Theremin, which allows for continuous note selection, our interface quantizes the user input into predefined subregions corresponding to notes defined by the user. The subregions and allowed notes can be programmed by the user in the interactive MATLAB GUI. To play the desired note, the user must move their hand vertically to the position corresponding to that note. Similarly, the secondary parameters, cross-range oscillation, and velocity can be adjusted by the user by oscillating their hand back-and-forth in the y-direction or moving to the next note with a high or low velocity. The built-in audio output tool employs the cross-range oscillation to control a vibrato effect (low-frequency modulation of the audio signal). Thus, using this tool, the user can select the desired note by varying the range and perform vibrato at a desired rate by oscillating their hand at the same rate. Alternatively, the MIDI output tool provides the cross-range oscillation and velocity as MIDI parameters to be specified by the user in a virtual instrument environment connected to the MIDI output of our musical interface. Hence, our proposed algorithms are implemented to operate similarly to a MIDI keyboard with the hand range controlling the note selection and cross-range oscillation and velocity acting as MIDI parameters for the user to assign.

C. Simple Feature Extraction and Tracking Algorithm Signal Processing Chain

The signal processing chain for the simple feature extraction and tracking method is shown in Fig. 4. The beat signal is loaded into MATLAB where the preprocessing discussed in the previous section is performed (RMA and peak finding) and the user inputs (2-D location and velocity) are converted into audio or MIDI output by extracting the spatiotemporal features using (9) and (11). In this article, the location and velocity of the user's hand are used for musical gestural interface; however, our novel algorithms can easily be applied to many different HCI applications and even for 3-D localization, provided a sufficient 2-D array. The reconstructed RMA image and raw feature extracted by the classical techniques can be utilized by the particle filter algorithm and super-resolution FCNN to improve the tracking performance.

D. Super-Resolution Framework - Training FCNN and Implementing Particle Filter Algorithm

To implement our super-resolution feature extraction and tracking framework, the super-resolution FCNN must be first

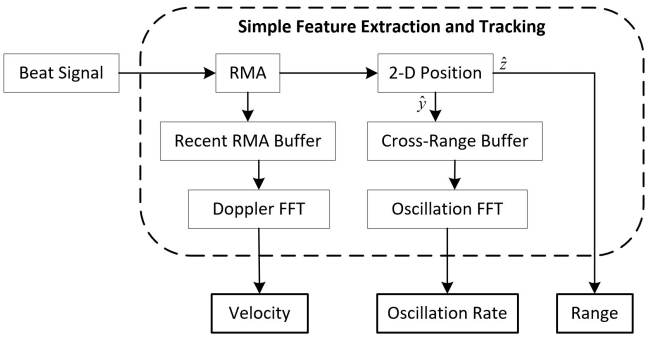


Fig. 4. Simple tracking signal processing chain. After RMA is performed on beat signal, features are extracted directly from raw RMA image.

trained. The enhancement FCNN is trained using both real data from a human hand and simulated data corrupted by additive real radar noise. The FCNN is trained using 65536 simulated and 23040 real human hand RMA images as the input and output images with $\sigma_y = \sigma_z = 1$ mm resulting in cross-range and range resolutions of 1.18 mm. Each simulated sample is generated at a random location in the ROI $y \in [-0.1, 0.1]$, $z \in [0.1, 0.5]$. The synthetic data cover the entire ROI allowing the network to generalize well to location while learning the non-idealities of the imaging scheme. 512 samples of a real hand are collected at each of the 45 locations throughout the ROI as shown in Fig. 5. For both the synthetic samples and real human samples, corresponding ground truth images are generated using (14) and used as training labels. Thus, the training set is comprised of features consisting of real and simulated data and labels consisting of the ideal expected response at each known location.

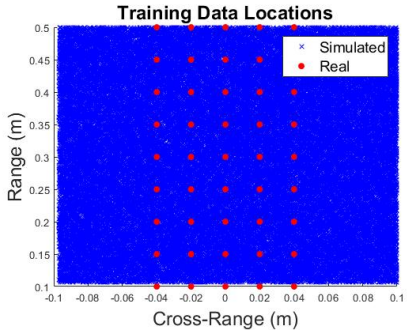


Fig. 5. Locations of the training data used to train the enhancement FCNN. Real data (red) are collected by keeping the hand static at known locations. Simulated data (blue) are generated by choosing locations randomly from the continuous ROI.

The architecture of the proposed enhancement FCNN is shown in Fig. 6. The network consists of four convolution layers of decreasing kernel size each followed by a nonlinear Rectified Linear Unit (ReLU) layer. Each convolutional layer is zero-padded such that the output is identical in size to the input. Training the network for 100 epochs takes 5 hours on a machine with a single NVIDIA GTX1080TI graphics card. Other network architectures and training durations are investigated, but this combination yields high performance while offering real-time efficiency.

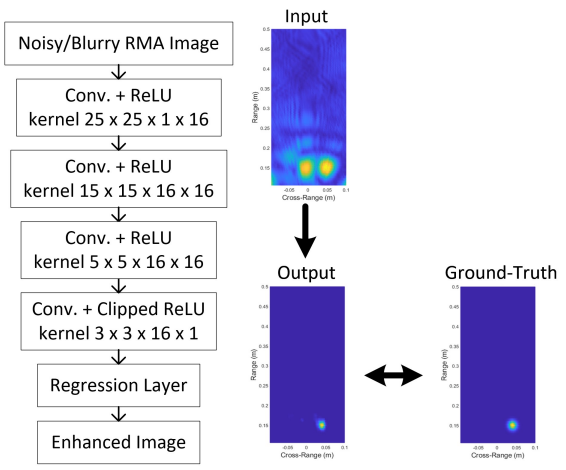


Fig. 6. Architecture of the enhancement FCNN. The selected kernel and layer sizes are capable of adequately learning the non-ideal shape of the distorted RMA image while maintaining high computational efficiency for real-time implementation.

Once the super-resolution FCNN has been trained by the proposed technique, our novel tracking algorithm can be implemented using the particle filter discussed previously. The Doppler-corroborated particle filter is employed to track the position of the hand in the y-z plane and two additional particle filters are used to track the Doppler velocity and cross-range oscillation. The entire signal processing chain for the enhanced feature extraction and tracking method is shown in Fig. 7. The spatiotemporal features are outputted from the algorithm and can be used for many tracking applications. Additionally, if the 2-D location of the hand is desired over the range and cross-range oscillation rate, the algorithm can be easily adapted to output the desired spatial features.

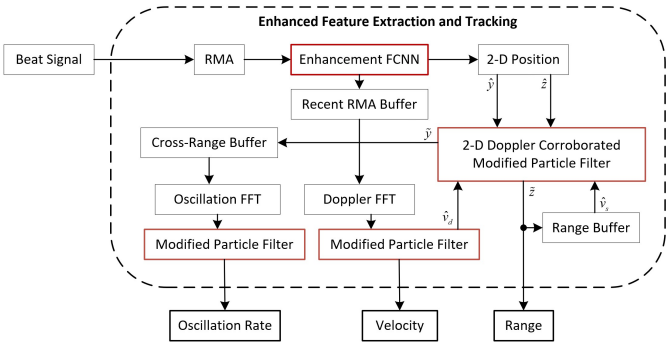


Fig. 7. Enhanced tracking signal processing chain. Key elements to the enhanced methods are highlighted in red.

V. Spatiotemporal Feature Extraction and Tracking Results

In this section, we overview the results of our novel tracking and feature enhancement algorithms beginning with the simple, classical techniques and comparing against our proposed methods. Our enhanced tracking regime demonstrates considerable performance improvement compared with the traditional methods and allows for robust super-resolution tracking on a small radar platform unattainable by existing methods.

A. Ground Truth - Ideal Motion Profile

To verify the feature estimation techniques, a virtual prototyping approach is adopted. A point target is simulated in motion with y-z location and velocity shown in Fig. 8 using (1). This ideal motion profile is employed to compare the tracking performance of our proposed methods to the traditional techniques. Real noise collected from the radar with an empty scene is added to each synthetic beat signal as

$$\tilde{s}(y_T, y_R, k) = \frac{p}{R_T R_R} e^{jk(R_T+R_R)} + \alpha\tilde{\omega}(y_T, y_R, k), \quad (15)$$

where $\tilde{\omega}$ is a complex-valued noise sample corrupting the amplitude and phase of the ideal simulated beat signal and α controls the SNR.

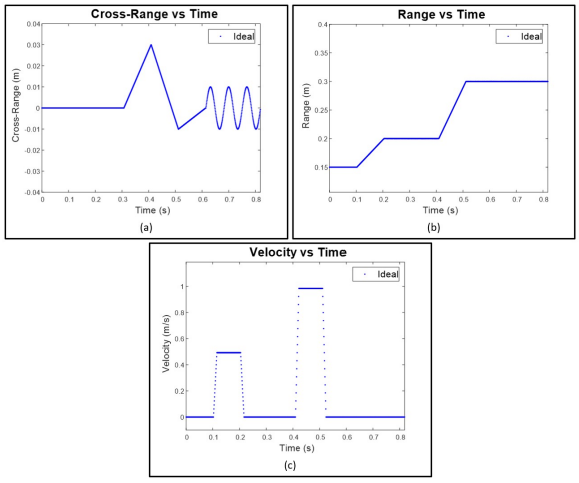


Fig. 8. Ideal motion profile of the target in the (a) cross-range and (b) range directions as well as the (c) range velocity profile against time.

The motion profile shown in Fig. 8 shows the ideal range (z), cross-range (y), and velocity (v) of the target. The motion profile includes independent and joint movement in the range and cross-range domains in addition to sinusoidal cross-range oscillation. For our simulations, 4096 time samples are generated using $p \in [0.5, 1]$ to simulate the variance in the hand's empirical radar cross-section (RCS) as observed from prior hand data and $\alpha \in [1, 3]$ to vary the SNR among samples. Values for p and α are selected randomly within the specified intervals for each time sample and provide a level of stochastic realism to the simulated data.

B. Classical Spatiotemporal Imaging Results

First, the simple tracking methods discussed in Section III-A are implemented to provide baseline performance metrics. The signal processing chain shown in Fig. 4 is performed, extracting the spatiotemporal features. At each iteration, the features are extracted directly from the raw RMA images and are therefore prone to erratic behavior.

Fig. 9 shows the features estimated from the data generated by (15) using the simple methods. The real radar noise and

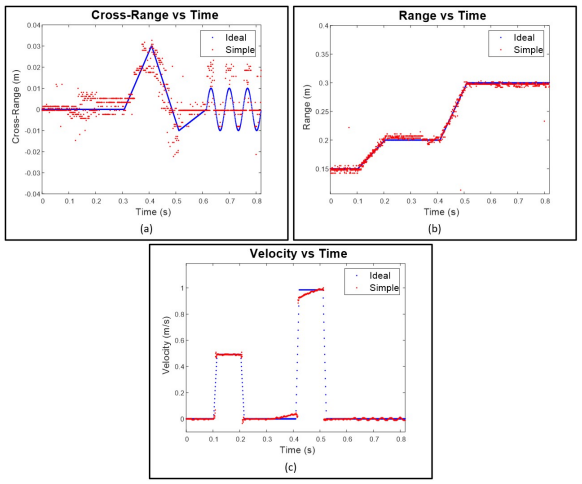


Fig. 9. Motion profile using simple features extraction techniques on each frame for every time step (red) compared with the ideal motion and velocity profiles (blue). The (a) cross-range and (b) range are measured directly from the peak of the RMA image of each frame and the (c) velocity is measured using the Doppler FFT of the raw RMA images using (10) and (11).

varying reflectivity result in outliers and errors in the estimated location and velocity of the target, particularly in the cross-range domain. Without more robust feature extraction and tracking techniques, the performance leaves much to be desired. In the following sections, the performance of the simple tracking methods is quantitatively compared to the enhanced tracking methods and design considerations are discussed.

C. FCNN-Based Super-Resolution Tracking Results

Assuming the motion profile in Fig. 8, our proposed particle filter algorithm is employed in an attempt to more robustly track the 2-D position and Doppler velocity of the target across time, improving the user's control over the interface significantly[1].

First, the particle filter algorithm (PF) without Doppler corroboration is implemented using the data in Fig. 9 as elements of the noisy measurement vector $\boldsymbol{r}$. The PF reduces the effect of the noise on the position estimation and improves the spatiotemporal tracking performance as shown in Fig. 10. The cross-range position tracking is most improved compared to the traditional methods. Next, the Doppler-corroborated particle filter (DPF) is applied to the same set of data further improving the estimation of the range. The outliers in Fig. 10b are mitigated by the DPF in Fig. 10d because the outlying samples result in a sample velocity $\hat{v}_s$ contradicted by the Doppler velocity $\hat{v}_d$ and are weighted as unimportant in the resampling process. The DPF algorithm improves the user experience of our interface by providing a robust, consistent tracking algorithm to smoothly estimate the 2-D position and spatiotemporal signatures of the user's hand. However, the PF and DPF can be further improved by implementing the proposed enhancement FCNN.

After the super-resolution FCNN is trained using the technique discussed in Section IV-D, a validation dataset of

[1]Supplemental material for the reader can be downloaded at http://ieeexplore.org/

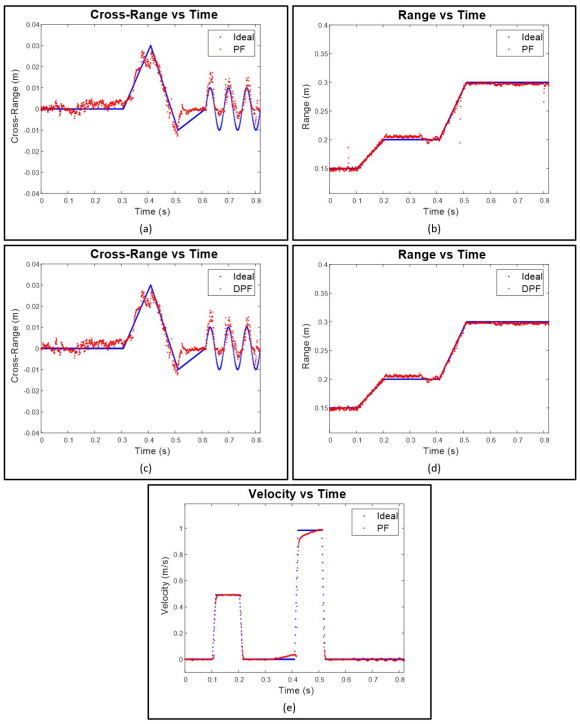


Fig. 10. The particle filter (PF) and Doppler-corroborated particle filter (DPF) algorithms employed for robust spatiotemporal tracking of the simulated gestures through time: improved tracking of the (a) cross-range and (b) range versus time using the PF, (c) cross-range and (d) range versus time using the DPF with $N_z = 16$, and (e) Doppler velocity versus time using a PF approach.

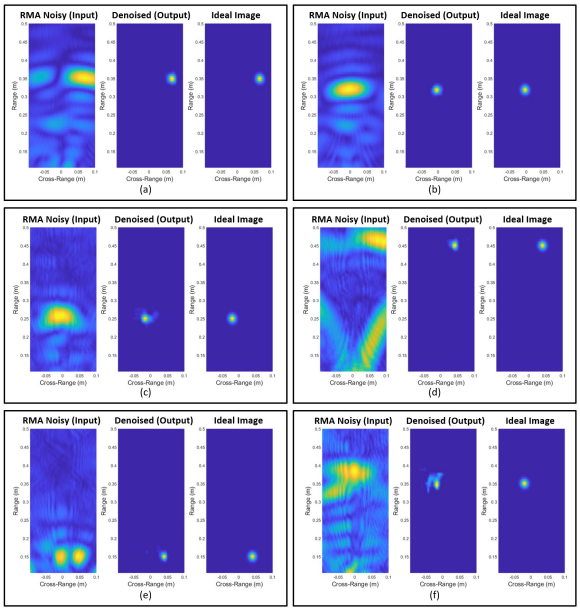


Fig. 11. Enhancement FCNN applied to simulated (a,b) and real hand (c-f) RMA images for image enhancement and improved localization.

identical size to the training set is collected. Fig. 11 shows images enhanced by the enhancement FCNN demonstrating the robustness of the network. Figs. 11a and 11b show simulated point targets enhanced by the FCNN resulting in localization super-resolution. Fig. 11c is an RMA image reconstructed

from a real hand capture close to the middle of the cross-range domain. The 2-D position of the hand is accurately located compared with the ideal image. Similarly, Figs. 11d-11f demonstrate the network's ability to enhance images degraded by small hand RCS in comparison to noise, ghosting due to non-ideal beam patterns, ambient and device noise, and other non-idealities. The proposed enhancement FCNN simultaneously enables localization super-resolution and overcomes device and environment issues. Hence, the features extracted from the enhanced images are much improved compared to the raw RMA images before the FCNN and result in superior tracking performance.

TABLE I
SIMPLE VS ENHANCED LOCALIZATION RMSE

	y (m)	z (m)
Simple	0.0154	0.023
Enhanced	0.0085	0.0083

To quantitatively compare the localization improvement of the enhancement FCNN compared to the simple method, the RMSE in the range and cross-range position are computed on the validation dataset using the two techniques and shown in Table I. The enhancement FCNN improves both the resolution of the RMA images and the localization accuracy for both simulated and real data.

Applying the FCNN and DPF (FCNN-DPF) to the raw data following the ideal motion profile in Fig, 8, yields further tracking improvement over the DPF alone. Fig. 12 demonstrates the tracking performance of the FCNN-DPF on the same data as the previous tracking examples. Applying the FCNN-DPF, the range and cross-range tracking of the target is nearly identical to the ideal motion profile and an improvement in the velocity estimation. Using the identical sporadic data resulting in the poorly estimated cross-range positions in Fig. 9a, the FCNN-DPF yields an estimation nearly identical to the ideal motion profile. Similarly, the cross-range estimates in Fig. 10a and Fig. 10c are outperformed by the FCNN-DPF in Fig. 12a. Compared to the classical techniques and PF/DPF alone, the localization performance of the FCNN-DPF is considerably superior.

Further, the FCNN improves the Doppler estimation robustness. As shown in Fig. 13, the Doppler spectrum SNR is improved when the Doppler processing is performed on the enhanced RMA images as compared to Doppler processing on the raw RMA images. Hence, the enhancement network improves the reliability of the Doppler velocity estimation aiding spatiotemporal tracking.

VI. DISCUSSION AND FUTURE WORK

To quantitatively compare the tracking performance of the various proposed methods, 4096 unique motion profiles are generated and corresponding tracking RMSE is computed for the cross-range, range, and velocity. Displayed in Table II, the RMSE for the cross-range (y), range (z), and velocity (v) improve with the novel algorithms proposed in this paper.

As expected, the baseline simple method yields the greatest error for all three features. Comparing PF and DPF, the cross-

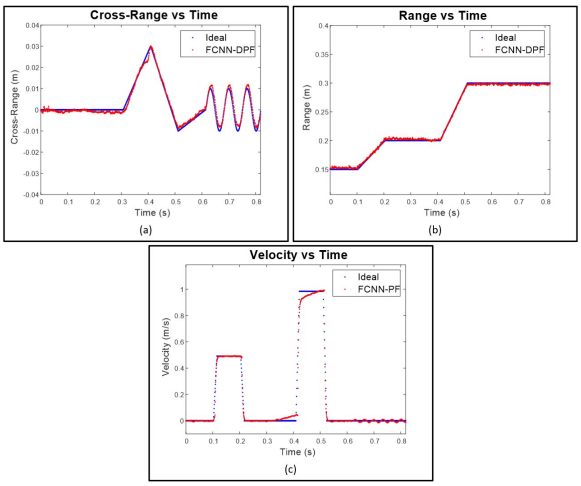


Fig. 12. The FCNN enhanced Doppler-corroborated modified particle filter algorithm.

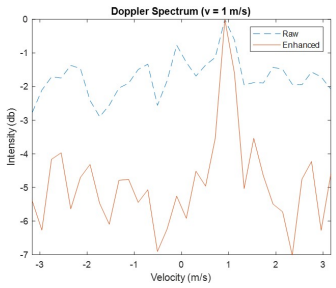


Fig. 13. Comparison of the Doppler velocity spectrum when the Doppler FFT and video pulse integration steps are performed on the raw RMA images compared to the enhanced RMA images. The simulated data contains 128 frames and uses $\alpha = 3$ for every capture to simulate a low SNR scenario.

range and velocity RMSE are identical between the two techniques but the range RMSE is improved due to the dynamic weighting technique. The FCNN alone outperforms the simple method but can be improved by including the PF and DPF after image enhancement. Finally, the FCNN-PF and FCNN-DPF yield identical results for the cross-range and velocity RMSE, as expected, but significant improvement can be noted in the range error. The results in Table II demonstrate the considerably superior tracking performance of the enhanced tracking methods, namely the FCNN-DPF, compared with the simple tracking method. The performance gain realized by implementing the super-resolution FCNN demonstrates the ability of the network to learn the system noise and ambiguities during the training phase using both real and synthetic data.

The average latency of each method, $\bar{\tau}$, is measured as the time duration between the new sample being captured and the estimation process being completed on that sample. The resulting estimates are streamed across the MIDI port or sent to the built-in audio signal generation tool. Additional latency contributed by the subsequent synthesis engine is highly dependent on the software used and device under test; thus, it is not considered as part of the latency due to our methods.

The enhanced tracking methods outperform the exist-

TABLE II
AVERAGE RMSE FOR TRACKING METHODS

	y (mm)	z (mm)	v (mm/s)	$\bar{\tau}$ (ms)
Simple	7.86	22.0	72.4	2.29
PF	5.27	13.6	52.9	2.36
DPF	5.27	6.85	52.9	2.41
FCNN	7.74	12.3	58.4	2.67
FCNN-PF	3.70	7.44	44.5	3.92
FCNN-DPF	3.70	3.07	44.5	3.96

ing techniques in localization resolution, Doppler spectrum SNR, and tracking accuracy; however, there are some necessary trade-offs for this performance gain. The novel super-resolution FCNN yields noteworthy resolution improvement over the theoretical bounds. In ideal conditions, the cross-range and range resolutions of our system are bounded by $\delta_y = 7.5$ cm and $\delta_z = 3.75$ cm, respectively. Using the combination of real hand data and synthetic data used in Table I, the spatial resolution in each direction is computed empirically as $\delta_y = 2.3$ mm and $\delta_z = 1.96$ mm. On the other hand, the effectiveness of the enhancement FCNN is limited by the training set. Since the FCNN is only trained on images within the expected region, extending the ROI outside of the trained region results in performance degradation. If the ROI is changed, the FCNN should be retrained accordingly. In contrast, the simple methods are highly flexible but cannot compete with the performance of the enhancement techniques. However, we have studied the limitations of the particular TI mmWave radar device and found that if the hand is placed outside the ROI defined in the previous section, it will not be detected. Due to device SNR and beamwidth, for most hand sizes, the reflections back to the radar will not be strong enough for detection. Additionally, we have tested the proposed FCNN in smaller ROIs and found similar results without retraining. For other array topologies, the proposed methods can be easily applied, although the FCNN will need to be trained accordingly.

While the Doppler-corroborated particle filter improves the tracking robustness, it requires a high throughput framework to function properly. Since the DPF relies on accurate Doppler velocity estimation, the pulse repetition interval PRI, T_{PRI}, must be sufficiently small such that high hand velocities are within the resolvable range. The PRI is impacted most significantly by the time per iteration of the signal processing chain. Given our framework is currently released in the prototyping stage as a MATLAB program, the latency performance does not match that of a real-time implementation on a more efficient embedded device. Hence, typical throughput times limit the PRI to around 4 ms. At $T_{PRI} = 4$ ms using a 77 GHz device, the maximum resolvable velocity is 0.24 m/s. With this limitation, some rapid movements at high velocities may result in Doppler spectrum aliasing.

While the software package presented in this article serves as a framework for demonstrating and prototyping the proposed tracking and super-resolution algorithms, the inherent latency of the signal processing steps is a key issue in HCI and must be addressed. In our research, the largest contributor of latency in our proposed system is the hand-off between the radar device and MATLAB over UDP and shared memory, at an average of 1.93 ms. Rather than streaming to data MATLAB, a real-time solution can be implemented on the TI radar device's built-in DSP, thus providing a more efficient throughput as the DSP has direct access to the samples as they are taken. Additionally, several steps in the signal processing chain will increase in efficiency with an embedded solution. Employing small window sizes, $N_z = 16$ and the number of FFT spatial points is 64, the DPF and FFT computation times can be further reduced compared to the relatively inefficient MATLAB implementation. We would also like to note that a significant decrease in latency was achieved by optimizing the implementation using GPU accelerated coding. A similar approach could be taken on an embedded solution leveraging the highly parallelizable nature of many of the steps in the signal processing chain (FFT, CNN, Gaussian distribution computation). Comparing the computational efficiency among the algorithms, the latency cost for the more robust algorithms is insignificant in proportion to the performance gain, even in the MATLAB implementation. In latency tests, the average response time using the FCNN-DPF was 3.96 ms from user input to MIDI signaling. While most MIDI interfaces outperform this metric, we believe our framework demonstrates a competitive throughput cycle time compared to existing technology and can be further improved by a more efficient implementation.

Hand-tracking using a mmWave radar has both advantages and drawbacks compared with other sensing regimes. In this article, we employ a single radar to develop and demonstrate robust tracking algorithms for mmWave devices. While the best performance is likely achieved through a sensor fusion technique, a radar-based implementation may be optimal if privacy is a concern using optical cameras or issues such as occlusion and lighting conditions must be taken into account. Compared to optical and RGB+D solutions, mmWave is more versatile and reliable, operating well under occlusion, in any temperature or lighting environment, and offers precise depth information of the entire scene. For a musical interface, these advantages may not be often fully realized; however, the novel tracking methods proposed in this article are applicable for many HCI applications. On the other hand, mmWave sensors cannot meet the performance of optical solutions when it comes to cross-range resolution due to the limited aperture size, making multi-object and finger tracking much more challenging. As such, many applications in HCI, computer vision, automated driving, etc. employ radar (and lidar) and optical imaging devices with sensor fusion algorithms to achieve further improved performance at an increased cost. For these applications, our proposed algorithms can aid in sensor fusion by significantly increasing the performance contribution from the radar sensors.

Several alternatives exist to mmWave radar sensing, namely wearable, handheld, and optical devices. Wearable and handheld sensing solutions offer highly precise spatiotemporal features but are often not preferable compared to contactless sensors [29], [30]. In terms of cost, mmWave radar devices are in the same price bracket as the popular Kinect and Leap Motion optical sensors on the order of $\$100 - \$200$. Attempts

using multiple RGB cameras [31], [32] show promising results; however, a single device is much preferred due to the cumbersome nature of multi-camera systems. Single RGB+D solutions have been proposed using generative pose tracking [33], [34] and learning-based generative pose tracking [35], [36]. However, all of these methods suffer tremendously under occlusion or scene clutter, both of which can be overcome using mmWave radar. Some deep learning-oriented solutions have shown quite promising results [37], [38], but constructing a sufficient dataset for meaningful supervised training remains a challenge.

Our proposed interface tracks the 2-D position and velocity of the user's hand to control note selection and two user-selected parameters, a marked improvement over the prior work on mmWave radar using the Google Soli tracking only 1-D range for parameter control [16]. However, optical solutions enable tracking of both hands [3], [7], [9], [33]–[38] or hand and finger position [8], [12] for even finer musical control, with some scenario-specific drawbacks. As radar technology improves and larger apertures become widely available, tracking individual fingers will become increasingly plausible and could yield comparable or superior results to optical solutions due to superior depth resolution.

Compared to prior work on hand-tracking with mmWave devices, our proposed methods yield competitive results. Past work using radar devices achieves, at best, an average range tracking error of 2 cm on human hand localization [17]. Our enhanced tracking technique yields a mean range tracking error of 1.89 mm, improving tracking by more than a factor of ten. In [39], a 4 GHz bandwidth mmWave sensor achieves a 2-D position RMSE of 1.16 mm tracking a thumb, at distances closer than 10 cm. Comparatively, our enhanced tracking technique tracks a human hand across much larger distances and still achieves a competitive 2-D position RMSE of 3.4 mm. At the time of this paper, we are not aware of any other prior work on hand-tracking using mmWave devices. To our knowledge, the system proposed in this paper offers unprecedented hand gesture tracking performance using a single mmWave sensor.

The most direct musical interface comparison to our framework, is the Theremin, as both are controlled by the hand's proximity to the sensor. The pitch of the Theremin is controlled continuously by the hand's vertical location, whereas our interface tracks the range of the hand digitally and selects a note from the user-defined scale. While the Theremin uses two antennas, one for volume control and the other for pitch control, a total of two degrees of freedom, our framework offers three degrees of freedom (range, cross-range, and velocity), thus providing three controllable parameters. As previously mentioned, the musical interface promoted in this article supports Theremin-like gestures for note selection and parameter control. However, high-velocity percussive gestures could be implemented using our high-fidelity tracking algorithms, with some limitations. Small values of the weighting factor, $\boldsymbol{A}$, in the particle filter algorithm can result an excessively smoothed and overly damped system limiting the ability of the system to track sudden movements. Depending on the desired application, finely tuning this parameter is essential for enabling proper gestural control. Our proposed interface is an evolved Theremin, utilizing a modern mmWave sensor for precise tracking in 2-D space (expansion to 3-D can be easily implemented with the proper hardware). In contrast to a Theremin, our musical interface is significantly less effortful in note selection, allowing simple and intuitive inclusion of the additional parameter controls and increasing accessibility to the user-base. One of the authors is a skilled guitar and violin instrumentalist with a background in electronic music production. From the perspective of an experienced musician, the proposed methods offer an elegant new musical interface capable of generating unique phrases previously only possible via manual transcription and provides the musician a sufficient and consistent level of control.

For future work, several promising routes are left to be explored. First, further development can be explored by implementing our proposed methods onto a real-time embedded platform. Additionally, using multiple MIMO radars or a larger MIMO array, a multiple-hand and individual finger tracking interface can be investigated, thus further extending the application space of our robust tracking methods. Finally, our novel super-resolution tracking algorithms can easily be adapted to offer an elegant, efficient solution to a host of acute hand-tracking problems in the HCI domain and even employed in sensor-fusion systems.

VII. Conclusion

Our FCNN-based super-resolution framework successfully demonstrates the viability of acute human hand-tracking for HCI using mmWave sensors. We validated and implemented our spatiotemporal signal processing algorithms and robust tracking algorithms in the form of a contactless musical interface; however, this article also serves to demonstrate the broad effectiveness of mmWave technology for a multitude of near-field acute hand-tracking applications. First, simple feature extraction and tracking methods were introduced, followed by an enhanced approach leveraging the Doppler-corroborated particle filter algorithm and enhancement FCNN to achieve robust tracking and super-resolution in a non-ideal imaging scenario. The methods are compared demonstrating noticeable improvement using the FCNN-DPF over the classical techniques. Additionally, our work offers competitive tracking estimation and localization performance compared to prior methods in the literature for both mmWave and optical implementations. Our entire software implementation and real-time radar interface platform are freely available at request. The novel FCNN-based super-resolution and tracking algorithms presented in this article offer an elegant solution to many contactless HCI problems.

Acknowledgment

The first author's work was supported by the imec USA summer internship program. We would like to extend thanks to Dr. Gonzalo Vaca Castano for his insights in developing the particle filter algorithm and computer vision approach.

REFERENCES

[1] T. Winkler, "Making motion musical: Gesture mapping strategies for interactive computer music," in *Proc. Int. Computer Music Conf.*, Banff, Canada, 1995, pp. 261–264.

[2] K. D. Skeldon, L. M. Reid, V. McInally, B. Dougan, and C. Fulton, "Physics of the theremin," *American Journal of Physics*, vol. 66, no. 11, pp. 945–955, 1998.

[3] R. Polfreman, "Multi-modal instrument: towards a platform for comparative controller evaluation," in *Proc. Int. Computer Music Conf.* Proc. Int. Computer Music Conf., July 2011, pp. 147–150. [Online]. Available: https://eprints.soton.ac.uk/353226/

[4] S. Trail, M. Dean, G. Odowichuk, T. F. Tavares, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect." in *Proc. Int. Conf. New Interfaces for Musical Expression*, 2012.

[5] S. Sentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg, "Crossole: A gestural interface for composition, improvisation and performance using kinect." in *Proc. Int. Conf. New Interfaces for Musical Expression*, 2012.

[6] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Trans. on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[7] A. R. Jensenius, "Kinectofon: Performing with shapes in planes," in *Proc. Int. Conf. on New Interfaces for Musical Expression*, Daejeon, Korea, 2013, pp. 196–197.

[8] J. Han and N. Gold, "Lessons learned in exploring the leap motion™ sensor for gesture-based instrument design," in *Proc. Int. Conf. on New Interfaces for Musical Expression*. London, United Kingdom: Goldsmiths University of London, 2014, pp. 371–374.

[9] L. Hantrakul and K. Kaczmarek, "Implementations of the leap motion in sound synthesis, effects modulation and assistive performance tools." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, London, United Kingdom, 2014.

[10] D. Brown, N. Renney, A. Stark, C. Nash, and T. Mitchell, "Leimu: Gloveless music interaction using a wrist mounted leap motion," in *Proc. Int. Conf. on New Interfaces for Musical Expression*, Brisbane, Austrailia, 2016, pp. 300–304.

[11] A. Tindale, A. Kapur, and G. Tzanetakis, "Training surrogate sensors in musical gesture acquisition systems," *IEEE Trans. on Multimedia*, vol. 13, no. 1, pp. 50–59, 2011.

[12] O. Nieto and D. Shasha, "Hand gesture recognition in mobile devices: Enhancing the musical experience," *Proc. Computer Music Multidisciplinary Research*, vol. 13, 2013.

[13] M. Akbari and H. Cheng, "Real-time piano music transcription based on computer vision," *IEEE Trans. on Multimedia*, vol. 17, no. 12, pp. 2113–2121, 2015.

[14] J. W. Smith, S. Thiagarajan, R. Willis, Y. Makris, and M. Torlak, "Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique," *IEEE Access*, vol. 9, pp. 10 893–10 902, 2021.

[15] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3041–3048, 2019.

[16] F. Bernardo, N. Arner, and P. Batchelor, "O soli mio: exploring millimeter wave radar for musical interaction." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, vol. 17, 2017, pp. 283–286.

[17] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using rf backscatter," in *Proc. USENIX Symposium on Networked Systems Design and Implementation*, 2015, pp. 189–204.

[18] Y. Dai, T. Jin, Y. Song, H. Du, and D. Zhao, "Cnn-based multiple-input multiple-output radar image enhancement method," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6840–6844, 2019.

[19] Y. Sun, X. Liang, H. Fan, M. Imran, and H. Heidari, "Visual hand tracking on depth image using 2-d matched filter," in *Proc. UK/China Emerging Technologies*, Aug. 21-22, 2019, Glasgow, United Kingdom, pp. 1–4.

[20] S. Rao, "Intro to mmwave sensing : Fmcw radars," Jul 2020. [Online]. Available: https://training.ti.com/node/1139153

[21] J. W. Smith, M. E. Yanik, and M. Torlak, "Near-field mimo-isar millimeter-wave imaging," in *Proc. IEEE Radar Conf.*, 2020, pp. 1–6.

[22] M. E. Yanik and M. Torlak, "Near-field mimo-sar millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31 801–31 819, 2019.

[23] M. E. Yanik, D. Wang, and M. Torlak, "Development and demonstration of mimo-sar mmwave imaging testbeds," *IEEE Access*, vol. 8, pp. 126 019–126 038, 2020.

[24] V. Winkler, "Range doppler detection for automotive fmcw radars," in *Proc. European Radar Conf.*, Oct. 10-12, 2007, Munich, Germany, pp. 166–169.

[25] J. Kim, J. Chun, and S. Song, "Joint range and angle estimation for fmcw mimo radar and its application," *arXiv:1811.06715*, 2018.

[26] J. García, A. Gardel, I. Bravo, J. L. Lázaro, and M. Martínez, "Tracking people motion based on extended condensation algorithm," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 606–618, 2013.

[27] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 35–39, 2019.

[28] "Texas instruments mmwave studio." [Online]. Available: https://www.ti.com/tool/MMWAVE-STUDIO

[29] L. Pardue and W. Sebastian, "Hand-controller for combined tactile control and motion tracking." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, 2013, pp. 90–93.

[30] P. Neto, J. N. Pires, and A. P. Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *Industrial Robot*, vol. 37, no. 2, pp. 137–147, 2010.

[31] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Proc. European Conf. on Computer Vision*. Springer, 2012, pp. 640–653.

[32] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proc. IEEE Int. Conf. on Computer Vision*, 2013, pp. 2456–2463.

[33] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect." in *Proc. Brit. Mach. Vision Conf*, vol. 1, no. 2, 2011, pp. 101.1–101.11.

[34] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proc. IEEE Int. Conf. on Computer Vision*, 2015, pp. 3325–3333.

[35] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3213–3221.

[36] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. on Graphics*, vol. 35, no. 4, pp. 1–12, 2016.

[37] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. on Graphics*, vol. 33, no. 5, pp. 1–10, 2014.

[38] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *Proc. European Conf. on Computer Vision*. Springer, 2016, pp. 346–361.

[39] Z. Li, Z. Lei, A. Yan, E. Solovey, and K. Pahlavan, "Thumouse: A micro-gesture cursor input through mmwave radar-based interaction," in *Proc. IEEE Int. Conf. on Consumer Electronics*, Jan. 4-6, 2020, Las Vegas, NV, USA, pp. 1–9.

An FCNN-Based Super-Resolution mmWave Radar Framework for Contactless Musical Instrument Interface

Josiah W. Smith[1], Orges Furxhi[2], and Murat Torlak[1]

[1]Dept. of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX, United States
[2]Camera Systems and Computational Imaging, Imec, Kissimmee, FL, United States

***Abstract*—In this paper, we propose a framework for contactless human-computer interaction (HCI) using novel tracking techniques based on deep learning-based super-resolution and tracking algorithms. Our system offers unprecedented high-resolution tracking of hand position and motion characteristics by leveraging spatial and temporal features embedded in the reflected radar waveform. Rather than classifying sample from a predefined a set of hand gestures, as common in existing work on deep learning with mmWave radar, our proposed imager employs a regressive full convolutional neural network (FCNN) approach to achieve spatial super-resolution improving localization. While the proposed techniques are suitable for a host of tracking applications, this article focuses on their application as a musical interface to demonstrate the robustness of the gesture sensing pipeline and deep learning signal processing chain. The user can control the instrument by varying the position and velocity of their hand above the vertically-facing sensor. By employing a commercially-available multiple-input-multiple-output (MIMO) radar rather than a traditional optical sensor, our framework demonstrates the efficacy of the mmWave sensing modality for fine motion tracking and offers an elegant solution to a host of HCI tasks. Additionally, we provide a freely-available software package and user interface for controlling the device, streaming the data to MATLAB in real-time, and increasing accessibility to the signal processing and device interface functionality utilized in this article.**

***Index Terms*—deep learning, human-computer interaction (HCI), fully-convolutional neural network (FCNN), millimeter-wave (mmWave), multiple-input multiple-output (MIMO), radar perception, super-resolution**

I. Introduction

Radar perception for human-computer interaction (HCI) on multiple-input-multiple-output (MIMO) millimeter-wave (mmWave) radars has emerged as a promising solution to a variety of sensing problems. The physical nature of millimeter-waves offers a safe method for high-resolution imaging where optical sensors may fail due to insufficient lighting, fog, or other line-of-sight interference. Additionally, mmWave sensors are considered less-invasive than optical counterparts and promote user privacy. Ultra-wideband MIMO devices enable centimeter-level spatial resolution with a small profile device. As a result, precise spatial information of a target scene can be easily acquired from such imaging devices at a low cost.

mmWave sensors are relatively modern technology, but some of the earliest electronic interfaces were contactless devices for physically expressive musical control including the Radio Drum and Theremin [1]. Russian physicist Leon Theremin demonstrated his noncontact musical instrument in 1921, an interface controlled by the proximity of the musician's hand to an antenna using beat-frequency oscillators and a capacitive sensing apparatus [2]. More recently, computer vision approaches have been adopted for the innovation of contactless new musical interfaces (NMIs), most of which rely on optical camera solutions. Extensive prior work exists on optical-based NMIs using popular sensors such as the Microsoft Kinect and Leap Motion.

In [3], Polfreman uses the Kinect to track the 3-D position of both hands of a standing performer to construct a multi-modal instrument. Trail *et al.* present a pitched percussion hyper-instrument to track the tips of two mallets simultaneously with the Kinect [4]. Crossole, designed by Senturk *et al.*, is a Kinect-based metainstrument that visualizes chord progressions as virtual blocks resembling a crossword puzzle [5]. Schramm *et al.* use the Kinect to analyze and classify motions of a orchestral conductor [6]. In [7], the Kinect is used to track hand motion across time and then translated to music using the inverse Fourier transform of the physical pattern using a sonification technique called sonomotiongram.

Alternatively, the popular Leap Motion controller is capable of modeling the entire hand, including the fingers, which allows for even more detailed hand posture-based gesture control to be explored for musical interface development. Using the Leap Motion sensor, Han *et al.* developed two NMIs, *Air Keys* and *Air Pad*. *Air Keys* tracks the motion and position of each finger to recognize when and which keys the musician is pressing and playing the desired notes. Similarly, *Air Pad* tracks the hand position to create a 2-D virtual drum pad played by pressing specific regions in a 2-D horizontal plane, thus requiring accurate 3-D hand-tracking [8]. Hantrakul and Kaczmarek use the Leap Motion controller to track both hands for controlling MIDI (Musical Instrument Digital Interface) instruments and virtual effects [9]. Similarly, Leimu pairs the Leap Motion with an inertial measurement unit (IMU) demonstrating improved performance over the Leap Motion controller alone for musical interface [10]. Other solutions have been attempted, such as employing non-invasive force sensing resistors to enhance "traditional" instruments by learning and monitoring for gestures performed by the musician [11].

In the optical HCI domain, [12] proposes a musical interface using only a portable RGB camera to recognize hand gestures using a gesture classification technique. Akbari and Cheng developed a system to transcribe music played on a piano in real-time using optical cameras positioned to view the keys [13]. These projects have yielded high-performing real-time musical interfaces capable of consistent high-accuracy motion tracking but require several key design constraints, namely specific lighting conditions and line-of-sight. As shown in this article, mmWave sensors overcome these major obstacles while providing superior privacy through the means of advanced spatiotemporal algorithms. However, little work has been done towards gestural musical interfaces on mmWave radar sensors using hand-tracking techniques. Even though extensive research exists on static and dynamic gesture recognition using deep learning models and mmWave radars [14], [15], Google ATAP's Project Soli is the only effort using mmWave radar for musical interface, using gesture recognition and 1-D position estimation to control the parameters of audio synthesizers [16].

The novel framework presented in this article offers a major advancement for near-field mmWave hand-tracking and an accessible MATLAB software platform for further investigation into real-time mmWave HCI and algorithm innovation. 2-D localization performance is considerably improved from past work [17] by employing a novel deep learning-based technique to improve the resolution beyond the theoretical limitations.

It is important to note our approach is contrary to gesture classification, wherein the objective is to determine the class of a sample from a set of predefined classes as in [14], [15]; rather, we apply a novel fully convolutional neural network (FCNN) to preserve the geometry of the image and perform super-resolution for improved localization. Prior work on resolution improvement using FCNNs has been limited to the far-field domain with large apertures [18]; however, our novel approach unifies FCNN-based super-resolution with near-field imaging on a small (8-channel) array and is shown to improve hand-tracking performance significantly. Additionally, a particle filter tracking algorithm is presented to further improve tracking robustness by employing the Doppler effect. Compared to prior work on gesture tracking using optical solutions [3], [7]–[9], [12], [19], our approach offers fine hand-tracking using a single mmWave sensor offering higher depth resolution with superior privacy. This article proposes a novel hand-tracking method for musical interface by fusing spatiotemporal algorithms, deep learning-enhanced feature extraction, and robust position tracking algorithms. To aid further development and prototyping for real-time mmWave gesture applications, the entire software implementation is available by request to the corresponding author. To our knowledge, this proposed framework is the first openly available software package supporting real-time data streaming from a mmWave radar into MATLAB for streamlined signal processing and deep learning algorithm development.

The rest of this paper is formatted as follows. Section II provides an overview of the frequency modulated continuous wave (FMCW) radar signal model and feature extraction methods. In Section III, two robust tracking algorithms and estimation techniques are presented. The system implementation is discussed in Section IV and results are shown in Section V. Section VI provides a discussion of the performance, design constraints, and distinct advantages of the two tracking methods in Sections III-A and III-B, followed finally by conclusions.

Notation: Throughout this paper, vectors and matrices are set in boldface, using lowercase letters for vectors and uppercase letters for matrices. The superscripts ^{T} and * denote the transpose and conjugation operations, respectively. The identity matrix of size $N \times N$ is expressed as $\boldsymbol{I}_N$ and $\mathbf{1}_N$ is the all-ones vector of size $N \times 1$. Spatial coordinates are treated as continuous to support continuously distributed target scenes and all time variables are modeled in discrete-time.

II. Preliminaries of MIMO-FMCW Radar Signaling

In this section, we overview the propagation model for the FMCW radar chirp signal and examine the spatiotemporal features of a target in motion. The imaging scenario, as shown in Fig. 1, consists of a multistatic linear MIMO array facing vertically. Orthogonality is leveraged in time by employing time-division-multiplexing MIMO (TDM-MIMO), wherein the transmitters are activated at separate time instances. Throughout this paper, the musician's hand is modeled as a point reflector located at the point (y, z), an assumption that holds given the physical limitations of the device and scenario examined in this article and has been verified empirically.

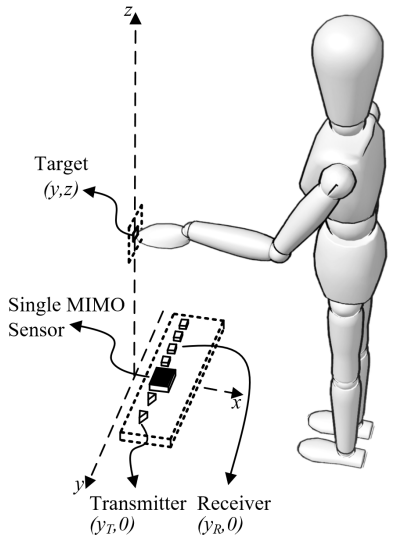


Fig. 1. The imaging geometry, where the linear MIMO array faces vertically and the musician moves their hand throughout the y-z plane.

A. MIMO-FMCW Signal Model

The FMCW chirp signal model is well documented in literature [20]–[22] and is discussed in this section for reference and continuity throughout this paper. Considering a single transmitter/receiver pair located at $(y_T, 0)$ and $(y_R, 0)$ in the y-z plane, respectively, and an ideal point target with reflectivity

p located at (y, z), the time sampled FMCW beat signal can be expressed in discrete-time as

$$s(y_T, y_R, n_k) = \frac{p}{R_T R_R} e^{j(k_0+\Delta n_k)(R_T+R_R)}, \quad (1)$$

where R_T, R_R are the distances from the transmitter and receiver to the point target, respectively, n_k is the wavenumber index, $k_0 = 2\pi f_0/c$ is the starting wavenumber corresponding to the starting frequency f_0, and $\Delta = 2\pi K/(cf_S)$ is the wavenumber step size with K being the chirp slope, f_S being the sampling frequency, and c is the speed of light.

To ease the subsequent signal processing, it is desirable to approximate the multistatic MIMO beat signal, represented in (1) as its corresponding monostatic equivalent using the approximation developed in [23] as

$$\hat{s}(y', n_k) = s(y_T, y_R, n_k) e^{-j(k_0+\Delta n_k)\frac{d_y^2}{4z_0}}, \quad (2)$$

valid only for small d_y, the distance between the transmitter and receiver elements, where z_0 is a reference plane typically given as the center of the target scene. Taking y' as the locations of the virtual elements located at the midpoints between each transceiver pair and R as the corresponding distance from each virtual element to the point reflector, the resulting monostatic beat signal approximates to

$$\hat{s}(y', n_k) \approx \frac{p}{R^2} e^{j2(k_0+\Delta n_k)R}. \quad (3)$$

From (3), the spatial location, (y, z), of the target is embedded in the radar beat signal, in the form of the radial distance R.

B. Doppler Radar Signal Processing

The relative velocity of a target can be extracted from the beat signal expressed in (3) by exploiting the Doppler effect. As discussed in [24], by transmitting a series of chirp waveforms at a known pulse repetition interval (PRI), T_{PRI}, the velocity of a moving target can be identified as the frequency component along the chirp index dimension given by

$$\hat{s}(y', n_k, n_c) = \frac{p}{R^2} e^{j(2(k_0+\Delta n_k)R + \frac{4\pi v T_{PRI}}{\lambda_0} n_c)}, \quad (4)$$

where R is the initial range of the target, v is the velocity of the target, λ_0 is the wavelength corresponding to f_0, and n_c is the chirp index,

Thus, the beat signal sampled across time is a 2-D complex sinusoidal with frequencies corresponding to the range and velocity of the target on the first and second dimensions, respectively. Subsequently, to extract the range and velocity, traditional methods perform a 2-D fast Fourier transform (FFT) over a matrix whose rows or columns consist of subsequent chirps.

C. Range Migration Algorithm Image Reconstruction

To achieve high-fidelity 2-D localization, we employ the range migration algorithm (RMA) over traditional range-angle FFT methods [20], whose localization accuracy is known to be inferior [25]. The primary goal of the RMA is to reconstruct the target scene's reflectivity function, $p(y, z)$. For a distributed target, the beat signal can be modeled as the superposition of the backscattered signal at every point in the scene, neglecting the amplitude terms, as

$$\hat{s}(y', n_k) = \iint p(y, z) e^{j2(k_0+\Delta n_k)R} dy dz, \quad (5)$$

This target model assumes a spatially distributed target whose reflectivity only depends on spatial location and neglects the any frequency dependence of the reflectivity function. Inverting (5) using the method of stationary phase, the reflectivity function, $p(y, z)$, can be estimated efficiently by

$$\hat{p}(y, z) = \text{IFT}_{\text{2D}}^{(k_y,k_z)} \left[\mathcal{S} \left[\text{IFT}_{\text{1D}}^{(y')} [\hat{s}^*(y', n_k)] \right] \right], \quad (6)$$

where $\mathcal{S}[\bullet]$ is the Stolt interpolation operation [23] and FT[$\bullet$], IFT[$\bullet$] are the forward and inverse Fourier transform operators. To avoid aliasing in the image sampling criteria must be considered [22]. Spatial resolution along the y and z directions are constrained by the physical and device limitations and are expressed as

$$\delta_y = \frac{\lambda_c z_0}{2D_y}, \quad (7)$$

$$\delta_z = \frac{c}{2B}, \quad (8)$$

where λ_c is the wavelength corresponding to the frequency at the center of the chirp sweep, D_y is the aperture size along the y direction, and z_0 is the center of the imaging scene [22].

After the 2-D reflectivity function of the target scene is recovered, the hand position is estimated subsequently as

$$\{\hat{y}, \hat{z}\} = \arg\max_{\{y,z\}} \hat{p}(y, z). \quad (9)$$

Further, the aforementioned Doppler principle can be leveraged to extract the velocity of the target by Fourier analysis over successive chirps. To optimally exploit the deep learning framework discussed in Section III-B2 and reduce the required computation complexity, the velocity is extracted after the RMA is performed and hand location is estimated.

As evident in (4), the velocity is decoupled from the wavenumber index and is the scaled frequency component along the chirp index dimension. As a result, the phase term corresponding to the velocity is preserved in the reconstructed image, $\hat{p}(y, z)$. Therefore, the velocity profile can be obtained by performing an FFT across the chirp index, n_c, dimension of the recent images. Rather than performing the FFT across the 3-D array, $\hat{p}(y, z, n_c)$, we perform the FFT over the slice of the image corresponding to the estimated position, $\hat{y}$, yielding the velocity profile along the z-direction, where n_d is the velocity index, as

$$\hat{d}(z, n_d) = \text{FFT}_{\text{1D}}^{(n_c)} \left[\hat{p}(y, z, n_c) \Big|_{y=\hat{y}} \right]. \quad (10)$$

Finally, the velocity can be estimated from (10) using video pulse integration by

$$\hat{v}_d = \arg\max \sqrt{\int |\tilde{d}(z, n_d)|^2 dz}. \quad (11)$$

The velocity computed by this method is referred to as the Doppler velocity. The recovered velocity using this approach is limited by the timing and physical constraints between $[-\frac{\lambda_0}{4T_{PRI}}, \frac{\lambda_0}{4T_{PRI}}]$. Later, the Doppler velocity is employed to improve the tracking performance using the Doppler corroborated particle filter.

III. Spatiotemporal Imaging on mmWave Radar

In this section, we present the methods for our proposed imager capable of high accuracy hand-tracking for HCI. The contribution of this article is the advancement in algorithm performance for 2-D localization by utilizing both the novel super-resolution FCNN and proposed tracking algorithm. While we will investigate the application of such algorithms as an NMI, our mmWave radar-based sensing algorithms can be applied to a host of HCI problems.

It is important to note that this work is not intended to compete with the computational efficiency of embedded HCI solutions and existing musical interfaces. Rather, the main contributions of this article are novel algorithms for super-resolution spatiotemporal hand-tracking and a freely-downloadable platform to increase accessibility and encourage further research in this arena. As such, we will focus primarily on the development of the algorithms and their localization performance. Discussions on performance and implementation issues are considered secondary and are addressed in Sections IV and VI.

A. Classical Spatiotemporal Feature Extraction Techniques

In this section, we introduce the simple approach to spatiotemporal sensing for contactless musical instrument interface. While our system generally tracks the 2-D position and velocity of the user's hand, we have identified three underlying features to achieve fine control of the musical interface: range, cross-range oscillation, and velocity. By the geometry given in Fig. 1, we define the range as the position of the hand along the z-axis, i.e. the vertical displacement between the sensor and the user's hand. Similarly, cross-range is defined as the position of the hand along the y-axis. Subsequently, cross-range oscillation is the rate at which the hand oscillates in the cross-range direction. Velocity is given by the velocity of the hand with respect to the range z-axis. These parameters are selected such that the output musical interface is controlled primarily by the range of the musician's hand and secondarily by the cross-range oscillation and velocity. However, these parameters can be assigned by the user based on preference using the MIDI interface, as discussed later. Throughout the remainder of this article, we will refer to these parameters as features extracted from the radar beat signal.

Under the simple gesture tracking regime, the 2-D location and velocity $(\hat{y}, \hat{z}, \hat{v}_d)$ are extracted from the reconstructed image and buffer of recent images using (9) and (11). In the next section, the three parameters extracted from the raw data are treated as a vector called the noisy measurement vector $\boldsymbol{r}$. In the optimal scenario, the bandwidth, antenna array size, and the signal-to-noise-ratio (SNR) are quite large, tending towards infinity. For the case of an 8 channel automotive mmWave radar and a human hand, the bandwidth is limited (4 GHz), the antenna array size is small ($D_y = 2\lambda_c$), and the reflectivity of the hand is not high compared to the noise level. As a result, simply extracting the maximum from the reconstructed RMA images yields sporadic location and velocity estimates. Even in the ideal case, the spatial resolution of our system along the y and z directions is $\delta_y = 7.5$ cm and $\delta_z = 3.75$ cm, respectively. Several other factors are not taken into account in the classical, direct tracking method including beam-pattern, residual phase errors, and antenna coupling. All these limitations and non-idealities in the imaging scenario degrade the image and result in noisy location and velocity estimates; however, many of these issues analytical forms and cannot be solved directly classical methods. To address these issues, we present a novel data-driven approach employing an FCNN for super-resolution and image enhancement.

B. FCNN-Based Super-Resolution Feature Extraction and Particle Filter Tracking Methods

In this section, we improve upon the simple tracking techniques to overcome noise and foundational non-idealities in the imaging scenario, yielding a much-improved user experience. The concepts demonstrated in this section are applicable for many tracking and high-resolution imaging applications beyond the scope of musical interfaces.

To improve the tracking robustness of the proposed musical interface, we adopt the well-known particle filter [26] and present a novel modification. While traditional methods such as the extended Kalman filter (EKF) employ a motion model, our implementation of the particle filter bypasses the need for a deterministic motion model. The particle filter is selected for this application as other traditional approaches have demonstrated poor tracking performance in our experimentation, yielding either sporadic localization or overly damped, sluggish estimation. Additionally, the particle filter is advantageous as it can track non-linear dynamics and does not require prior knowledge of the motion model or noise parameters for robust localization.

In our modification of the particle filter, the control input is a weighted movement towards the newest measurement. To demonstrate our proposed algorithm, consider the case of simultaneous location estimation along the y and z directions. The new noisy measurement vector, $\boldsymbol{r}$, has two elements, the newest estimates of location, $\hat{y}$ and $\hat{z}$, which are extracted by the methods described in the prior section. Algorithm 1 details the modified particle filter implementation. For 2-D localization, $\boldsymbol{X}_n$ is a matrix of size $N \times 2$, whose rows are the (y, z) coordinates of each particle at time index n, where N is the number of particles, and $\boldsymbol{w}_n$ is the vector of weights corresponding to each particle. The estimates of the 2-D location (also known as the estimated states) form the vector $\boldsymbol{s}_n$ and the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$ is denoted as $G(\boldsymbol{\mu}, \boldsymbol{\Sigma})$. Before executing the iterative algorithm, the initial particle states matrix, $\boldsymbol{X}_0$, and initial weights vector, $\boldsymbol{w}_0$, are initialized with as random locations throughout the region of interest (ROI) and uniform weights, respectively.

Algorithm 1: Modified Particle Filter Algorithm

input : $\boldsymbol{r} = [\hat{y}, \hat{z}]^T$
output: $\boldsymbol{s}_n = [\tilde{y}, \tilde{z}]^T$
1 $\boldsymbol{X}_n \leftarrow$ rows of $\boldsymbol{X}_{n-1}$ sampled using weights $\boldsymbol{w}_{n-1}$;
2 $\boldsymbol{X}_n \leftarrow \boldsymbol{X}_n + \mathbf{1}_N \boldsymbol{a}^T (\boldsymbol{r} - \boldsymbol{s}_{n-1}) + \boldsymbol{\psi}$;
3 $\boldsymbol{w}_n \leftarrow e^{-\frac{1}{2}(\boldsymbol{X}_n - \boldsymbol{s}_{n-1})^T \boldsymbol{\Sigma}_w^{-1} (\boldsymbol{X}_n - \boldsymbol{s}_{n-1})}$;
4 $\boldsymbol{s}_n \leftarrow \frac{1}{\mathbf{1}_N^T \boldsymbol{w}_n} \boldsymbol{X}_n^T \boldsymbol{w}_n$;

Proper handling of the key steps, (step 2) resampling of the particle states and (step 3) computing new weights, is essential to effectively implement our novel particle filter algorithm.

The particle resampling process involves moving the particles towards the new measurement by a specified weight. $\boldsymbol{a} = [a_y, a_z]^T$ is a vector whose two elements provide weight to the noisy estimates $\hat{y}$ and $\hat{z}$, respectively. The size of $\boldsymbol{a}$, $\boldsymbol{r}$, and $\boldsymbol{s}_n$ can be varied depending on the number of parameters to be tracked by the particle filter. Hence, the new measurements do not dominate the motion tracking but have a weighted influence on the localization procedure. Fig. 2 demonstrates the resampling process with $a_y = a_z = 0.5$. Note that before computing the new weights, particle diffusion is performed by adding the perturbation term $\boldsymbol{\psi}$. The random vector $\boldsymbol{\psi}$ is Gaussian distributed with zero mean and predefined covariance matrix $\boldsymbol{\Sigma}_{\boldsymbol{\psi}}$.

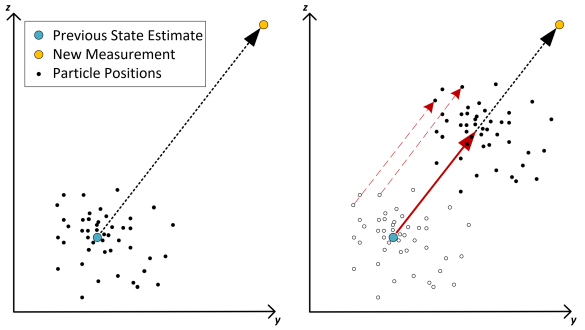


Fig. 2. A visual example of the modified particle filter algorithm resampling process. The particle locations are resampled by a shift transformation towards the new measurement according to the weight vector $\boldsymbol{a}$, where $a_y = a_z = 0.5$.

The new weights are computed from a multivariate Gaussian distribution with the previously estimated states, $\boldsymbol{s}_{n-1}$, as the mean vector and a predefined covariance matrix $\boldsymbol{\Sigma}_w$. Therefore, particles closer to the previously estimated state are assigned a higher weight than those farther away. This results in a tendency towards small changes in the state estimations while monitoring for movement from the current position. For many applications requiring precise and consistent localization and motion tracking on mmWave radar, our modified particle filter algorithm is an ideal fit as it tends to a steady-state estimation of the states but remains active in monitoring the noisy sensor input.

1) Doppler-Corroborated Real-Time Weighting: In this section, we present a dynamic weighting technique for updating $\boldsymbol{a}$ in real-time by exploiting the dependence between position and velocity. Our approach considers corroboration between the Doppler velocity estimate and the velocity estimated from the range samples as a measure of the new measurement's reliability. Thus, the dependability of the Doppler velocity can improve tracking of the target position along the range (z) dimension even in the presence of noisy position estimates. After the Doppler velocity is calculated by (11), the recent range estimates are used to calculate the sample velocity ($\hat{v}_s$) by the least squares estimator as

$$\hat{v}_s = \frac{N_z T_{PRI} \sum_m (\boldsymbol{z}^{(m)} m) - T_{PRI} \sum_m \boldsymbol{z}^{(m)} \sum_m m}{N_z \sum_m (\boldsymbol{z}^{(m)})^2 - \left(\sum_m \boldsymbol{z}^{(m)}\right)^2}, \tag{12}$$

where $\boldsymbol{z}^{(m)}$ is the m^{th} element of the vector of recent $\hat{z}$ estimates, $\boldsymbol{z}$, with $\boldsymbol{z}^{(N_z-1)}$ being the most recent.

The difference between the Doppler estimated velocity and sample estimated velocity is computed as $\Delta_v = |\hat{v}_d - \hat{v}_s|$ and used in the reward function (13) to update the weight placed on the new noisy measurement in real-time.

$$a_z(\Delta_v) = \begin{cases} a_{z,0} \cos\left(\frac{2\pi T_{PRI} \Delta_v}{\lambda_0}\right) & \text{if } \Delta_v \leq \frac{\lambda_0}{4T_{PRI}} \\ 0 & \text{if } \Delta_v > \frac{\lambda_0}{4T_{PRI}} \end{cases} \tag{13}$$

When the sample velocity is close to Doppler velocity, i.e. Δ_v is small, the reward function is close to $a_{z,0}$. Hence, the new measurement is corroborated by the reliable Doppler velocity and weighted accordingly. Outliers and erroneous measurements contradicting the Doppler velocity are given less importance during the particle resampling process. To implement the Doppler corroborated particle filter, $\boldsymbol{a} = [a_y, a_z(\Delta_v)]^T$ is dynamically updated by (13) at each iteration of Algorithm 1.

2) Improved 2-D Position Estimation by Enhancing FCNN: The modified particle filter algorithm improves the tracking consistency and smoothness; however, several issues such as instrumentation delay, ambient/device noise, multistatic effects, and non-spherical beam patterns remain unaddressed and degrade tracking performance. To overcome these non-idealities, we present a novel FCNN-based technique for image enhancement that improves the 2-D position estimation, subsequent tracking accuracy, and Doppler spectrum SNR. Compared to prior FCNN synthetic aperture radar (SAR) techniques employing far-field assumptions and trained on synthetically generated data [18], our enhancement FCNN method operates on near-field images, improves localization even with a small aperture, and is trained using a novel technique allowing the network to learn the environment and device noise, near-field beam pattern, and multistatic effects.

To train the enhancement FCNN, we construct a dataset consisting of both real human hand data and synthetically generated data. Real hand data are collected by capturing frames while the user holds their hand at known locations relative to the device and synthetic data are used to supplement the training set. Each synthetic sample is generated by simulating a MIMO beat signal using (5) with one ideal point target located at a known location and additive real device noise, collected from the radar. The simulated locations are randomized to uniformly cover the ROI. Both the real and synthetic data are

used as features in the FCNN training process, thus enabling the network to fit the non-ideal beam pattern, real multipath and multistatic effects, empirical reflection of a human hand, device and ambient noise, and hand positions throughout the ROI.

To train the image-to-image regression FCNN, each training feature (real or synthetic image) must correspond to a ground truth label. The ground truth label images are synthetically generated by the model

$$\mathcal{I}(y,z) = e^{-(y-y_0)^2/\sigma_y^2-(z-z_0)^2/\sigma_z^2} \tag{14}$$

where the width of the expected target located at (y_0, z_0) is dictated by σ_y and σ_z in the y and z dimensions, respectively, yielding resolutions of $1.18\sigma_y$ and $1.18\sigma_z$ according to the 3 dB beamwidth definition [27]. Each label is generated using the requisite knowledge of the location of the human hand or target of each feature image. During training, the FCNN learns the highly nonlinear relationship between distorted, blurred RMA images and the ideal images generated using (14). Our novel training technique results in a robust and generalizable FCNN that improves image SNR and localization by fitting to the non-ideal imaging constraints. Further, the trained network enables localization precision beyond the physical limitations of the device improving tracking performance significantly. FCNN training is discussed in Section IV-D and results are presented and discussed in Section V-C.

Additionally, by isolating the peak corresponding to the human hand, clutter and phase noise at other positions are mitigated thereby improving the Doppler spectrum SNR and subsequent velocity estimation. Thus, the FCNN enhances both the spatial and temporal features extracted from the radar beat signal before the particle filter. Uniting the proposed particle filter and enhancement FCNN, the range, cross-range oscillation, and velocity are robustly tracked by our novel algorithms and mapped to musical interface controls.

IV. System Design and Implementation

In this section, we present the system implementation for both the classical tracking techniques and our novel super-resolution feature extraction and tracking algorithms discussed in the previous section.

A. Hardware and Software Implementation

The hardware employed in the proposed system consists of a Texas Instruments (TI) AWR1243 automotive radar in conjunction with a DCA1000EVM real-time data capture adapter. The TI radar is a MIMO-FMCW mmWave radar with an operating bandwidth of 4 GHz and a center frequency of 79 GHz. In this research, we utilize the linear MIMO array consisting of 2 transmit antenna (TX) elements, separated by $2\lambda_c$, and 4 receive antenna (RX) elements, separated by $\lambda_c/2$. The resulting virtual array has 8 equally spaced virtual elements separated by $\lambda_c/4$ [20]. The calibration methodology discussed in [23] is adopted to mitigate range bias, constant phase errors, and instrumentation delay. In this process, data are captured from a corner reflector at a known location and used to identify range bias and phase offsets among the antennas. Unlike an optical or infrared calibration, this process is invariant of lighting and temperature constraints as well as user hand sizes, etc. Thus, the one-time calibration applies to a variety of environments and users.

The software platform for signal processing, visualization, and machine learning is written in MATLAB. Despite its inferior computational efficiency compared to other languages, MATLAB is employed to provide an accessible platform for researchers to engage with this work and rapidly prototype custom real-time algorithms using our custom tools. Once the algorithms are validated on a PC, they can be implemented onto such embedded devices for optimized application-specific usage. For a positive user experience as a musical interface, latency and timing issues must be taken into account, and are discussed in Section VI.

B. Real-Time Data Retrieval and Interactive MATLAB User Interface

To stream the data from the device into MATLAB, a custom UDP interface software is written. This routine is implemented efficiently in C++ and is capable of receiving the sequential UDP packets, organizing the packets to form each chirp, and providing the data to MATLAB over shared memory.

A custom interactive MATLAB graphical user interface (GUI), shown in Fig. 3, is written to serve as the single user interface for our framework. The MATLAB GUI interfaces with TI mmWave Studio [28] to control the hardware setup and initializes the UDP interface, bypassing the need for user setup outside our GUI. The radar continuously captures and streams data into MATLAB using the fully-integrated implementation. While MATLAB does not offer the computational speed necessary for real-time system implementation, it is capable of completing the data capture, signal processing, deep learning, visualization, and signal output at around 250 Hz, from our experimentation. In the early prototyping phase, we consider this throughput sufficient for investigating the performance of the super-resolution tracking algorithms and a simple musical interface.

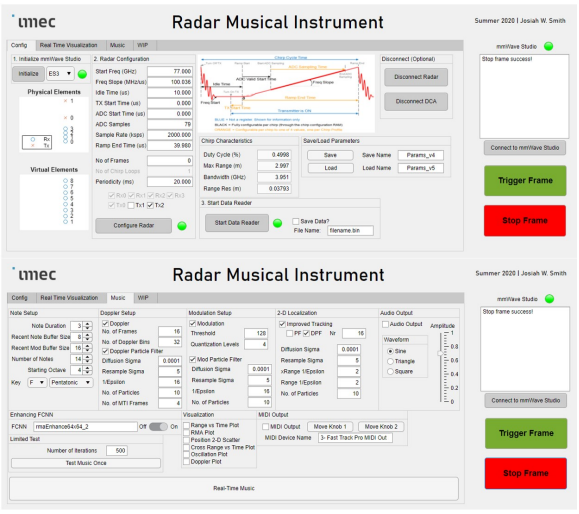


Fig. 3. Interactive MATLAB GUI: device setup and music generation pages.

Using our proposed methods, the software extracts high-resolution spatiotemporal features of the user's hand and maps them to corresponding output using either a built-in audio output tool or the included MIDI output. The custom MATLAB GUI provides an accessible option for investigating and demonstrating our methods as well as an open-source platform to stimulate further collaborative investigation by the multimedia and radar communities.

As previously mentioned, the primary mechanism to control the output of the proposed musical interface is the range (z-position) of the user's hand. Using the built-in audio output tool and the MIDI output, the range of the user's hand controls the note selection directly. Unlike the Theremin, which allows for continuous note selection, our interface quantizes the user input into predefined subregions corresponding to notes defined by the user. The subregions and allowed notes can be programmed by the user in the interactive MATLAB GUI. To play the desired note, the user must move their hand vertically to the position corresponding to that note. Similarly, the secondary parameters, cross-range oscillation, and velocity can be adjusted by the user by oscillating their hand back-and-forth in the y-direction or moving to the next note with a high or low velocity. The built-in audio output tool employs the cross-range oscillation to control a vibrato effect (low-frequency modulation of the audio signal). Thus, using this tool, the user can select the desired note by varying the range and perform vibrato at a desired rate by oscillating their hand at the same rate. Alternatively, the MIDI output tool provides the cross-range oscillation and velocity as MIDI parameters to be specified by the user in a virtual instrument environment connected to the MIDI output of our musical interface. Hence, our proposed algorithms are implemented to operate similarly to a MIDI keyboard with the hand range controlling the note selection and cross-range oscillation and velocity acting as MIDI parameters for the user to assign.

C. Simple Feature Extraction and Tracking Algorithm Signal Processing Chain

The signal processing chain for the simple feature extraction and tracking method is shown in Fig. 4. The beat signal is loaded into MATLAB where the preprocessing discussed in the previous section is performed (RMA and peak finding) and the user inputs (2-D location and velocity) are converted into audio or MIDI output by extracting the spatiotemporal features using (9) and (11). In this article, the location and velocity of the user's hand are used for musical gestural interface; however, our novel algorithms can easily be applied to many different HCI applications and even for 3-D localization, provided a sufficient 2-D array. The reconstructed RMA image and raw feature extracted by the classical techniques can be utilized by the particle filter algorithm and super-resolution FCNN to improve the tracking performance.

D. Super-Resolution Framework - Training FCNN and Implementing Particle Filter Algorithm

To implement our super-resolution feature extraction and tracking framework, the super-resolution FCNN must be first

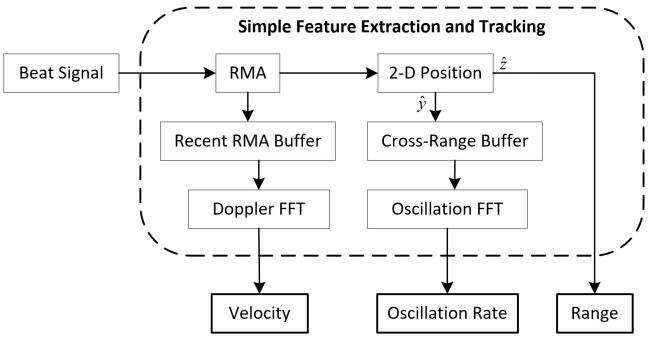


Fig. 4. Simple tracking signal processing chain. After RMA is performed on beat signal, features are extracted directly from raw RMA image.

trained. The enhancement FCNN is trained using both real data from a human hand and simulated data corrupted by additive real radar noise. The FCNN is trained using 65536 simulated and 23040 real human hand RMA images as the input and output images with $\sigma_y = \sigma_z = 1$ mm resulting in cross-range and range resolutions of 1.18 mm. Each simulated sample is generated at a random location in the ROI $y \in [-0.1, 0.1]$, $z \in [0.1, 0.5]$. The synthetic data cover the entire ROI allowing the network to generalize well to location while learning the non-idealities of the imaging scheme. 512 samples of a real hand are collected at each of the 45 locations throughout the ROI as shown in Fig. 5. For both the synthetic samples and real human samples, corresponding ground truth images are generated using (14) and used as training labels. Thus, the training set is comprised of features consisting of real and simulated data and labels consisting of the ideal expected response at each known location.

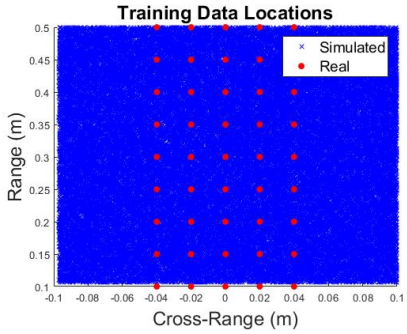


Fig. 5. Locations of the training data used to train the enhancement FCNN. Real data (red) are collected by keeping the hand static at known locations. Simulated data (blue) are generated by choosing locations randomly from the continuous ROI.

The architecture of the proposed enhancement FCNN is shown in Fig. 6. The network consists of four convolution layers of decreasing kernel size each followed by a nonlinear Rectified Linear Unit (ReLU) layer. Each convolutional layer is zero-padded such that the output is identical in size to the input. Training the network for 100 epochs takes 5 hours on a machine with a single NVIDIA GTX1080TI graphics card. Other network architectures and training durations are investigated, but this combination yields high performance while offering real-time efficiency.

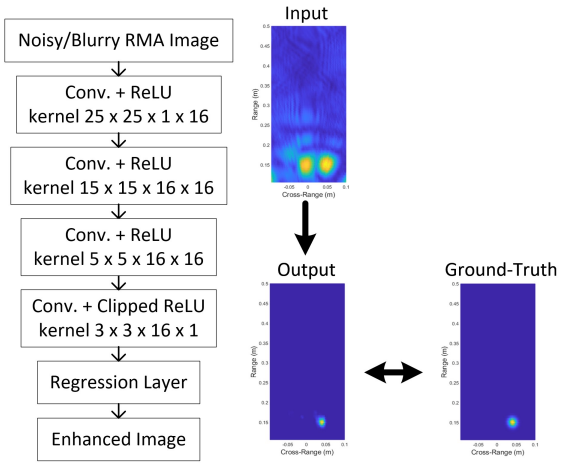


Fig. 6. Architecture of the enhancement FCNN. The selected kernel and layer sizes are capable of adequately learning the non-ideal shape of the distorted RMA image while maintaining high computational efficiency for real-time implementation.

Once the super-resolution FCNN has been trained by the proposed technique, our novel tracking algorithm can be implemented using the particle filter discussed previously. The Doppler-corroborated particle filter is employed to track the position of the hand in the y-z plane and two additional particle filters are used to track the Doppler velocity and cross-range oscillation. The entire signal processing chain for the enhanced feature extraction and tracking method is shown in Fig. 7. The spatiotemporal features are outputted from the algorithm and can be used for many tracking applications. Additionally, if the 2-D location of the hand is desired over the range and cross-range oscillation rate, the algorithm can be easily adapted to output the desired spatial features.

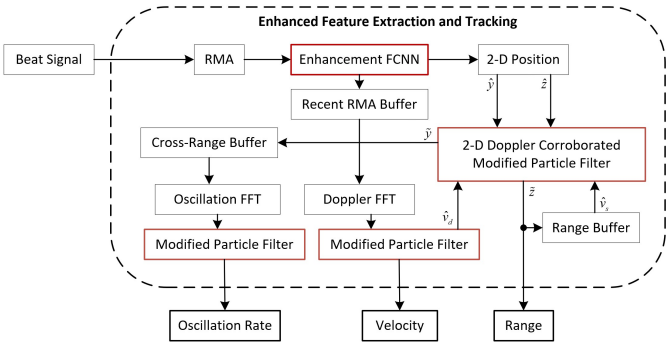


Fig. 7. Enhanced tracking signal processing chain. Key elements to the enhanced methods are highlighted in red.

V. Spatiotemporal Feature Extraction and Tracking Results

In this section, we overview the results of our novel tracking and feature enhancement algorithms beginning with the simple, classical techniques and comparing against our proposed methods. Our enhanced tracking regime demonstrates considerable performance improvement compared with the traditional methods and allows for robust super-resolution tracking on a small radar platform unattainable by existing methods.

A. Ground Truth - Ideal Motion Profile

To verify the feature estimation techniques, a virtual prototyping approach is adopted. A point target is simulated in motion with y-z location and velocity shown in Fig. 8 using (1). This ideal motion profile is employed to compare the tracking performance of our proposed methods to the traditional techniques. Real noise collected from the radar with an empty scene is added to each synthetic beat signal as

$$\tilde{s}(y_T, y_R, k) = \frac{p}{R_T R_R} e^{jk(R_T+R_R)} + \alpha\tilde{\omega}(y_T, y_R, k), \quad (15)$$

where $\tilde{\omega}$ is a complex-valued noise sample corrupting the amplitude and phase of the ideal simulated beat signal and α controls the SNR.

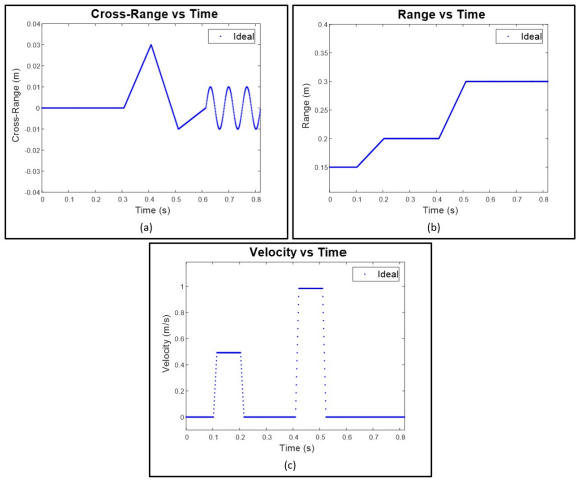


Fig. 8. Ideal motion profile of the target in the (a) cross-range and (b) range directions as well as the (c) range velocity profile against time.

The motion profile shown in Fig. 8 shows the ideal range (z), cross-range (y), and velocity (v) of the target. The motion profile includes independent and joint movement in the range and cross-range domains in addition to sinusoidal cross-range oscillation. For our simulations, 4096 time samples are generated using $p \in [0.5, 1]$ to simulate the variance in the hand's empirical radar cross-section (RCS) as observed from prior hand data and $\alpha \in [1, 3]$ to vary the SNR among samples. Values for p and α are selected randomly within the specified intervals for each time sample and provide a level of stochastic realism to the simulated data.

B. Classical Spatiotemporal Imaging Results

First, the simple tracking methods discussed in Section III-A are implemented to provide baseline performance metrics. The signal processing chain shown in Fig. 4 is performed, extracting the spatiotemporal features. At each iteration, the features are extracted directly from the raw RMA images and are therefore prone to erratic behavior.

Fig. 9 shows the features estimated from the data generated by (15) using the simple methods. The real radar noise and

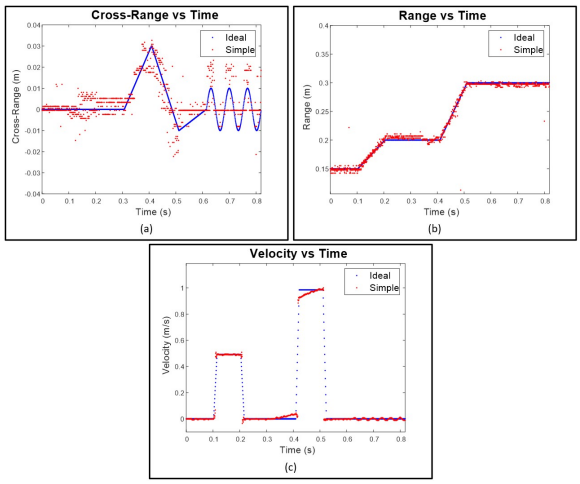


Fig. 9. Motion profile using simple features extraction techniques on each frame for every time step (red) compared with the ideal motion and velocity profiles (blue). The (a) cross-range and (b) range are measured directly from the peak of the RMA image of each frame and the (c) velocity is measured using the Doppler FFT of the raw RMA images using (10) and (11).

varying reflectivity result in outliers and errors in the estimated location and velocity of the target, particularly in the cross-range domain. Without more robust feature extraction and tracking techniques, the performance leaves much to be desired. In the following sections, the performance of the simple tracking methods is quantitatively compared to the enhanced tracking methods and design considerations are discussed.

C. FCNN-Based Super-Resolution Tracking Results

Assuming the motion profile in Fig. 8, our proposed particle filter algorithm is employed in an attempt to more robustly track the 2-D position and Doppler velocity of the target across time, improving the user's control over the interface significantly[1].

First, the particle filter algorithm (PF) without Doppler corroboration is implemented using the data in Fig. 9 as elements of the noisy measurement vector $\boldsymbol{r}$. The PF reduces the effect of the noise on the position estimation and improves the spatiotemporal tracking performance as shown in Fig. 10. The cross-range position tracking is most improved compared to the traditional methods. Next, the Doppler-corroborated particle filter (DPF) is applied to the same set of data further improving the estimation of the range. The outliers in Fig. 10b are mitigated by the DPF in Fig. 10d because the outlying samples result in a sample velocity $\hat{v}_s$ contradicted by the Doppler velocity $\hat{v}_d$ and are weighted as unimportant in the resampling process. The DPF algorithm improves the user experience of our interface by providing a robust, consistent tracking algorithm to smoothly estimate the 2-D position and spatiotemporal signatures of the user's hand. However, the PF and DPF can be further improved by implementing the proposed enhancement FCNN.

After the super-resolution FCNN is trained using the technique discussed in Section IV-D, a validation dataset of identical size to the training set is collected. Fig. 11 shows images enhanced by the enhancement FCNN demonstrating the robustness of the network. Figs. 11a and 11b show simulated point targets enhanced by the FCNN resulting in localization super-resolution. Fig. 11c is an RMA image reconstructed

[1]Supplemental material for the reader can be downloaded at http://ieeexplore.org/

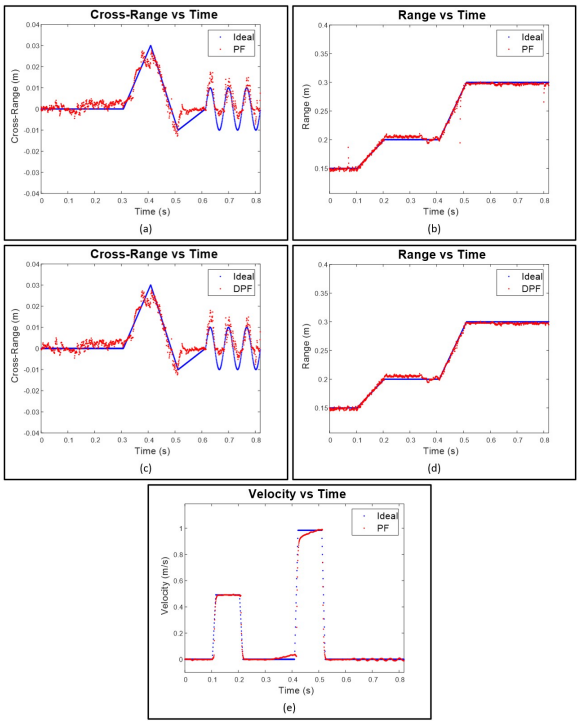


Fig. 10. The particle filter (PF) and Doppler-corroborated particle filter (DPF) algorithms employed for robust spatiotemporal tracking of the simulated gestures through time: improved tracking of the (a) cross-range and (b) range versus time using the PF, (c) cross-range and (d) range versus time using the DPF with $N_z = 16$, and (e) Doppler velocity versus time using a PF approach.

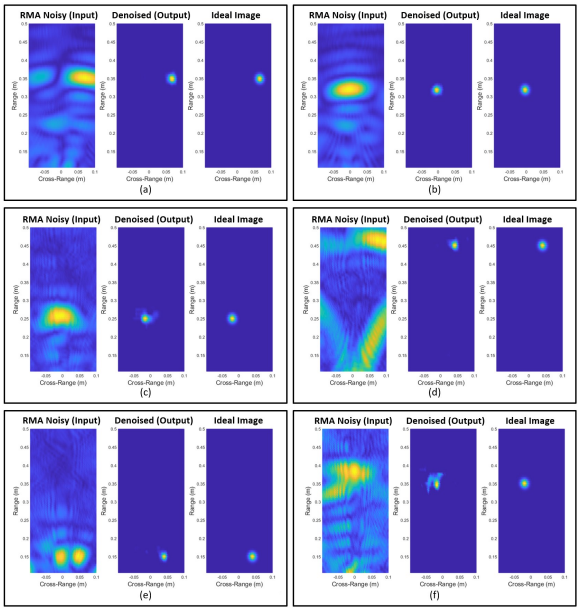


Fig. 11. Enhancement FCNN applied to simulated (a,b) and real hand (c-f) RMA images for image enhancement and improved localization.

from a real hand capture close to the middle of the cross-range domain. The 2-D position of the hand is accurately located compared with the ideal image. Similarly, Figs. 11d-11f demonstrate the network's ability to enhance images degraded by small hand RCS in comparison to noise, ghosting due to non-ideal beam patterns, ambient and device noise, and other non-idealities. The proposed enhancement FCNN simultaneously enables localization super-resolution and overcomes device and environment issues. Hence, the features extracted from the enhanced images are much improved compared to the raw RMA images before the FCNN and result in superior tracking performance.

TABLE I
SIMPLE VS ENHANCED LOCALIZATION RMSE

	y (m)	z (m)
Simple	0.0154	0.023
Enhanced	0.0085	0.0083

To quantitatively compare the localization improvement of the enhancement FCNN compared to the simple method, the RMSE in the range and cross-range position are computed on the validation dataset using the two techniques and shown in Table I. The enhancement FCNN improves both the resolution of the RMA images and the localization accuracy for both simulated and real data.

Applying the FCNN and DPF (FCNN-DPF) to the raw data following the ideal motion profile in Fig, 8, yields further tracking improvement over the DPF alone. Fig. 12 demonstrates the tracking performance of the FCNN-DPF on the same data as the previous tracking examples. Applying the FCNN-DPF, the range and cross-range tracking of the target is nearly identical to the ideal motion profile and an improvement in the velocity estimation. Using the identical sporadic data resulting in the poorly estimated cross-range positions in Fig. 9a, the FCNN-DPF yields an estimation nearly identical to the ideal motion profile. Similarly, the cross-range estimates in Fig. 10a and Fig. 10c are outperformed by the FCNN-DPF in Fig. 12a. Compared to the classical techniques and PF/DPF alone, the localization performance of the FCNN-DPF is considerably superior.

Further, the FCNN improves the Doppler estimation robustness. As shown in Fig. 13, the Doppler spectrum SNR is improved when the Doppler processing is performed on the enhanced RMA images as compared to Doppler processing on the raw RMA images. Hence, the enhancement network improves the reliability of the Doppler velocity estimation aiding spatiotemporal tracking.

VI. DISCUSSION AND FUTURE WORK

To quantitatively compare the tracking performance of the various proposed methods, 4096 unique motion profiles are generated and corresponding tracking RMSE is computed for the cross-range, range, and velocity. Displayed in Table II, the RMSE for the cross-range (y), range (z), and velocity (v) improve with the novel algorithms proposed in this paper.

As expected, the baseline simple method yields the greatest error for all three features. Comparing PF and DPF, the cross-range and velocity RMSE are identical between the two techniques but the range RMSE is improved due to the dynamic weighting technique. The FCNN alone outperforms the simple method but can be improved by including the PF and DPF after image enhancement. Finally, the FCNN-PF and FCNN-DPF yield identical results for the cross-range and velocity RMSE, as expected, but significant improvement can be noted in the range error. The results in Table II demonstrate the considerably superior tracking performance of the enhanced tracking methods, namely the FCNN-DPF, compared with the simple tracking method. The performance gain realized by implementing the super-resolution FCNN demonstrates the ability of the network to learn the system noise and ambiguities during the training phase using both real and synthetic data.

The average latency of each method, $\bar{\tau}$, is measured as the time duration between the new sample being captured and the estimation process being completed on that sample. The resulting estimates are streamed across the MIDI port or sent to the built-in audio signal generation tool. Additional latency contributed by the subsequent synthesis engine is highly dependent on the software used and device under test; thus, it is not considered as part of the latency due to our methods.

The enhanced tracking methods outperform the exist-

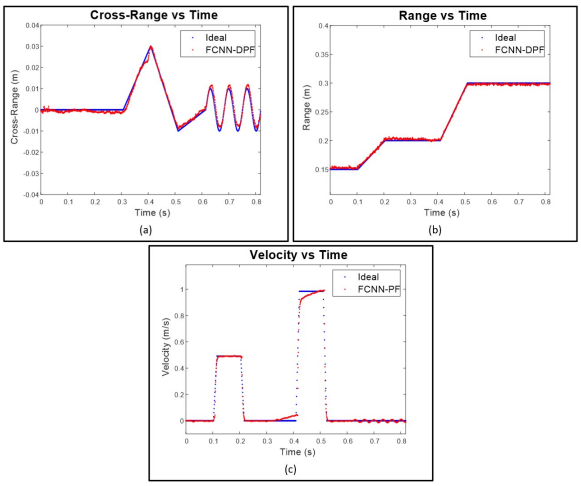


Fig. 12. The FCNN enhanced Doppler-corroborated modified particle filter algorithm.

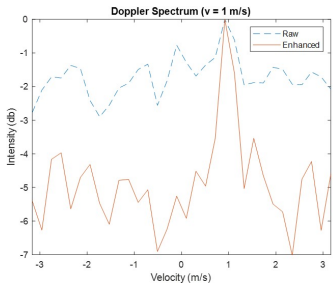


Fig. 13. Comparison of the Doppler velocity spectrum when the Doppler FFT and video pulse integration steps are performed on the raw RMA images compared to the enhanced RMA images. The simulated data contains 128 frames and uses $\alpha = 3$ for every capture to simulate a low SNR scenario.

TABLE II
AVERAGE RMSE FOR TRACKING METHODS

	y (mm)	z (mm)	v (mm/s)	$\bar{\tau}$ (ms)
Simple	7.86	22.0	72.4	2.29
PF	5.27	13.6	52.9	2.36
DPF	5.27	6.85	52.9	2.41
FCNN	7.74	12.3	58.4	2.67
FCNN-PF	3.70	7.44	44.5	3.92
FCNN-DPF	3.70	3.07	44.5	3.96

ing techniques in localization resolution, Doppler spectrum SNR, and tracking accuracy; however, there are some necessary trade-offs for this performance gain. The novel super-resolution FCNN yields noteworthy resolution improvement over the theoretical bounds. In ideal conditions, the cross-range and range resolutions of our system are bounded by $\delta_y = 7.5$ cm and $\delta_z = 3.75$ cm, respectively. Using the combination of real hand data and synthetic data used in Table I, the spatial resolution in each direction is computed empirically as $\delta_y = 2.3$ mm and $\delta_z = 1.96$ mm. On the other hand, the effectiveness of the enhancement FCNN is limited by the training set. Since the FCNN is only trained on images within the expected region, extending the ROI outside of the trained region results in performance degradation. If the ROI is changed, the FCNN should be retrained accordingly. In contrast, the simple methods are highly flexible but cannot compete with the performance of the enhancement techniques. However, we have studied the limitations of the particular TI mmWave radar device and found that if the hand is placed outside the ROI defined in the previous section, it will not be detected. Due to device SNR and beamwidth, for most hand sizes, the reflections back to the radar will not be strong enough for detection. Additionally, we have tested the proposed FCNN in smaller ROIs and found similar results without retraining. For other array topologies, the proposed methods can be easily applied, although the FCNN will need to be trained accordingly.

While the Doppler-corroborated particle filter improves the tracking robustness, it requires a high throughput framework to function properly. Since the DPF relies on accurate Doppler velocity estimation, the pulse repetition interval PRI, T_{PRI}, must be sufficiently small such that high hand velocities are within the resolvable range. The PRI is impacted most significantly by the time per iteration of the signal processing chain. Given our framework is currently released in the prototyping stage as a MATLAB program, the latency performance does not match that of a real-time implementation on a more efficient embedded device. Hence, typical throughput times limit the PRI to around 4 ms. At $T_{PRI} = 4$ ms using a 77 GHz device, the maximum resolvable velocity is 0.24 m/s. With this limitation, some rapid movements at high velocities may result in Doppler spectrum aliasing.

While the software package presented in this article serves as a framework for demonstrating and prototyping the proposed tracking and super-resolution algorithms, the inherent latency of the signal processing steps is a key issue in HCI and must be addressed. In our research, the largest contributor of latency in our proposed system is the hand-off between the radar device and MATLAB over UDP and shared memory, at an average of 1.93 ms. Rather than streaming to data MATLAB, a real-time solution can be implemented on the TI radar device's built-in DSP, thus providing a more efficient throughput as the DSP has direct access to the samples as they are taken. Additionally, several steps in the signal processing chain will increase in efficiency with an embedded solution. Employing small window sizes, $N_z = 16$ and the number of FFT spatial points is 64, the DPF and FFT computation times can be further reduced compared to the relatively inefficient MATLAB implementation. We would also like to note that a significant decrease in latency was achieved by optimizing the implementation using GPU accelerated coding. A similar approach could be taken on an embedded solution leveraging the highly parallelizable nature of many of the steps in the signal processing chain (FFT, CNN, Gaussian distribution computation). Comparing the computational efficiency among the algorithms, the latency cost for the more robust algorithms is insignificant in proportion to the performance gain, even in the MATLAB implementation. In latency tests, the average response time using the FCNN-DPF was 3.96 ms from user input to MIDI signaling. While most MIDI interfaces outperform this metric, we believe our framework demonstrates a competitive throughput cycle time compared to existing technology and can be further improved by a more efficient implementation.

Hand-tracking using a mmWave radar has both advantages and drawbacks compared with other sensing regimes. In this article, we employ a single radar to develop and demonstrate robust tracking algorithms for mmWave devices. While the best performance is likely achieved through a sensor fusion technique, a radar-based implementation may be optimal if privacy is a concern using optical cameras or issues such as occlusion and lighting conditions must be taken into account. Compared to optical and RGB+D solutions, mmWave is more versatile and reliable, operating well under occlusion, in any temperature or lighting environment, and offers precise depth information of the entire scene. For a musical interface, these advantages may not be often fully realized; however, the novel tracking methods proposed in this article are applicable for many HCI applications. On the other hand, mmWave sensors cannot meet the performance of optical solutions when it comes to cross-range resolution due to the limited aperture size, making multi-object and finger tracking much more challenging. As such, many applications in HCI, computer vision, automated driving, etc. employ radar (and lidar) and optical imaging devices with sensor fusion algorithms to achieve further improved performance at an increased cost. For these applications, our proposed algorithms can aid in sensor fusion by significantly increasing the performance contribution from the radar sensors.

Several alternatives exist to mmWave radar sensing, namely wearable, handheld, and optical devices. Wearable and handheld sensing solutions offer highly precise spatiotemporal features but are often not preferable compared to contactless sensors [29], [30]. In terms of cost, mmWave radar devices are in the same price bracket as the popular Kinect and Leap Motion optical sensors on the order of $\$100 - \$200$. Attempts

using multiple RGB cameras [31], [32] show promising results; however, a single device is much preferred due to the cumbersome nature of multi-camera systems. Single RGB+D solutions have been proposed using generative pose tracking [33], [34] and learning-based generative pose tracking [35], [36]. However, all of these methods suffer tremendously under occlusion or scene clutter, both of which can be overcome using mmWave radar. Some deep learning-oriented solutions have shown quite promising results [37], [38], but constructing a sufficient dataset for meaningful supervised training remains a challenge.

Our proposed interface tracks the 2-D position and velocity of the user's hand to control note selection and two user-selected parameters, a marked improvement over the prior work on mmWave radar using the Google Soli tracking only 1-D range for parameter control [16]. However, optical solutions enable tracking of both hands [3], [7], [9], [33]–[38] or hand and finger position [8], [12] for even finer musical control, with some scenario-specific drawbacks. As radar technology improves and larger apertures become widely available, tracking individual fingers will become increasingly plausible and could yield comparable or superior results to optical solutions due to superior depth resolution.

Compared to prior work on hand-tracking with mmWave devices, our proposed methods yield competitive results. Past work using radar devices achieves, at best, an average range tracking error of 2 cm on human hand localization [17]. Our enhanced tracking technique yields a mean range tracking error of 1.89 mm, improving tracking by more than a factor of ten. In [39], a 4 GHz bandwidth mmWave sensor achieves a 2-D position RMSE of 1.16 mm tracking a thumb, at distances closer than 10 cm. Comparatively, our enhanced tracking technique tracks a human hand across much larger distances and still achieves a competitive 2-D position RMSE of 3.4 mm. At the time of this paper, we are not aware of any other prior work on hand-tracking using mmWave devices. To our knowledge, the system proposed in this paper offers unprecedented hand gesture tracking performance using a single mmWave sensor.

The most direct musical interface comparison to our framework, is the Theremin, as both are controlled by the hand's proximity to the sensor. The pitch of the Theremin is controlled continuously by the hand's vertical location, whereas our interface tracks the range of the hand digitally and selects a note from the user-defined scale. While the Theremin uses two antennas, one for volume control and the other for pitch control, a total of two degrees of freedom, our framework offers three degrees of freedom (range, cross-range, and velocity), thus providing three controllable parameters. As previously mentioned, the musical interface promoted in this article supports Theremin-like gestures for note selection and parameter control. However, high-velocity percussive gestures could be implemented using our high-fidelity tracking algorithms, with some limitations. Small values of the weighting factor, $\boldsymbol{A}$, in the particle filter algorithm can result an excessively smoothed and overly damped system limiting the ability of the system to track sudden movements. Depending on the desired application, finely tuning this parameter is essential for enabling proper gestural control. Our proposed interface is an evolved Theremin, utilizing a modern mmWave sensor for precise tracking in 2-D space (expansion to 3-D can be easily implemented with the proper hardware). In contrast to a Theremin, our musical interface is significantly less effortful in note selection, allowing simple and intuitive inclusion of the additional parameter controls and increasing accessibility to the user-base. One of the authors is a skilled guitar and violin instrumentalist with a background in electronic music production. From the perspective of an experienced musician, the proposed methods offer an elegant new musical interface capable of generating unique phrases previously only possible via manual transcription and provides the musician a sufficient and consistent level of control.

For future work, several promising routes are left to be explored. First, further development can be explored by implementing our proposed methods onto a real-time embedded platform. Additionally, using multiple MIMO radars or a larger MIMO array, a multiple-hand and individual finger tracking interface can be investigated, thus further extending the application space of our robust tracking methods. Finally, our novel super-resolution tracking algorithms can easily be adapted to offer an elegant, efficient solution to a host of acute hand-tracking problems in the HCI domain and even employed in sensor-fusion systems.

VII. Conclusion

Our FCNN-based super-resolution framework successfully demonstrates the viability of acute human hand-tracking for HCI using mmWave sensors. We validated and implemented our spatiotemporal signal processing algorithms and robust tracking algorithms in the form of a contactless musical interface; however, this article also serves to demonstrate the broad effectiveness of mmWave technology for a multitude of near-field acute hand-tracking applications. First, simple feature extraction and tracking methods were introduced, followed by an enhanced approach leveraging the Doppler-corroborated particle filter algorithm and enhancement FCNN to achieve robust tracking and super-resolution in a non-ideal imaging scenario. The methods are compared demonstrating noticeable improvement using the FCNN-DPF over the classical techniques. Additionally, our work offers competitive tracking estimation and localization performance compared to prior methods in the literature for both mmWave and optical implementations. Our entire software implementation and real-time radar interface platform are freely available at request. The novel FCNN-based super-resolution and tracking algorithms presented in this article offer an elegant solution to many contactless HCI problems.

Acknowledgment

The first author's work was supported by the imec USA summer internship program. We would like to extend thanks to Dr. Gonzalo Vaca Castano for his insights in developing the particle filter algorithm and computer vision approach.

REFERENCES

[1] T. Winkler, "Making motion musical: Gesture mapping strategies for interactive computer music," in *Proc. Int. Computer Music Conf.*, Banff, Canada, 1995, pp. 261–264.

[2] K. D. Skeldon, L. M. Reid, V. McInally, B. Dougan, and C. Fulton, "Physics of the theremin," *American Journal of Physics*, vol. 66, no. 11, pp. 945–955, 1998.

[3] R. Polfreman, "Multi-modal instrument: towards a platform for comparative controller evaluation," in *Proc. Int. Computer Music Conf.* Proc. Int. Computer Music Conf., July 2011, pp. 147–150. [Online]. Available: https://eprints.soton.ac.uk/353226/

[4] S. Trail, M. Dean, G. Odowichuk, T. F. Tavares, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect." in *Proc. Int. Conf. New Interfaces for Musical Expression*, 2012.

[5] S. Sentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg, "Crossole: A gestural interface for composition, improvisation and performance using kinect." in *Proc. Int. Conf. New Interfaces for Musical Expression*, 2012.

[6] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Trans. on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[7] A. R. Jensenius, "Kinectofon: Performing with shapes in planes," in *Proc. Int. Conf. on New Interfaces for Musical Expression*, Daejeon, Korea, 2013, pp. 196–197.

[8] J. Han and N. Gold, "Lessons learned in exploring the leap motion™ sensor for gesture-based instrument design," in *Proc. Int. Conf. on New Interfaces for Musical Expression*. London, United Kingdom: Goldsmiths University of London, 2014, pp. 371–374.

[9] L. Hantrakul and K. Kaczmarek, "Implementations of the leap motion in sound synthesis, effects modulation and assistive performance tools." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, London, United Kingdom, 2014.

[10] D. Brown, N. Renney, A. Stark, C. Nash, and T. Mitchell, "Leimu: Gloveless music interaction using a wrist mounted leap motion," in *Proc. Int. Conf. on New Interfaces for Musical Expression*, Brisbane, Austrailia, 2016, pp. 300–304.

[11] A. Tindale, A. Kapur, and G. Tzanetakis, "Training surrogate sensors in musical gesture acquisition systems," *IEEE Trans. on Multimedia*, vol. 13, no. 1, pp. 50–59, 2011.

[12] O. Nieto and D. Shasha, "Hand gesture recognition in mobile devices: Enhancing the musical experience," *Proc. Computer Music Multidisciplinary Research*, vol. 13, 2013.

[13] M. Akbari and H. Cheng, "Real-time piano music transcription based on computer vision," *IEEE Trans. on Multimedia*, vol. 17, no. 12, pp. 2113–2121, 2015.

[14] J. W. Smith, S. Thiagarajan, R. Willis, Y. Makris, and M. Torlak, "Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique," *IEEE Access*, vol. 9, pp. 10 893–10 902, 2021.

[15] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3041–3048, 2019.

[16] F. Bernardo, N. Arner, and P. Batchelor, "O soli mio: exploring millimeter wave radar for musical interaction." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, vol. 17, 2017, pp. 283–286.

[17] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using rf backscatter," in *Proc. USENIX Symposium on Networked Systems Design and Implementation*, 2015, pp. 189–204.

[18] Y. Dai, T. Jin, Y. Song, H. Du, and D. Zhao, "Cnn-based multiple-input multiple-output radar image enhancement method," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6840–6844, 2019.

[19] Y. Sun, X. Liang, H. Fan, M. Imran, and H. Heidari, "Visual hand tracking on depth image using 2-d matched filter," in *Proc. UK/China Emerging Technologies*, Aug. 21-22, 2019, Glasgow, United Kingdom, pp. 1–4.

[20] S. Rao, "Intro to mmwave sensing : Fmcw radars," Jul 2020. [Online]. Available: https://training.ti.com/node/1139153

[21] J. W. Smith, M. E. Yanik, and M. Torlak, "Near-field mimo-isar millimeter-wave imaging," in *Proc. IEEE Radar Conf.*, 2020, pp. 1–6.

[22] M. E. Yanik and M. Torlak, "Near-field mimo-sar millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31 801–31 819, 2019.

[23] M. E. Yanik, D. Wang, and M. Torlak, "Development and demonstration of mimo-sar mmwave imaging testbeds," *IEEE Access*, vol. 8, pp. 126 019–126 038, 2020.

[24] V. Winkler, "Range doppler detection for automotive fmcw radars," in *Proc. European Radar Conf.*, Oct. 10-12, 2007, Munich, Germany, pp. 166–169.

[25] J. Kim, J. Chun, and S. Song, "Joint range and angle estimation for fmcw mimo radar and its application," *arXiv:1811.06715*, 2018.

[26] J. García, A. Gardel, I. Bravo, J. L. Lázaro, and M. Martínez, "Tracking people motion based on extended condensation algorithm," *IEEE Trans. on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 606–618, 2013.

[27] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 35–39, 2019.

[28] "Texas instruments mmwave studio." [Online]. Available: https://www.ti.com/tool/MMWAVE-STUDIO

[29] L. Pardue and W. Sebastian, "Hand-controller for combined tactile control and motion tracking." in *Proc. Int. Conf. on New Interfaces for Musical Expression*, 2013, pp. 90–93.

[30] P. Neto, J. N. Pires, and A. P. Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *Industrial Robot*, vol. 37, no. 2, pp. 137–147, 2010.

[31] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *Proc. European Conf. on Computer Vision*. Springer, 2012, pp. 640–653.

[32] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proc. IEEE Int. Conf. on Computer Vision*, 2013, pp. 2456–2463.

[33] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect." in *Proc. Brit. Mach. Vision Conf*, vol. 1, no. 2, 2011, pp. 101.1–101.11.

[34] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proc. IEEE Int. Conf. on Computer Vision*, 2015, pp. 3325–3333.

[35] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2015, pp. 3213–3221.

[36] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. on Graphics*, vol. 35, no. 4, pp. 1–12, 2016.

[37] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. on Graphics*, vol. 33, no. 5, pp. 1–10, 2014.

[38] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *Proc. European Conf. on Computer Vision*. Springer, 2016, pp. 346–361.

[39] Z. Li, Z. Lei, A. Yan, E. Solovey, and K. Pahlavan, "Thumouse: A micro-gesture cursor input through mmwave radar-based interaction," in *Proc. IEEE Int. Conf. on Consumer Electronics*, Jan. 4-6, 2020, Las Vegas, NV, USA, pp. 1–9.

Radar Musical Instrument - A Spatiotemporal Real-Time mmWave Sensor for Contactless Human-Computer Interaction

Josiah Smith[1], Orges Furxhi[2], and Murat Torlak[1]

[1]Dept. of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX, United States
[2]Computational Imaging, imec USA, FL, United States

***Abstract*—Millimeter-wave (mmWave) radar sensing is transforming many applications that have traditionally required different modes of sensing, as exemplified by self-driving cars, vital signs monitoring, fall detection, occupancy detection, and many more. Human-computer interaction (HCI) can benefit from the use of mmWave radars because of the fine depth and cross-range resolution of such devices, enabling accurate tracking of user-performed actions in space. Additionally, signatures embedded in the return signal from frequency-modulated continuous-wave (FMCW) radars contain spatiotemporal features that can be extracted in real time to perform accurate position and pose tracking for HCI. In this paper, we propose and demonstrate a novel real-time mmWave interface that leverages spatiotemporal information from a multiple-input-multiple-output (MIMO)-FMCW radar to create a new musical interface (NMI) controllable by specific hand positions and motions. After constructing the necessary real-time framework, a simple signal processing chain and feature extraction method is presented and subsequently extended to an enhanced tracking technique employing novel localization algorithms and deep-learning-based spatiotemporal enhancement. The novel system we propose in this paper allows for real-time human-computer interaction to create a new musical interface controlled solely by the precise tracking of the musician's hand.**

***Index Terms*—millimeter-wave (mmWave), multiple-input multiple-output (MIMO), human-computer interaction (HCI), radar perception, new musical interface (NMI), deep learning, fully-convolutional neural network (FCNN)**

I. Introduction

Radar perception for human-computer interaction on multiple-input-multiple-output (MIMO) frequency modulated continuous wave (FMCW) millimeter-wave (mmWave) radars has emerged as a promising solution to a variety of sensing problems. The physical nature of millimeter-waves offers a safe method for high-resolution imaging where optical sensors may fail due to insufficient lighting, fog, or other line-of-sight interference. Additionally, ultra-wideband FMCW devices enable depth resolution on the order of centimeters and compact MIMO radars allow for task-enabling cross-range resolutions on a small scale. As a result, precise spatial information of a target scene can be easily acquired from such imaging devices at a low cost.

mmWave sensing solutions are relatively modern technology, but some of the earliest electronic interfaces, for any application, were contactless devices for physically expressive musical control including the Radio Drum and Theremin [1]. Russian physicist Leon Theremin demonstrated his noncontact musical instrument in 1921, an interface controlled by the proximity of the musician's hand to an antenna using beat-frequency oscillators and a capacitive sensing apparatus [2]. More recently, computer vision approaches have been adopted for the innovation of contactless new musical interfaces (NMIs), most of which rely on optical camera solutions. Extensive prior work exists on optical-based NMIs using popular sensors such as the Microsoft Kinect and Leap Motion. Polfreman uses the Kinect to track the 3-D position of both hands of a standing performer to construct a multi-modal instrument [3]. Trail *et al.* present a pitched percussion hyperinstrument to track the tips of two mallets simultaneously with the Kinect [4]. Crossole, designed by Senturk *et al.*, is a Kinect-based metainstrument visualizing chord progressions as virtual blocks resembling a crossword puzzle [5]. Schramm *et al.* use the Kinect to analyze and classify motions of a orchestral conductor [6]. In [7], the Kinect is used to track hand motion across time and then translated to music using the inverse Fourier transform of the physical pattern using a sonification technique called sonomotiongram. Alternatively, the Leap Motion controller is capable of modeling the entire hand, including the fingers, which allows for even more detailed hand posture-based gesture control to be explored for musical interface development. Using the Leap Motion sensor, Han *et al.* developed two NMIs, *Air Keys* and *Air Pad*. *Air Keys* tracks the motion and position of each finger in the hand to recognize when and which keys the musician is pressing and playing the desired notes. Similarly, *Air Pad* tracks the hand position to create a 2-D virtual drum pad played by pressing specific regions in a 2-D horizontal plane, thus requiring accurate 3-D hand tracking [8]. Hantrakul and Kaczmarek use the Leap Motion controller to track both hands for controlling MIDI instruments and virtual effects [9]. Similarly, Leimu pairs the Leap Motion with an inertial measurement unit (IMU) demonstrating improved performance over the Leap Motion controller alone for musical interface [10]. Other solutions have been attempted, such as employing non-invasive force sensing resistors to enhance "traditional" by learning and monitoring for gestures performed by the musician [11]. In [12], a musical interface is constructed using exclusively a portable RGB camera to recognize hand gestures. Akbari and Cheng

developed a system to transcribe music played on a piano in real-time using optical cameras positioned to view the keys [13]. These projects have yielded high performing real-time musical interfaces capable of consistent high-accuracy motion tracking but require several key design constraints, namely specific lighting conditions and line-of-sight. As shown in this article, mmWave sensors overcome these major obstacles and demonstrate the broad application space of mmWave imagers and advanced spatiotemporal algorithms. However, little work has been done towards gestural musical interfaces on mmWave radar sensors. Even though extensive research exists on static and dynamic gesture recognition using mmWave radars [14], [15], Google ATAP's Project Soli is the only effort using mmWave radar for musical interface, using gesture recognition and 1-D position estimation to control the parameters of audio synthesizers [16].

The novel Radar Musical Instrument presented in this paper offers a major advancement for near-field mmWave hand tracking and an accessible MATLAB software platform for further investigation into real-time mmWave HCI and algorithm innovation. 2-D localization performance is considerably improved from past work [17] by employing a novel deep learning-based image enhancement technique. Additionally, a spatiotemporal tracking algorithm is presented with a novel Doppler-corroboration extension further improving tracking robustness. Compared to prior work on gesture tracking using optical solutions [3], [7]–[9], [12], [18] or optical-mmWave [19] sensor fusion, our approach addresses the issue of hand gesture tracking using a singular mmWave sensor. The methods presented in this work offer a novel approach to gesture tracking by fusing spatiotemporal algorithms, deep learning-enhanced feature extraction, and robust position tracking algorithms. To aid further development and prototyping for real-time mmWave gesture applications, the entire software implementation is being made available by request. To our knowledge, the Radar Musical Instrument is the first openly available software package supporting real-time data streaming from a mmWave radar into MATLAB for streamlined signal processing and deep learning algorithm development.

The rest of this paper is formatted as follows. In section II, a brief overview of relevant music theory is provided for a requisite understanding of the applicable musical topics discussed throughout the paper. Section III provides an overview of the MIMO-FMCW radar signal model and feature extraction methods. In section IV, the novel Radar Musical Instrument is introduced, presenting two robust tracking algorithms and estimation techniques, the results of which are shown in section V. Section VI contains a discussion of the performance, design constraints, and distinct advantages of the two tracking methods in sections IV-B and IV-C, followed finally by conclusions.

II. Music Theory for Engineers

A. Notes, Pitch, Frets, and Vibrato

The fundamental unit of music is the note and every instrument and interface requires the musician to input or select the note differently. Notes are primarily distinguished by the human brain according to their pitch, which is simply the fundamental frequency of the acoustic wave, notated with an alphabetical letter from A to G. In standard Western equal temperament, the A above middle C, also called A4, has been assigned the fundamental frequency of 440Hz. Interestingly, humans perceive musical intervals as an approximately logarithmic relation to fundamental frequency. The interval between 440Hz and 880Hz sounds identical to the interval between 220Hz and 440Hz. As a result, Western music theory dictates that the frequencies of the standard equally tempered notes are distributed exponentially [20]. Thus, the pitch (frequency) of each note can be found by the following relation,

$$f = f_0 * 2^{n/m}, \tag{1}$$

where f_0 is the reference frequency relative to which one wishes to find a note separated by n semitones and m is the number of equal-tempered semitones in an octave [21]. In the standard Western equal temperament $m = 12$, as there are 12 semitones in each octave. For example, if one desires to find the fundamental frequency of F4, using A4 as the reference, $f_0 = 440$Hz (the fundamental frequency of A4) and $n = -4$ (F4 is four semitones below A4). Using (1), the fundamental frequency of F4 is found to be 349.23Hz.

When it comes to how a note is selected, there are generally two types of instruments, "non-fretted" and "fretted", specifically within the stringed instrument family. Violins are an example of a non-fretted instrument as the musician selects notes by pressing down along the smooth neck. Whereas the violin is capable of playing any note on a continuous spectrum, playing the desired note requires a significant level of skill. On the contrary, the guitar has metal implants called frets that aid in note selection. On a fretted instrument, when the musician presses down near a fret, the effective length of the string is dictated by the metal fret rather than by the musician's finger, resulting in quantization applied to the user input. The concept behind frets is useful for effectively quantizing the musician's input into predefined regions allowing for improved intonation in the presence of imperfect musicianship, which can be viewed as noise. Playing the desired frequency is essential to generating music; however, a technique called "vibrato" produces an audibly pleasing phenomenon by modulating the pitch slightly above and below the center frequency, often heard at the end of vocal phrases.

B. Modern Virtual Instruments

With the rise of personal computers, most modern digital musical instruments are purely virtual. Entire platforms exist for musicians to control numerous parameters, sequence a plethora of sound-bytes, and switch between virtual instruments in real-time. Most virtual instruments rely on the MIDI standard to connect to musical interfaces. As such, a virtual instrument can be easily obtained and loaded into a digital audio workstation (DAW) or live audio platform and can be controlled by a vast number of different interfaces.

The novel Radar Musical Instrument developed in this paper will rely heavily on the music theory as discussed in this

section for note selection, pitch and vibrato generation, and MIDI interface to leverage the features that can be extracted from the mmWave radar return signal. As such, the musical instrument interface detailed in this paper will focus primarily on providing the MIDI signals to a virtual instrument, allowing the virtual instrument to handle the intricacies of the audio output and instrument synthesis. However, a custom real-time audio output tool is also developed along with the musical instrument interface for demonstration and validation. In this manner, the proposed novel Radar Musical Instrument demonstrates the viability of mmWave imaging systems for real-time acute human gesture recognition and computer vision.

III. Radar Theory for Musicians

In this section, we overview the propagation model for the MIMO-FMCW radar chirp signal and examine the spatiotemporal features of a target in motion. The geometry of the Radar Musical Instrument, as shown in Fig. 1, consists of a multistatic linear MIMO array facing vertically. Throughout this paper, the musician's hand is modeled as a point reflector located at the point (y, z). Given the range and cross-range resolution limitations of the radar sensor, dictated by the chirp bandwidth, beam pattern, element locations, and the number of virtual elements, approximating the human hand as a point reflector holds valid for most hand sizes.

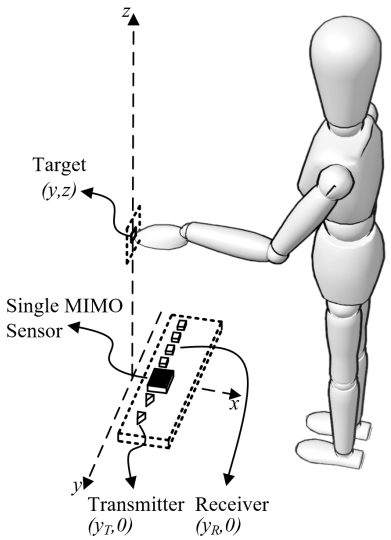


Fig. 1. The geometry of the Radar Musical Instrument, where the linear MIMO array faces vertically and the musician moves their hand throughout the y-z plane.

A. MIMO-FMCW Signal Model

The FMCW chirp signal model is well documented in literature [22] and is described here for reference and continuity throughout this paper. Considering a single transceiver pair located at $(y_T, 0)$ and $(y_R, 0)$in the y-z plane, respectively, the FMCW IF signal or beat signal can be expressed as [23]

$$s(y_T, y_R, k) = \frac{p}{R_T R_R} e^{jk(R_T + R_R)}, \tag{2}$$

where R_T is the distance from the transmitter to the point target, R_R is the distance from the receiver to the point target, $k = \frac{2\pi f}{c}$ is the instantaneous wavenumber of the frequency $f = f_0 + Kt$, f_0 is the carrier frequency at time $t = 0$, K is the chirp slope, and T is the chirp duration in seconds.

(2) shows the multistatic beat signal of a given target, but the monostatic equivalent is much more convenient to work with in the later signal processing steps. Using (y_0, z_0) as a reference point at the center of the target domain, the received multistatic beat signal can be converted to its corresponding monostatic equivalent using the approximation developed in [24] as

$$\hat{s}(y', k) = s(y_T, y_R, k) e^{-jk\frac{d_y^2}{4z_0}}, \tag{3}$$

valid only for small d_y, the distance between the transmitter and receiver elements. Taking y' as the locations of the virtual elements located at the midpoints between each transceiver pair and R as the corresponding distance from each virtual element to the point reflector, the resulting monostatic beat signal approximates to

$$\hat{s}(y', k) \approx \frac{p}{R^2} e^{j2kR}. \tag{4}$$

From (4), the radial distance from the radar to the target can already be identified as the frequency of the beat signal.

B. FMCW Doppler Radar

Under the monostatic radar regime, whose beat signal is shown in (4), the Doppler effect can be observed by transmitting several chirps at a known pulse repetition interval (PRI), T_{PRI} and tracking the phase change. Now, we extend the model from a stationary point reflector to a target in constant radial velocity. As discussed in detail in [25], the resulting beat signal can be expressed as

$$\hat{s}(y, k, n_c) = \frac{p}{R^2} e^{j(2kR + \frac{4\pi v T_{PRI}}{\lambda_0} n_c)}, \tag{5}$$

where R is the initial range of the target, v is target's constant velocity, λ_0 is the wavelength of the starting frequency of the FMCW chirp f_0, and n_c is the chirp index,

From this equation, it is obvious that the beat signal is a 2-D complex sinusoidal with frequencies corresponding to the range and velocity of the target on the first and second dimensions, respectively. Subsequently, the traditional method for identifying the target's range and velocity consists of gathering the slow-time series data into a time vs. slow-time data matrix and performing an efficient fast Fourier transform (FFT).

C. Feature Extraction Methods

Now, after the multistatic-to-monostatic conversion in (3), the approximate monostatic signal can be processed to extract spatiotemporal features used to generate music based on the unique characteristics of the hand's position and temporal qualities.

1) Range Migration Algorithm: Taking the monostatic received signal (4), the 2-D position and the velocity of the single point target are now estimated. Whereas traditional methods exist to estimate the range and angle of a target via a 2-D fast Fourier transform (FFT) and simple maximum finding [26], position estimation accuracy is quite low [27]. Instead, we will use a Doppler velocity preserving range migration algorithm (RMA) approach to reconstruct a 2-D range vs. cross-range image of the target scene, from which spatial and temporal information can be extracted.

The primary goal of the RMA is to reconstruct the target scene's reflectivity $p(y, z)$ function. Neglecting the amplitude terms in (4), the approximated monostatic beat signal is the superposition of the backscattered echo signal at every point in the scene, as modeled by

$$\hat{s}(y', k) = \iint p(y, z) e^{j2kR} dy dz. \tag{6}$$

Inverting this equation and simplifying using the method of stationary phase, the reflectivity function $p(y, z)$ can be estimated efficiently by

$$\hat{p}(y, z) = IFT_{2D}^{(k_y, k_z)} \left[\text{STOLT} \left[FT_{1D}^{(y)} [\hat{s}(y', k)] \right] \right], \tag{7}$$

where STOLT[.] is the Stolt interpolation operation and FT and IFT are the forward and backward Fourier transform operators.

2) Doppler Range Migration Algorithm: As discussed above, Doppler velocity information of a target can be estimated by performing an FFT across the slow-time (chirp) dimension of the radar return data; however, the same Doppler velocity information contained in the phase of the return signal is preserved in the RMA algorithm. Notice (5) contains the same beat signal as (4) with an additional, disjoint, frequency corresponding the target velocity. As a result, equivalent techniques can be employed to extract Doppler velocity from the RMA images.

For the signal model above, N_c chirps are captured and stored in for target velocity estimation as $\hat{s}(y', k, n_c)$, where n_c is the chirp index. Once the RMA is performed for each chirp, the image is cropped to include only the region of interest (ROI), wherein we expect to find the hand, and whose dimensions are N_y x N_z. The stored data matrix contains the complex RMA image for each chirp as $\hat{p}(y, z, n_c)$. For the sake of computational efficiency, the cross-range location is estimated and used to narrow the data to two dimensions as $\hat{p}(z, n_c)$. Next, the Doppler FFT is taken across the slow-time dimension yielding the range-Doppler map $d(z, n_d)$ [26], where n_d is the Doppler index corresponding to the Doppler velocities between $[-\frac{\lambda_0}{4T_{PRI}}, \frac{\lambda_0}{4T_{PRI}}]$, T_{PRI} is the chirp periodicity or PRI, and $\hat{y}_p$ is the estimated cross-range position of the target, as shown in (8),

$$d(z, n_d) = FT_{1D}^{(n_c)} \left[\hat{p}(y, z, n_c) \Big|_{y=\hat{y}_p} \right]. \tag{8}$$

IV. The Radar Musical Instrument

In this section, we present the novel Radar Musical Instrument, a mmWave imaging system capable of high accuracy hand tracking for human-computer interaction. The success of the Radar Musical Instrument both demonstrates the specific application of mmWave radar spatiotemporal algorithms for audio signal generation and verifies the legitimacy of such imaging systems for a host of human-computer interaction and computer vision applications.

A. Hardware and Software Setup

The hardware employed in the proposed system consists of a Texas Instruments (TI) automotive radar in conjunction with a real-time data capture adapter. In this research, signal processing, visualization, and machine learning are performed in real-time on a personal computer (PC) in MATLAB. Once the algorithms are validated on a PC, they can be implemented onto such embedded devices for application-specific usage; however, this paper serves as a broad demonstration of the viability of mmWave sensors for real-time computer vision and radar perception applications.

The TI radar is a multistatic MIMO FMCW mmWave radar with an operating bandwidth of 4GHz and a center frequency of 79GHz. In this research, we will be utilizing its linear MIMO array consisting of 2 transmit antenna (TX) elements, separated by $2\lambda_c$, and 4 receive antenna (RX) elements, separated by $\lambda_c/2$, resulting in a virtual array of 8 equally spaced virtual elements separated by $\lambda_c/4$, where λ_c is the wavelength corresponding to the center frequency [26]. Before collecting any meaningful data, the calibration methodology discussed in [24] is adopted to mitigate range bias, constant phase errors, and instrumentation delay.

1) Real-Time Data Retrieval and MATLAB Interface: First, a real-time data reading routine is developed to interface with the data-capture card over its UDP interface. The Radar Musical Instrument requires a custom solution capable of a.) receiving the sequential UDP packets, b.) organizing the packets into entire chirp or frames (sequence of chirps), and c.) providing the data to MATLAB in real-time for signal processing and deep learning.

A custom real-time MATLAB graphical user interface (GUI) is written to serve as the single user interface required for any musician to easily use the Radar Musical Instrument. The MATLAB GUI interfaces with the TI mmWave Studio software [28] to set up the radar board and data capture adapter, and then initializes the UDP interface software to properly read in each packet and organize them into frames based on the user-specified radar parameters. While the radar is continuously capturing responses of the target scene, the beat signal is read into the data retrieval software and is delivered to MATLAB. Even though MATLAB does not offer tremendous computational efficiency, it is fast enough to read in frames at a relatively high rate (50Hz-250Hz). The GUI is capable of extracting and visualizing the range, cross-range, Doppler velocity, and cross-range oscillation in real-time. And finally, the GUI enables precise mapping from the dynamic gestures of the musician to audible notes and a MIDI interface. The custom MATLAB GUI functions either as an end-to-end instrument providing the entire interface, signal processing, and audio output functionality or as a MIDI interface enabling

portability and continuity among a multitude of applications and musical venues.

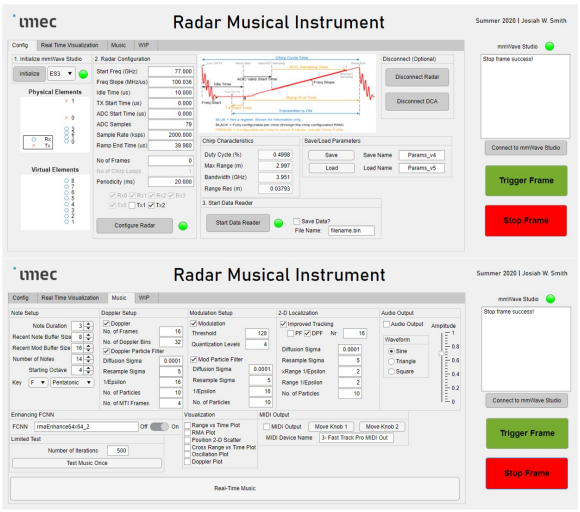


Fig. 2. Radar Musical Instrument custom MATLAB GUI: device setup and music generation pages.

B. Simple Gesture Tracking

This section presents the fundamentals of the Radar Musical Instrument. In its most basic form, the Radar Musical Instrument monitors the spatial and temporal signatures of the musician's hand as it is moved throughout the scene converting these features to music. The three parameters controlling the Radar Musical Instrument audio output are range, cross-range oscillation, and Doppler velocity.

For localization, the RMA is used to reconstruct a 2-D image of the hand in the y-z plane, as discussed in section III-C1. Once the image is reconstructed and cropped to the user-specified region of interest, a simple peak finding routine locates the peak and estimates the 2-D position on a discrete y-z grid. This type of image reconstruction and peak finding suffers from several key factors. First, due to limited bandwidth and the number of antennas, the range and cross-range resolution are on the order of centimeters, even without the presence of noise. Additionally, the RMA assumes an ideal beam pattern from a monostatic full-duplex array, but the physical setup consists of antenna elements with varying, non-ideal near-field beam patterns and a multistatic MIMO array. Whereas the multistatic effects can be somewhat compensated for using the phase correction discussed in section III-A, noticeable image degradation occurs at some positions due to the RMA's monostatic array assumption. These challenges will be further addressed in section IV-C.

Despite the non-ideal imaging environment, the 2-D position of the hand can be estimated within several centimeters. Then the range and cross-range positions are stored in circular buffers. The real-time position estimate is used to control musical output. To select a note, the musician moves their hand vertically through the region of interest. Instead of selecting from a continuous range of frequencies, "virtual frets" are implemented to quantize the range into subregions corresponding to predefined notes. The note selection and virtual fret parameters are all customizable in the Radar Musical Instrument GUI.

Next, as the cross-range location is stored, its oscillation rate is extracted. Due to possible timing issues in the data retrieval tool discussed previously, a non-uniform fast Fourier transform (NUFFT) is performed is used to extract the oscillation rate. Similarly, a NUFFT is implemented to extract the Doppler velocity as discussed in section III-C2.

These temporal features can serve several purposes for musical output. Using the built-in audio output tool, the cross-range oscillation rate is used to control the vibrato specified to each note. If the musician is moving their hand back and forth along the y axis, the Radar Musical Instrument plays the desired note with a vibrato effect at the same rate as the hand. Simply, the frequency of the current note is modulated several Hertz above and below the actual fundamental frequency of the note in a sinusoidal fashion.

Alternatively, the MATLAB GUI is capable of connecting to a virtual instrument via MIDI using the range, cross-range oscillation, and Doppler velocity to control the instrument. In this manner, the Radar Musical Instrument acts as a musical interface controlling the software-based instrument similar to a MIDI keyboard. As with the audio output tool, the MIDI output tool discretizes the range into regions corresponding to MIDI notes (whose possible values range from 0 to 127). Now, the cross-range oscillation rate and Doppler velocity are treated as MIDI controller knobs. The musician can program these features to control any tunable parameter within the virtual instrument enabling novel music generation techniques and dynamic control of a MIDI instrument unique to the Radar Musical Instrument. As a result, precise non-contact gesture control allows artists to generate musical sequences only attainable in real-time using the Radar Musical Instrument.

The entire signal processing chain of the Radar Musical Instrument from the input echo signal to the musical features is shown in Fig. 3. The echo signal is read into MATLAB where the preprocessing discussed in this section is performed and the user inputs are converted into audio or MIDI output by extracting the spatiotemporal features (range, cross-range oscillation, and Doppler velocity) in real-time. In the next section, these three features are called the new noisy measurements or the new noisy measurement vector $\boldsymbol{z}$.

In the simple gesture tracking method, range, cross-range, Doppler velocity, and cross-range oscillation frequency are extracted from the FMCW beat signal using traditional techniques and used to control the musical instrument. The simple tracking techniques discussed in this section allow tracking of spatial and temporal features; however, non-idealities due to noise and RMA assumptions degrade RMA image quality and subsequent spatiotemporal tracking performance. We will next demonstrate an improved tracking technique based on a modified particle filter algorithm and deep learning-aided feature extraction method.

C. Enhanced Gesture Tracking

In this section, we improve the capabilities of the Radar Musical Instrument from the simple tracking techniques for

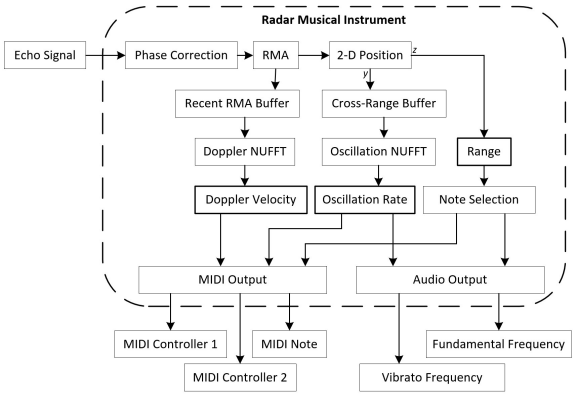


Fig. 3. Radar Musical Instrument signal processing chain: converting the echo signal input to audio or MIDI output.

spatiotemporal tracking and RMA image enhancement to overcome foundational non-idealities in the imaging scenario and inconsistencies in the user input. The concepts demonstrated in this section are applicable novelties for many tracking and high-resolution imaging applications far beyond the scope of HCI.

1) Improved Spatial and Temporal Tracking: We proceed to improve the Radar Musical Instrument's spatiotemporal tracking robustness by utilizing a modified particle filter algorithm [29]. As an example, we begin by employing a particle filter for 2-D spatial localization. A key assumption in our modified model is that the musician's hand is likely to remain stationary, rather than assuming some constant velocity or known deterministic motion model; we will refer to this assumption as the semi-stationary assumption. As a result, our tracking methodology yields highly consistent and smooth localization while finely tracking the hand location throughout the region of interest, but does suffer performance degradation at high hand velocities.

Algorithm 1: Modified Particle Filter Algorithm

randomize n particle states $\boldsymbol{x}_{t-1}$ throughout ROI;
initialize n uniform weights $\boldsymbol{w}_{t-1}$;
while *true* **do**
  retieve beat signal $s(y,k)$;
  extract measurement vector $\boldsymbol{z}$;
  sample particle states $\boldsymbol{x}_{t-1}$, with replacement, using weights $\boldsymbol{w}_{t-1}$ to obtain $\boldsymbol{x}_t$;
  resample particles $\boldsymbol{x}_t = \boldsymbol{x}_t + \boldsymbol{\epsilon}(\boldsymbol{z}_t - \boldsymbol{s}_{t-1}) + \boldsymbol{\psi}$, where $p(\boldsymbol{\psi}) \sim N(\mathbf{0}, \Sigma_\psi)$;
  compute weights $\boldsymbol{w}_t$ by $p(\boldsymbol{z}_t|\boldsymbol{x}_t) \sim N(\boldsymbol{s}_{t-1}, \boldsymbol{\Sigma}_w)$;
  normalize weights $\sum_n \boldsymbol{w}_t^{(n)} = 1$;
  estimate $\boldsymbol{s}_t = \sum_n \boldsymbol{w}_t^{(n)} \boldsymbol{x}_t^{(n)}$;
end

Under the semi-stationary assumption, instead of having a deterministic control input, the effective control input is a weighted movement towards the newest measurement. For 2-D localization, the new noisy measurement, stored in the vector $\boldsymbol{z}$, is made by performing the RMA and locating the peak. $\boldsymbol{z}$ has two elements, the newest estimates of the range, and cross-range. Algorithm 1 details the modified particle filter implementation, using $\boldsymbol{x}$ as the vector containing particle locations in the 2-D plane, $\boldsymbol{w}$ as the vector of weights corresponding to each particle, $\boldsymbol{z}$ as the measurement vector taken at each time step by extracting the features from the beat signal $s(y,k)$, $\boldsymbol{s}$ as the estimated state vector, and $N(\boldsymbol{\mu}, \boldsymbol{\Sigma})$ as the multivariate Gaussian distribution with mean vector $\boldsymbol{\mu}$ and covariance matrix $\boldsymbol{\Sigma}$.

Proper handling of the key steps, 1) resampling of the particle states and 2) computing new weights, is essential to effectively implementing the semi-stationary modification to the particle filter algorithm.

The particle resampling process involves moving the particles towards the new measurement by a specified weight. $\boldsymbol{\epsilon}$ is a diagonal matrix whose elements weight the importance of each new noisy measurement for the corresponding feature. In this way, the new measurements do not dominate the motion tracking but are included in the localization procedure while bypassing the requirement of a motion model. Fig. 4 demonstrates the resampling process with $\boldsymbol{\epsilon} = \frac{1}{2}\boldsymbol{I}$, where $\boldsymbol{I}$ is the identity matrix. Note that before computing the new weights, particle diffusion is performed by adding the zero-mean Gaussian noise term $\boldsymbol{\psi}$.

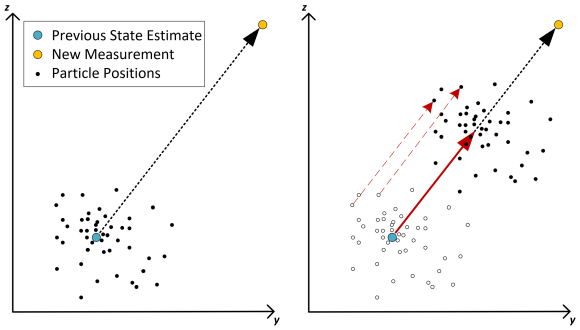


Fig. 4. A visual example of the modified particle filter algorithm resampling process. The particle locations are resampled by a shift transformation towards the new measurement according to the diagonal elements of the weight matrix $\boldsymbol{\epsilon} = \frac{1}{2}\boldsymbol{I}$.

The new weights are computed from the multivariate Gaussian distribution with $\boldsymbol{s}_{t-1}$, the previously estimated states, as the mean vector and a predefined covariance matrix. Therefore, particles closer to the previously estimated state have a higher weight than those farther away. This results in a tendency towards small changes in the state estimations while monitoring for movement from the current position. For many applications requiring precise and consistent localization and motion tracking, our modified particle filter algorithm is an ideal fit as it tends to a steady-state estimation of the states but remains active in monitoring the noisy sensor input.

Just as the 2-D position can be robustly tracked with our modified particle filter algorithm, an identical implementation can be used to estimate and track the velocity of the target and its cross-range oscillation rate as well, both of which are utilized further in the development of the Radar Musical Instrument.

2) Doppler-Corroborated Real-Time Weighting: The resampling and new weight calculation processes result in a similar technique to the Kalman filter with several advantages. It can easily track non-linear dynamics and does not require prior knowledge of the motion model or noise parameters. Now, we extend the constant weighting method to a real-time weighting technique specific for the range domain. Our approach considers corroboration between the Doppler velocity estimate and the velocity estimated from the range samples as a measure of the new measurement's reliability. Thus, the dependability of the Doppler velocity can improve tracking of the target velocity along the range (z) dimension even in the presence of noisy position estimates. First, a Doppler NUFFT is performed across the N_r recent complex RMA images and the Doppler velocity ($\hat{v}_d$) is estimated after video pulse integration by

$$\hat{v}_d = \arg\max \sqrt{\int |\tilde{d}(z, n_d)|^2 dz}, \tag{9}$$

where $\tilde{d}(z, n_d)$ is the recent range-Doppler map as discussed previously. Then, the new measurement is used to calculate the sample velocity ($\hat{v}_s$) based on the recent range locations using the least squares estimator as

$$\hat{v}_s = \frac{N_r \sum_i (\tilde{r}_i t_i) - \sum_i \tilde{r}_i \sum_i t_i}{N_r \sum_i (\tilde{r}_i^2) - \left(\sum_i \tilde{r}_i\right)^2}, \tag{10}$$

$$t_i = 0, T_{PRI}, \ldots, (i-1)T_{PRI}, \ldots, (N_r - 1)T_{PRI}, \tag{11}$$

where $\tilde{r}_i$ is the buffer of recent range estimates. Note that the new range measurement is stored at the end of the recent range estimation buffer, $\tilde{r}_{N_r}$, and all the other elements are the previous estimations by the Doppler-corroborated modified particle filter algorithm.

Now, the difference between the Doppler estimated velocity and sample estimated velocity is computed as $\Delta v = |\hat{v}_d - \hat{v}_r|$ and used in the reward function below to update the weight placed on the new noisy measurement in real-time.

$$\epsilon_r(\Delta v) = \begin{cases} \epsilon_0 \cos\left(\frac{2\pi T_{PRI} \Delta v}{\lambda_0}\right) & \text{if } \Delta v \leq \frac{\lambda_0}{4T_{PRI}} \\ 0 & \text{if } \Delta v > \frac{\lambda_0}{4T_{PRI}} \end{cases} \tag{12}$$

When the sample velocity is close to Doppler velocity, Δv is quite small and the reward function is close to ϵ_0. In this way, the new measurement is corroborated with the more reliable Doppler velocity and weighted accordingly. Outliers and erroneous measurements contradicting the Doppler velocity are given less importance during the particle resampling process.

Now, the modified particle filter algorithm can be extended to include Doppler corroboration of the new measurement. Simply, the range resampling weight ϵ_r is computed by (12) and included in the weighting matrix $\boldsymbol{\epsilon}$.

3) Improved 2-D Position Estimation by Enhancing FCNN: Whereas the Doppler-corroborated modified particle filter algorithm improves the tracking consistency and smoothness, non-idealities such as instrumentation delay, ambient/device noise, multistatic effects, and non-spherical beam patterns remain unaddressed. To solve these issues, we present a fully convolutional neural network (FCNN) for image enhancement that improves the 2-D position estimation and subsequent tracking accuracy. Whereas prior CNN techniques are typically based on far-field assumptions and applied in SAR scenarios with large virtual aperture topologies [30], our enhancement FCNN method operates on near-field images, improves localization even with a small virtual aperture of eight elements, and is trained using a novel technique allowing the network to learn the environment and device noise, near-field beam pattern, and multistatic effects.

The architecture adopted in this paper for the enhancement FCNN is shown in Fig. 5. Four convolution layers of decreasing kernel size are each followed by a nonlinear Rectified Linear Unit (ReLU) layer. Each convolutional layer is zero-padded such that its output is the same size as the input.

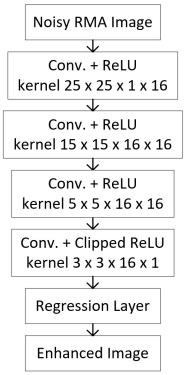


Fig. 5. Architecture of the enhancement FCNN. The selected kernel and layer sizes are capable of adequately learning the non-ideal shape of the distorted RMA image while maintaining high computational efficiency for real-time implementation.

Our enhancement FCNN is trained on both real human hand data and simulated point target data. First, hand data is collected by capturing frames while the user holds their hand at known locations in the ROI. The magnitude of each pixel is taken, $|\hat{p}(y, z)|$, and the real-valued image is the input for training the FCNN allowing the network to fit the non-ideal beam pattern, real multipath and multistatic effects, and actual reflection of a human hand. Next, simulated data is used to supplement the training set. Each simulated sample is generated by simulating a MIMO beat signal with one ideal point target located at a random known location and additive real device noise. The real device noise is captured from the radar with an empty scene. Now, the network learns the device and ambient environment noise as well as the theoretical multistatic effects that remain even after the multistatic-to-monostatic phase correction, thereby improving its generalizability. Our novel training technique results in a robust and generalizable FCNN that improves image signal to noise ratio (SNR) and localization by fitting to the non-ideal imaging constraints.

The output of the FCNN is generated according to the expected images as modeled by

$$\mathcal{I}(y, z) = e^{-(y-y_0)^2/\sigma_y^2 - (z-z_0)^2/\sigma_z^2} \tag{13}$$

where the width of the expected target located at (y_0, z_0) is dictated by σ_y and σ_z in the y and z dimensions, respectively,

yielding resolutions of $1.18\sigma_y$ and $1.18\sigma_z$ according to the 3dB beamwidth [31]. The output data is generated requiring knowledge of the exact location of the human hand or point target of each input data sample. During training, the FCNN learns to match each noisy, distorted real-valued RMA image to the expected image learning the various non-idealities inherent to the Radar Musical Instrument scenario. Once trained, the network denoises the RMA images and provides a narrow peak at a more precise location allowing for improved localization and subsequent tracking. Results are presented in section V.

Additionally, the enhancement RMA image can be multiplied element-wise by the complex-valued RMA to isolate the complex-valued peak from the hand and mitigate clutter and noise. As a result, the Doppler velocity spectrum SNR is improved, as shown later, since clutter and noise, whose phase adversely affects the Doppler estimation, are greatly reduced. In this way, the enhancement FCNN improves both the spatial and temporal feature estimation in real-time. Pairing the enhancement FCNN with the Doppler-corroborated modified particle filter algorithm, the signal processing chain for the enhanced gesture tracking enabled Radar Musical Instrument is shown in Fig. 6.

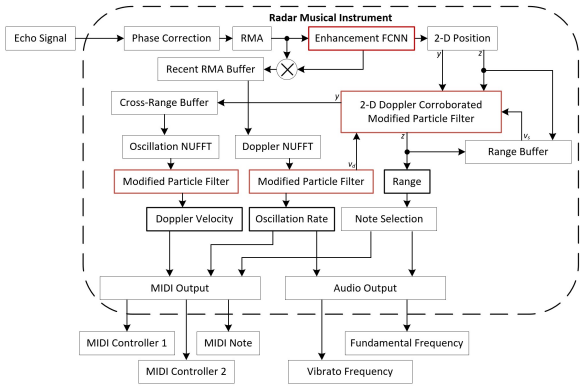


Fig. 6. Radar Musical Instrument signal processing chain from input to musical output now with enhanced gesture tracking by including the Doppler-corroborated modified particle filter algorithm and the enhancement FCNN. Key elements to the enhanced tracking signal processing are highlighted in red.

Utilizing the modified particle filter algorithm and its extension to include Doppler-corroboration improves the robustness of the gesture tracking allowing the Radar Musical Instrument to finely track acute hand gestures even in the presence of noisy user input. Furthermore, the inclusion of the enhancement FCNN enables highly precise position estimation by learning the non-idealities in the device itself and the environment.

V. Results

A. Simple Gesture Tracking Results

As discussed in the previous section, the simple gesture tracking method relies on the traditional techniques presented in section III without any additional tracking algorithm. In real-time, the signal processing chain shown in Fig. 3 is performed, extracting the spatiotemporal features (range, cross-range oscillation, and Doppler velocity) and converting them to musical output via audio or MIDI signals. At each iteration, the states (spatiotemporal features) are estimated directly from the extracted measurements and are therefore prone to erratic behavior.

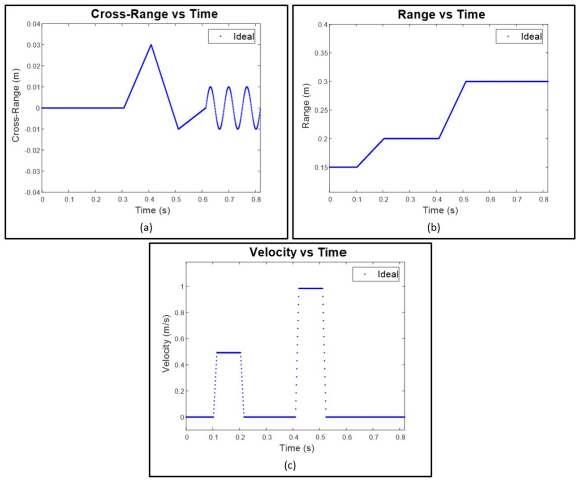


Fig. 7. Ideal motion profile of the target in the (a) cross-range and (b) range directions as well as the (c) range velocity profile against time.

To verify the feature estimation techniques, a virtual prototyping approach is adopted. A point target is simulated in motion with locations and velocity shown in Fig. 7 using (2). Then, real noise collected from the radar with an empty scene is added to each beat signal as

$$\tilde{s}(y_T, y_R, k) = \frac{p}{R_T R_R} e^{jk(R_T+R_R)} + \alpha\tilde{\omega}(y_T, y_R, k), \quad (14)$$

where $\tilde{\omega}$ is a complex-valued randomly selected noise sample corrupting the amplitude and phase of the ideal simulated beat signal and α controls the SNR.

The motion profile shown in Fig. 7 shows the range (z), cross-range (y), and velocity of the target. The motion profile includes independent and joint movement in the range and cross-range domains in addition to sinusoidal cross-range oscillation. For our simulations, 4096 time samples are generated using $p \in [0.5, 1]$ to simulate the variance in the hand's empirical radar cross-section (RCS), as observed from prior hand data, and $\alpha \in [1, 3]$ to vary the SNR from sample to sample. Values for p and α are selected randomly, within the specified intervals, for each time sample and provide a level of stochastic realism to the simulated data.

Fig. 8 shows the features extracted from the simulated noisy beat signals using the simple method. The real radar noise and varying reflectivity result in outliers and errors in the estimated location and velocity of the target. The measurements directly from the RMA image generally follow the motion profile and velocity profile of the target, however suffering from an inherently noisy and inconsistent environment. In the following sections, the performance of the simple gesture tracking approach is quantitatively compared to the enhanced tracking method and design considerations are discussed.[1]

[1]Supplemental material for the reader can be downloaded at http://ieeexplore.org/

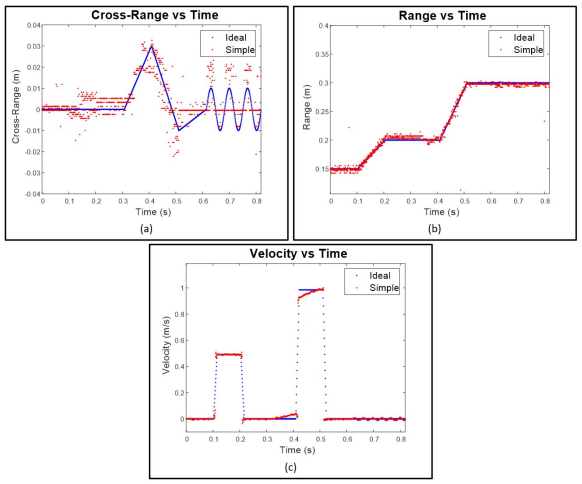


Fig. 8. Motion profile using simple features extraction techniques on each frame for every time step (red) compared with the ideal motion and velocity profiles (blue). The (a) cross-range and (b) range are measured directly from the peak of the RMA image of each frame and the (c) velocity is measured using the Doppler FFT of the raw RMA images using (8) and (9).

B. Enhanced Gesture Tracking Results

Assuming the same motion profile in Fig. 7, the modified particle filter algorithm is employed in an attempt to more robustly track the 2-D position and Doppler velocity of the target across time, improving the musician's control over the Radar Musical Instrument.

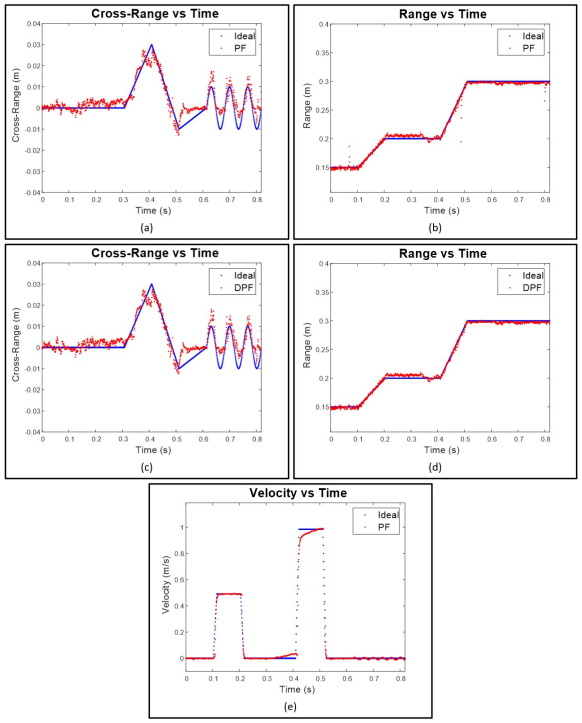


Fig. 9. The modified particle filter (PF) and Doppler-corroborated modified particle filter (DPF) algorithms employed for robust spatiotemporal tracking of the simulated gestures through time: improved tracking of the (a) cross-range and (b) range versus time using the PF, (c) cross-range and (d) range versus time using the DPF with $N_r = 16$, and (e) Doppler velocity versus time using a PF approach.

First, the particle filter algorithm (PF) is implemented using the data in Fig. 8 as elements of the noisy measurement vector $\boldsymbol{z}$. The PF reduces the effect of the noise on the position estimation and improves the spatiotemporal tracking performance as shown in Fig. 9.

Next, the Doppler-corroborated modified particle filter (DPF) is applied to the same set of data additionally improving the state estimation of the range. Notice the outliers in Fig. 9b are mitigated by the DPF in Fig. 9d. This is because the outlying samples result in a sample velocity contradicted by the Doppler velocity and weighted as unimportant. The DPF algorithm improves the playability of the Radar Musical Instrument and the user experience by providing a robust, consistent tracking algorithm to smoothly estimate the 2-D position and spatiotemporal signatures of the musician's gestures. However, key issues discussed previously result in degraded RMA images and subsequent noisy measurements reducing tracking performance.

An image-enhancing fully convolutional neural network is implemented to accommodate non-idealities in the device and environment. The enhancement FCNN is trained using both real data from a human hand and simulated data corrupted by additive real radar noise. The FCNN is trained using 65536 simulated and 23040 real human hand RMA images as the input and output images with $\sigma_y = \sigma_z = 0.001$m resulting in cross-range and range resolutions of 1.18mm. Each simulated sample is generated at a random location in the ROI $z \in [0.1, 0.5], y \in [-0.1, 0.1]$, and the real hand data consists of 512 samples collected at each of the 45 locations throughout the ROI as shown in Fig. 10. The simulated samples cover the entire ROI allowing the network to generalize well to location while learning the non-idealities of the imaging scheme.

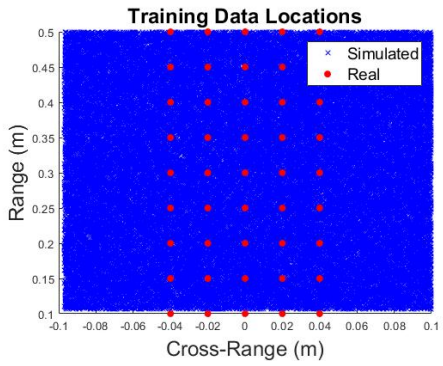


Fig. 10. Locations of the training data used to train the enhancement FCNN. Real data (red) are collected by keeping the hand static at known locations. Simulated data (blue) are generated by choosing locations randomly from the continuous ROI. Simulated data cover nearly the entire ROI allowing the enhancement FCNN to learn locations throughout the ROI.

Training the network for 100 epochs takes 5 hours on a machine with an AMD 3900X processor with 64GB of RAM and a single NVIDIA GTX1080TI graphics card. Other network architectures and training durations are investigated, but this combination yields high performance while offering real-time efficiency.

Now, a dataset similar to the training set is collected and simulated for validation. Fig. 11 shows the images enhanced

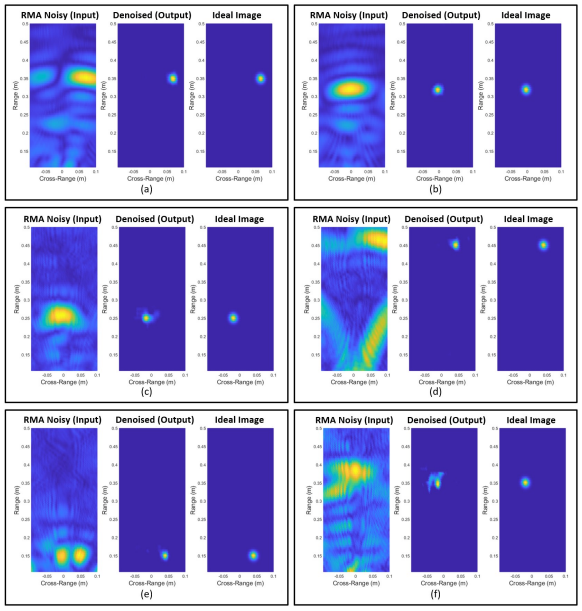


Fig. 11. Enhancement FCNN applied to simulated (a,b) and real hand (c-f) RMA images for image enhancement and improved localization.

by the enhancement FCNN demonstrating the robustness of the network. Figs. 11a and 11b show simulated point targets denoised by the FCNN resulting in improved resolution and more precise localization in the denoised image. Fig. 11c is an RMA image reconstructed from a real hand capture close to the middle of the cross-range domain. The 2-D position of the hand is accurately located compared with the ideal image. Similarly, Figs. 11d-11f demonstrate the network's capability to enhance images degraded by small hand RCS in comparison to noise, the non-ideal beam pattern of each element resulting in ghosting, ambient and device noise, or other non-idealities.

TABLE I
SIMPLE VS ENHANCED LOCALIZATION RMSE

	y (m)	z (m)
Simple	0.0154	0.023
Enhanced	0.0085	0.0083

To quantitatively compare the localization improvement of the enhancement FCNN compared to the simple method, the RMSE in the range and cross-range position is computed on the validation dataset using the two techniques and shown in Table I. The enhancement FCNN not only improves the resolution of the RMA images but results in more accurate and precise localization for both simulated and real data. Applying the enhancement FCNN can further improve the tracking of the spatiotemporal features.

First, the network improves the Doppler velocity spectrum SNR. Following the signal processing chain in Fig. 6, the real-valued enhanced RMA image is multiplied element-wise with the complex-valued raw RMA image to preserve the target velocity phase term while mitigating noise and clutter. As shown in Fig. 12, the Doppler spectrum SNR is improved when the Doppler processing is performed on the enhanced RMA images as compared with the raw RMA images, reducing the likelihood of erroneously estimating the velocity. As a result, the enhancement network improves the reliability of the Doppler velocity estimation aiding spatial tracking. ‘ Additionally, the enhancement FCNN improves the tracking accuracy of the Doppler-modified particle filter tracking by improving localization accuracy and dependability of the Doppler velocity. The FCNN enhanced Doppler-corroborated modified particle filter algorithm is abbreviated by FCNN-DPF. Fig. 13 demonstrates the tracking using the enhancement FCNN paired with the DPF on the same data as the previous tracking examples. Now, the range and cross-range tracking of the target is nearly identical to the ideal motion profile and an improvement in the velocity estimation.

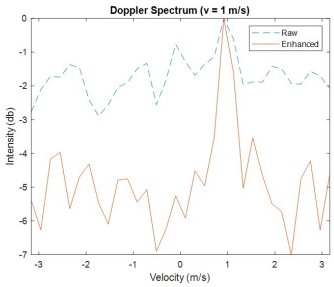


Fig. 12. Comparison of the Doppler velocity spectrum when the Doppler FFT and video pulse integration steps are performed on the raw RMA images compared to the enhanced RMA images. The simulated data contains 128 frames and uses $\alpha = 3$ for every capture to simulate a low SNR scenario.

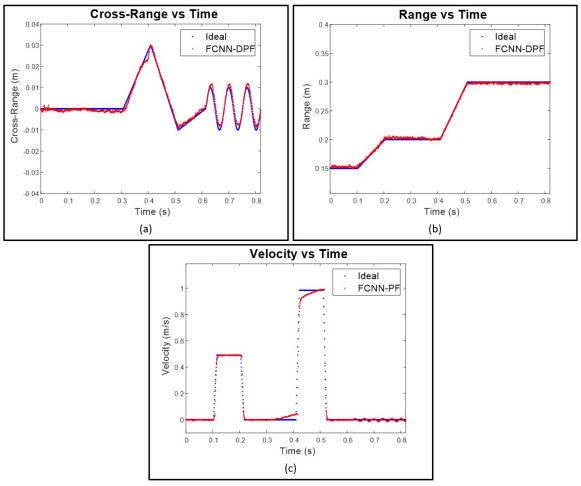


Fig. 13. The FCNN enhanced Doppler-corroborated modified particle filter algorithm.

To quantitatively compare the tracking performance among the various methods described, 4096 unique motion profiles are generated and corresponding tracking RMSE is computed for the cross-range, range, and velocity. Displayed in Table II, the RMSE for the cross-range (y), range (z), and velocity (v) improve with the novel algorithms proposed in this paper.

As expected, the simple method yields the greatest error for all three features. Comparing PF and DPF, the cross-range and velocity RMSE are identical between the two techniques but the range RMSE is improved due to the real-time range importance weighting. The FCNN alone outperforms

the simple method but can be improved by including the PF and DPF after image enhancement. Finally, the FCNN-PF and FCNN-DPF yield identical results for the cross-range and velocity RMSE, as expected, but significant improvement can be noted in the range RMSE. The results in Table II demonstrate the considerably superior tracking performance of the enhancement tracking methods, namely the FCNN-DPF, compared with the simple tracking technique along with their respective latencies in the full instrument implementation.

TABLE II
AVERAGE RMSE FOR TRACKING METHODS

	y-Cross-Range (mm)	z-Range (mm)	v-Velocity (m/s)	Latency (ms)
Simple	7.8627	22.0393	0.0724	17
PF	5.2667	13.6412	0.0529	22
DPF	5.2667	6.8543	0.0529	23
FCNN	7.7365	12.2636	0.0584	25
FCNN-PF	3.6974	7.4431	0.0445	28
FCNN-DPF	3.6972	3.0742	0.0445	29

VI. DISCUSSION AND FUTURE WORK

The enhanced gesture tracking method outperforms the simple method in localization accuracy, Doppler SNR, and tracking accuracy; however, there are some necessary trade-offs for this performance gain. First, the effectiveness of the enhancement FCNN is limited by its training set. Specifically, the network is trained on images from a specific region of interest. Due to the limited aperture size, the farther the target is from the radar, the wider the target appears to be. Since the enhancement FCNN is only trained on images within the expected region, extending the ROI outside of the trained region results in performance degradation. In contrast, the simple methods are highly flexible but cannot compete with the performance of the enhancement techniques.

Even though the Doppler-corroborated particle filter improves the tracking robustness, in some cases it can degrade performance. Since the DPF relies on accurate and current Doppler velocity estimation, the system must operate at a fast enough rate that common velocities are within the resolvable range. The UDP interface operates on the order of milliseconds, but the limitation comes in the time between calls to the MATLAB data retrieval function (MEX function). The rate at which the system is acquires the most recent frame is dependent on the speed of the signal processing chain between MEX function calls. As the signal processing time increases, the maximum resolvable velocity can become too small. As a result, real-time tracking performance is degraded since aliasing will occur in the Doppler spectrum domain if the musician's hand is moved too fast. In our work, we found the simple and enhanced methods required similar computation times limiting the frequency of the data retrieval function call to a maximum of around 250Hz. At a starting frequency of 77GHz, this yields a maximum resolvable velocity of 0.24m/s limiting the effectiveness of the DPF when the magnitude of the hand velocity is above this maximum. However, the software package presented in this article is meant to serve as a platform for further development of real-time HCI algorithms on mmWave devices and the latency performance will increase substantially when the algorithms are implemented on an embedded device.

In terms of computational efficiency, the addition of the FCNN and particle filter algorithm increases the computational cost slightly compared to the simple measurement method. However, with GPU acceleration more widely available on most PCs and many MCUs, our comparison shows the latency impact to be negligible among the proposed algorithms. The latencies of the methods introduced in this article show a low quantitative variance. In a small single-blind user study, participants consistently noticed a performance increase as more sophisticated algorithms were introduced, matching the trend in Table II, but were unable to perceive a difference in latency among the algorithms.

The system can be characterized by several key performance metrics. In latency tests, we were able to achieve response times averaging 29ms from gesture movement to MIDI output for the radar musical instrument, comparable to many MIDI interfaces. The latency can be further reduced significantly by implementing the algorithms on an embedded device, but we are pleased with the computational performance on a prototyping platform such as MATLAB. Similarly, the sampling rate of the human hand at around 250Hz can be improved by the same means. Given the scenario and radar under test, the expected range resolution and cross-range resolutions are 3.75cm and 15cm, respectively. However, the proposed methods yield a much improved spatial resolution of 1.96mm and 2.3mm for range and cross-range, respectively. The sensing area of the radar under test is only 0.5m along the range dimension and 0.15m along the cross-range dimension, given the weak reflection from the human hand and small antenna beamwidth. Using a device with more elements or elements with a larger beamwidth would increase the cross-range resolution and sensing area. The software package presented in this article is designed for flexible development and prototyping of algorithms and devices, enabling streamlined hardware changes and upgrades. Lastly, the 2-D position tracking techniques and provided software can easily be extended to 3-D if a 2-D array is employed allowing an added degree of control to the musical instrument at the cost of a larger array topology.

The choice of a mmWave sensor for this application is an effort to prove the concept of acute hand position tracking using radar technology. Compared to optical and RGB+D solutions, mmWave radar is more versatile and reliable operating well under occlusion, in any temperature or lighting environment, and offering precise depth information of the entire scene. For a musical interface, these advantages may not be often fully realized; however, the novel tracking methods proposed in this article are applicable for many tracking applications and serve as a demonstration of their efficacy for HCI and beyond. On the other hand, mmWave sensors cannot meet the performance of optical solutions when it comes to cross-range resolution due to the limited number of radar antennas, making multi-object and finger tracking much more challenging. As such, many applications in HCI, computer vision, automated driving, etc. employ radar (and lidar) and optical imaging devices

with sensor fusion algorithms to achieve further improved performance at an increased cost.

Several alternatives exist to mmWave radar sensing, namely hand-held and optical devices. Hand-held sensing solutions offer highly precise spatiotemporal features, but are not preferable compared to contactless sensors [32], [33]. On the optical front, much effort towards accurate hand position and pose tracking has been made. On the order of $\$100 - \$200$, the Kinect and Leap Motion are in the same price bracket as most mmWave radar devices, which are becoming increasingly inexpensive. Attempts using multiple RGB cameras [34], [35] show promising results; however, a single device is much preferred as multiple cameras setups are cumbersome. Single RGB+D solutions have been proposed using generative pose tracking [36], [37] and learning-based generative pose tracking [38], [39]. However, all of these methods suffer tremendously under occlusion from objects or scene clutter, one of the strengths of mmWave radar. Some deep learning oriented solutions have shown quite promising results [40], [41], but constructing a sufficient dataset for meaningful supervised training remains a challenge.

The Radar Musical Instrument tracks the 2-D position and velocity of the musician's hand to control note selection and two user-selected parameters, a marked improvement over the prior work on mmWave radar tracking only 1-D range for parameter control [16]. However, optical solutions enable tracking of both hands [3], [7], [9], [36]–[41] or hand and finger position [8], [12] for even finer musical control, with some scenario-specific drawbacks. As radar sensor technology improves, tracking the individual fingers on the hand will become increasingly plausible and we expect it to yield comparable or superior results to optical solutions due to higher depth resolution. Compared to prior on gesture tracking with mmWave devices, our proposed methods yield impressive results. Past work using radar devices achieves at best average range tracking error of 2cm [17]. Again using the 4096 simulated motion profiles with added noise, our enhanced gesture tracking technique yields a mean range tracking error of 1.89mm, improving by more than a factor of ten. In [19], a 4GHz bandwidth mmWave sensor is used in conjunction with two optical cameras to achieve a 2-D position RMSE of 1.16mm, at distances closer than 10cm. Comparatively, our enhanced gesture tracking algorithm uses a singular radar device and no optical cameras and still yields a competitive 2-D position RMSE of 3.4mm at a lower cost and computational load. Moreover, our method offers a mean range resolution of 1.96mm, a significant improvement over the 4GHz bandwidth range resolution of 3.75cm. At the time of this paper, we are not aware of any other prior work on hand tracking using mmWave devices. To our knowledge, the system proposed in this paper offers unprecedented hand gesture tracking performance using a single mmWave sensor.

The most direct comparison to the Radar Musical Instrument, however, is the Theremin, both being controlled by the hand's proximity to the sensor. The pitch of the Theremin is controlled continuously by the hand's vertical location, whereas the Radar Musical Instrument tracks the range of the hand digitally and selects a note from the user-defined scale. While the Theremin uses two antennas, one for volume control and the other for pitch control, a total of two degrees of freedom, the Radar Musical Instrument tracks range, cross-range, and velocity, providing three controllable parameters. The Radar Musical Instrument is an evolved Theremin, utilizing a modern mmWave sensor for precise tracking and control of the hand in 2-D space (expansion to 3-D can be easily implemented with the proper hardware), while the Theremin offers a different mode of input, being continuous note selection and using the other hand for volume control at another sensor. One of the authors is a skilled guitar and violin instrumentalist with a background in electronic music production. From the perspective of an experienced musician, the proposed methods offer an elegant new musical interface capable of generating unique phrases previously only available in offline manual transcription and provides the musician a sufficient and consistent level of control and latency, comparable to modern MIDI interfaces. In contrast to a Theremin, the Radar Musical Instrument is significantly less effortful in note selection, allowing simple and intuitive inclusion of the additional parameter controls and increasing accessibility to the expected user-base.

For future work, several promising routes are left to be explored. First, further development of high-throughput radar devices will increase the speed of the real-time data capture process, allowing for increased reliance on the Doppler velocity estimations. Using multiple MIMO radars or a larger MIMO array would enable a multiple-hand and individual finger tracking interface, thus further extending the application space of our robust tracking methods. Additionally, the MATLAB-based system implementation in this paper can be implemented on high-speed DSP controllers to create a portable system. Finally, the novel precise gesture tracking algorithms of Radar Musical Instrument can easily be extended to offer an elegant, efficient solution to a host of acute gesture tracking problems.

VII. Conclusion

The Radar Musical Instrument successfully demonstrates the viability of acute human hand gesture tracking for human-computer interaction using mmWave sensors. We validated and implemented our real-time spatiotemporal signal processing algorithms and robust tracking algorithms in the form of a musical interface; however, the Radar Musical Instrument demonstrates the broad effectiveness of mmWave technology for a multitude of near-field acute gesture tracking applications. First, simple feature extraction and tracking methods were introduced, followed by an enhanced approach leveraging the Doppler-corroborated modified particle filter algorithm and enhancement FCNN to achieve highly robust and accurate tracking in the presence of ambient noise and offering compensation for device non-idealities and multipath effects. The methods are compared demonstrating noticeable improvement using the FCNN-DPF. Additionally, our work offers superior tracking estimation and localization accuracy compared to prior methods in the literature for both mmWave and optical implementations. The novel Radar Musical Instrument presented in this paper offers an elegant solution to a myriad of

contactless human-computer interaction problems far beyond the scope of musical interfaces.

ACKNOWLEDGMENT

This work was supported by the imec USA summer internship program. We would like to extend thanks to Dr. Gonzalo Vaca Castano for his insights in developing the particle filter algorithm and computer vision approach.

REFERENCES

[1] T. Winkler, "Making motion musical: Gesture mapping strategies for interactive computer music," in *ICMC*, 1995, p. 26.

[2] K. D. Skeldon, L. M. Reid, V. McInally, B. Dougan, and C. Fulton, "Physics of the theremin," *American Journal of Physics*, vol. 66, no. 11, pp. 945–955, 1998.

[3] R. Polfreman, "Multi-modal instrument: towards a platform for comparative controller evaluation." in *ICMC*, 2011.

[4] S. Trail, M. Dean, G. Odowichuk, T. F. Tavares, P. F. Driessen, W. A. Schloss, and G. Tzanetakis, "Non-invasive sensing and gesture control for pitched percussion hyper-instruments using the kinect." in *NIME*, 2012.

[5] S. Sentürk, S. W. Lee, A. Sastry, A. Daruwalla, and G. Weinberg, "Crossole: A gestural interface for composition, improvisation and performance using kinect." in *NIME*, 2012.

[6] R. Schramm, C. R. Jung, and E. R. Miranda, "Dynamic time warping for music conducting gestures evaluation," *IEEE Transactions on Multimedia*, vol. 17, no. 2, pp. 243–255, 2015.

[7] A. R. Jensenius, "Kinectofon: Performing with shapes in planes," 2013.

[8] J. Han and N. Gold, "Lessons learned in exploring the leap motion™ sensor for gesture-based instrument design." Goldsmiths University of London, 2014.

[9] L. Hantrakul and K. Kaczmarek, "Implementations of the leap motion in sound synthesis, effects modulation and assistive performance tools." in *ICMC*, 2014.

[10] D. Brown, N. Renney, A. Stark, C. Nash, and T. Mitchell, "Leimu: Gloveless music interaction using a wrist mounted leap motion," 2016.

[11] A. Tindale, A. Kapur, and G. Tzanetakis, "Training surrogate sensors in musical gesture acquisition systems," *IEEE Transactions on Multimedia*, vol. 13, no. 1, pp. 50–59, 2011.

[12] O. Nieto and D. Shasha, "Hand gesture recognition in mobile devices: Enhancing the musical experience," *Proc. of CMMR*, vol. 13, 2013.

[13] M. Akbari and H. Cheng, "Real-time piano music transcription based on computer vision," *IEEE Transactions on Multimedia*, vol. 17, no. 12, pp. 2113–2121, 2015.

[14] J. W. Smith, S. Thiagarajan, R. Willis, Y. Makris., and M. Torlak, "Improved static hand gesture classification on deep convolutional neural networks using novel sterile training technique," *IEEE Access*, pp. 1–1, 2021.

[15] S. Skaria, A. Al-Hourani, M. Lech, and R. J. Evans, "Hand-gesture recognition using two-antenna doppler radar with deep convolutional neural networks," *IEEE Sensors Journal*, vol. 19, no. 8, pp. 3041–3048, 2019.

[16] F. Bernardo, N. Arner, and P. Batchelor, "O soli mio: exploring millimeter wave radar for musical interaction." in *NIME*, vol. 17, 2017, pp. 283–286.

[17] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, "Wideo: Fine-grained device-free motion tracing using rf backscatter," in *12th USENIX Symposium on Networked Systems Design and Implementation NSDI 15)*, 2015, pp. 189–204.

[18] Y. Sun, X. Liang, H. Fan, M. Imran, and H. Heidari, "Visual hand tracking on depth image using 2-d matched filter," in *2019 UK/ China Emerging Technologies (UCET)*, August 21-22, 2019, Glasgow, United Kingdom, pp. 1–4.

[19] Z. Li, Z. Lei, A. Yan, E. Solovey, and K. Pahlavan, "Thumouse: A micro-gesture cursor input through mmwave radar-based interaction," in *2020 IEEE International Conference on Consumer Electronics (ICCE)*, January 4-6, 2020, Las Vegas, NV, USA,, pp. 1–9.

[20] H. Helmholtz, *On the sensations of tone*. Courier Corporation, 2013.

[21] C. Schmidt-Jones, "Understanding basic music theory," 2013.

[22] J. W. Smith, M. E. Yanik, and M. Torlak, "Near-field mimo-isar millimeter-wave imaging," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

[23] M. E. Yanik and M. Torlak, "Near-field mimo-sar millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31 801–31 819, 2019.

[24] M. E. Yanik, D. Wang, and M. Torlak, "Development and demonstration of mimo-sar mmwave imaging testbeds," *IEEE Access*, vol. 8, pp. 126 019–126 038, 2020.

[25] V. Winkler, "Range doppler detection for automotive fmcw radars," in *Proc. European Radar Conf.*, October 10-12, 2007, Munich, Germany, pp. 166–169.

[26] S. Rao, "Introduction to mmwave sensing: Fmcw radars."

[27] J. Kim, J. Chun, and S. Song, "Joint range and angle estimation for fmcw mimo radar and its application," *arXiv preprint arXiv:1811.06715*, 2018.

[28] "Texas instruments mmwave studio." [Online]. Available: https://www.ti.com/tool/MMWAVE-STUDIO

[29] J. García, A. Gardel, I. Bravo, J. L. Lázaro, and M. Martínez, "Tracking people motion based on extended condensation algorithm," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 43, no. 3, pp. 606–618, 2013.

[30] Y. Dai, T. Jin, Y. Song, H. Du, and D. Zhao, "Cnn-based multiple-input multiple-output radar image enhancement method," *The Journal of Engineering*, vol. 2019, no. 20, pp. 6840–6844, 2019.

[31] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geoscience and Remote Sensing Letters*, vol. 16, no. 1, pp. 35–39, 2019.

[32] L. Pardue and W. Sebastian, "Hand-controller for combined tactile control and motion tracking." in *NIME*, 2013, pp. 90–93.

[33] P. Neto, J. N. Pires, and A. P. Moreira, "High-level programming and control for industrial robotics: using a hand-held accelerometer-based input device for gesture and posture recognition," *Industrial Robot: An International Journal*, 2010.

[34] L. Ballan, A. Taneja, J. Gall, L. Van Gool, and M. Pollefeys, "Motion capture of hands in action using discriminative salient points," in *European Conference on Computer Vision*. Springer, 2012, pp. 640–653.

[35] S. Sridhar, A. Oulasvirta, and C. Theobalt, "Interactive markerless articulated hand motion tracking using rgb and depth data," in *Proceedings of the IEEE international conference on computer vision*, 2013, pp. 2456–2463.

[36] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Efficient model-based 3d tracking of hand articulations using kinect." in *BmVC*, vol. 1, no. 2, 2011, p. 3.

[37] D. Tang, J. Taylor, P. Kohli, C. Keskin, T.-K. Kim, and J. Shotton, "Opening the black box: Hierarchical sampling optimization for estimating human hand pose," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 3325–3333.

[38] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt, "Fast and robust hand tracking using detection-guided optimization," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3213–3221.

[39] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff *et al.*, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–12, 2016.

[40] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Transactions on Graphics (ToG)*, vol. 33, no. 5, pp. 1–10, 2014.

[41] Q. Ye, S. Yuan, and T.-K. Kim, "Spatial attention deep net with partial pso for hierarchical hybrid hand pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 346–361.

Near-Field MIMO-ISAR Millimeter-Wave Imaging

Josiah Wayland Smith[1], Muhammet Emin Yanik[2], and Murat Torlak[1]

[1]Dept. of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX, United States
[2]Radar and Analytics, Texas Instruments, Dallas, TX, United States

***Abstract*—Multiple-input-multiple-output (MIMO) millimeter-wave (mmWave) sensors for synthetic aperture radar (SAR) and inverse SAR (ISAR) address the fundamental challenges of cost-effectiveness and scalability inherent to near-field imaging. In this paper, near-field MIMO-ISAR mmWave imaging systems are discussed and developed. The rotational ISAR (R-ISAR) regime investigated in this paper requires rotating the target at a constant radial distance from the transceiver and scanning the transceiver along a vertical track. Using a 77GHz mmWave radar, a high resolution three-dimensional (3-D) image can be reconstructed from this two-dimensional scanning taking into account the spherical near-field wavefront. While prior work in literature consists of single-input-single-output circular synthetic aperture radar (SISO-CSAR) algorithms or computationally sluggish MIMO-CSAR image reconstruction algorithms, this paper proposes a novel algorithm for efficient MIMO 3-D holographic imaging and details the design of a MIMO R-ISAR imaging system. The proposed algorithm applies a multistatic-to-monostatic phase compensation to the R-ISAR regime allowing for use of highly efficient monostatic algorithms. We demonstrate the algorithm's performance in real-world imaging scenarios on a prototyped MIMO R-ISAR platform. Our fully integrated system, consisting of an mechanical scanner and efficient imaging algorithm, is capable of pairing the scanning efficiency of the MIMO regime with the computational efficiency of single pixel image reconstruction algorithms.**

***Index Terms*—millimeter-wave (mmWave), multiple-input multiple-output (MIMO), inverse synthetic aperture radar (ISAR), three-dimensional (3-D) imaging.**

I. Introduction

Over the past several decades, developments in system-on-chip complementary metal oxide semiconductor (CMOS) radio frequency integrated circuits (RFIC) have resulted in the emergence of frequency modulated continuous wave (FMCW) millimeter wave (mmWave) radars as a cost-effective solution for imaging applications. The 3-D holographic imaging regime has been investigated in the rectilinear (planar) mode [1], [2] and in cylindrical mode [3]. Additionally, progress has been made towards efficient algorithms for single-input-single-output (SISO) monostatic array synthetic aperture radar (SAR) [4] and multi-input-multi-output (MIMO) multistatic array SAR [5]. Specifically, Gao's work at China's National University of Defense and Technology (NUDT) has demonstrated algorithms for 2-D circular SAR (CSAR) imaging [6] and 3-D MIMO-CSAR imaging [7]. While SISO-CSAR algorithms proposed by Sheen [8], Laviada [9], Gao, and others are efficient in generating high resolution 3-D holographic images, they ignore the multistatic effects from a MIMO array, resulting in aliasing and phase mismatch from the ideal SISO case. While MIMO-CSAR algorithms have been developed in attempt to solve such issues, these algorithms are computationally expensive and inefficient in comparison to their SISO counterparts. In this paper, we propose a resolution to this dilemma by leveraging the benefits of MIMO-CSAR, fewer antenna elements and cost efficiency, with the streamlined computational efficiency of the SISO-CSAR algorithms to produce a highly efficient high-resolution 3-D imaging algorithm. Under this MIMO rotation ISAR (R-ISAR) regime, a robust imaging system is prototyped to verify the proposed algorithm and demonstrate its performance.

The rest of this paper is formatted as follows. Section II discusses the return signal from the proposed MIMO R-ISAR scenario and the multistatic-to-monostatic conversion. Section III contains the derivation for the 3-D image reconstruction algorithm in the SISO R-ISAR regime and crucial multistatic-to-monostatic phase correction. Section IV overviews issues including sampling criteria and spatial resolution. Section V verifies the proposed algorithm in simulation. The imaging prototype is described in Section VI. Real 3-D imaging results are reported in Section VII, including a comparison of R-ISAR to SAR, followed finally by conclusions.

II. MIMO R-ISAR Signal Model

A. MIMO Rotational ISAR (R-ISAR) Echo Signal

The MIMO R-ISAR scenario, as shown in Fig. 1, consists of a rotational scanner whose center is the origin and a MIMO array scanned along the y-axis (vertically), located at a constant distance of R_0 from the center of the rotator. The distance of the transmitter, located at the vertical position y'_T, and receiver, located at the vertical position y'_R, from each point in the target domain (x, y, z) depends on the rotation angle θ and the distance R_0.

$$R_T = \sqrt{(x - R_0\cos\theta)^2 + (z - R_0\sin\theta)^2 + (y - y'_T)^2},$$
$$R_R = \sqrt{(x - R_0\cos\theta)^2 + (z - R_0\sin\theta)^2 + (y - y'_R)^2}. \quad (1)$$

978-1-7281-8942-0/20/$31.00 ©2020 IEEE

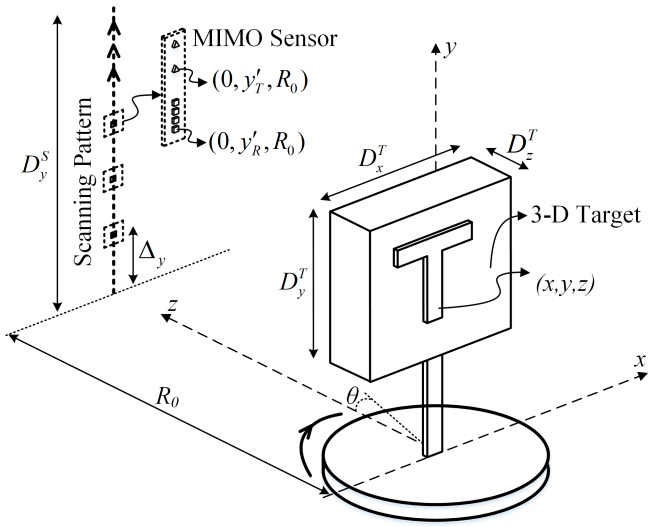


Fig. 1: The geometry of the MIMO R-ISAR imaging configuration, where a cylindrical aperture is synthesized by mechanically moving a linear MIMO array vertically and rotating the target.

The MIMO echo signal from the R-ISAR scenario can be modeled as:

$$\mathrm{s}(\theta, k, y'_T, y'_R) = \iiint \frac{\mathrm{p}(x,y,z)}{R_T R_R} e^{jk(R_T+R_R)} dxdydz, \quad (2)$$

where θ is the angle of rotation, and k is the wavenumber. For our purposes θ is allowed to be wide-angle up to $360°$.

B. Multistatic-to-Monostatic Conversion

The received echo data from the multistatic MIMO array undergo a simple transformation to approximate its counterpart echo signal from the virtual-SISO array elements. To convert this four-dimensional (4-D) MIMO signal to a 3-D virtual-SISO signal, the following phase compensation is performed [10], [11],

$$\hat{\mathrm{s}}(\theta, k, y') = \mathrm{s}(\theta, k, y'_T, y'_R) e^{-jk\frac{Sy^2}{4R_0}}, \quad (3)$$

where y' is the vertical scanning dimension containing all y'_T and y'_R positions, and d_y is the distance between pairs of transmitting and receiving elements. This multistatic-to-monostatic conversion only holds for small values of d_y

This compensation is a crucial step in the algorithm. By compensating the phase of the multistatic MIMO signal to obtain an approximate of the echo signal from virtual elements located at the midpoint of each MIMO transmitter/receiver pair, this virtual-SISO data can be fed into the efficient 3-D imaging algorithm derived in section III.

III. Derivation of 3-D Image SISO Reconstruction Algorithm

The derivations given in this section are similar to the CSAR algorithm derived by Sheen in [8], but contain several key differences vital to performing successful 3-D holographic image reconstruction for a circular scanning scenario.

Using the R-ISAR scenario shown in Fig. 1, the return signal from a monostatic SISO transceiver, neglecting amplitude terms, can be modeled as

$$\hat{\mathrm{s}}(\theta, k, y') = \iiint \mathrm{p}(x,y,z) e^{j2kR} dxdydz, \quad (4)$$

where

$$R = \sqrt{(x - R_0\cos\theta)^2 + (z - R_0\sin\theta)^2 + (y - y')^2}, \quad (5)$$

and $\mathrm{p}(x,y,z)$ is the complex reflectivity function of the target scene. Using the method of stationary phase (MSP), the exponential term in (4) can be decomposed by

$$e^{j2k\sqrt{(R_0\cos\theta - x)^2 + (R_0\sin\theta - z)^2 + (y-y')^2}} = \iint e^{jk_r\cos\phi(R_0\cos\theta - x) + jk_r\sin\phi(R_0\sin\theta - z) + jk_{y'}(y-y'))} d\phi dk_{y'}. \quad (6)$$

Note that an identical result can be found by decomposing the free-space Green's function of a point source in the spatial spectral domain [12]. The angle of each plane wave component in the x-z plane is ϕ, and $k_{y'}$ is the y-component of the wavenumber. Using the dispersion relation

$$4k^2 = k_x^2 + k_y^2 + k_z^2, \quad (7)$$

we define k_r as the wavenumber component in the x-z plane as

$$k_r = \sqrt{k_x^2 + k_z^2} = \sqrt{4k^2 - k_y^2}. \quad (8)$$

Combining the above relations yields

$$\hat{\mathrm{s}}(\theta, k, y') = \iint \left[\iiint \mathrm{p}(x,y,z) e^{-j(k_r\cos\phi)x - j(k_r\sin\phi)z - jk_{y'}y} dxdydz \right] \times e^{jk_r R\cos(\theta-\phi) + jk_{y'}y'} d\phi dk_{y'}, \quad (9)$$

The term inside the $[\bullet]$ brackets is the 3-D Fourier transform of the reflectivity function. Using the following Fourier transform pair in polar coordinates:

$$\mathrm{p}(x,y,z) \iff \mathrm{P}(k_r\cos\phi, k_y, k_r\sin\phi), \quad (10)$$

(9) yields

$$\hat{\mathrm{s}}(\theta, k, y') = \iint e^{jk_r R\cos(\theta-\phi)} \times \mathrm{P}(k_r\cos\phi, k_y, k_r\sin\phi) e^{jk_{y'}y'} d\phi dk_{y'}. \quad (11)$$

Taking the Fourier transform with respect to y' on both sides and dropping the distinction between y' and y due to coincidence of the domains:

$$\hat{\mathrm{S}}(\theta,k,k_y)=\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} e^{jk_rR\cos(\theta-\phi)}\mathrm{P}(k_r\cos\phi,k_y,k_r\sin\phi)d\phi. \tag{12}$$

Defining:

$$\hat{\mathrm{P}}(\phi,k_r,k_y)\triangleq \mathrm{P}(k_r\cos\phi,k_y,k_r\sin\phi) \tag{13}$$

$$\mathrm{g}(\theta,k_r)\triangleq e^{jk_rR_0\cos\theta}. \tag{14}$$

Now:

$$\hat{\mathrm{S}}(\theta,k,k_y)=\int_{-\frac{\pi}{2}}^{\frac{\pi}{2}} \mathrm{g}(\theta-\phi,k_r)\hat{\mathrm{P}}(\phi,k_r,k_y)d\phi, \tag{15}$$

which represents a convolution in the θ domain:

$$\hat{\mathrm{S}}(\theta,k,k_y)=\mathrm{g}(\theta,k_r)\circledast_\theta \hat{\mathrm{P}}(\theta,k_r,k_y), \tag{16}$$

where $\circledast_\theta$ is the convolution operator along the θ domain.

Taking the Fourier transform across the θ domain on both sides yields:

$$\hat{\mathrm{S}}(\Theta,k,k_y)=\mathrm{G}(\Theta,k_r)\tilde{\mathrm{P}}(\Theta,k_r,k_y), \tag{17}$$

where

$$\mathrm{G}(\Theta,k_r)=\mathrm{FT}_{1D}^{(\theta)}[\mathrm{g}(\theta,k_r)], \tag{18}$$

$$\tilde{\mathrm{P}}(\Theta,k_r,k_y)=\mathrm{FT}_{1D}^{(\theta)}[\hat{\mathrm{P}}(\theta,k_r,k_y)] \tag{19}$$

Solving for $\tilde{P}$ by taking the inverse filter $G^*(\Theta,k_r)$ and then taking an inverse Fourier transform across the Θ domain for both sides to obtain $\hat{P}(\theta,k_r,k_y)$:

$$\tilde{\mathrm{P}}(\Theta,k_r,k_y)=\hat{\mathrm{S}}(\Theta,k,k_y)\mathrm{G}^*(\Theta,k_r), \tag{20}$$

$$\hat{\mathrm{P}}(\theta,k_r,k_y)=\mathrm{IFT}_{1D}^{(\Theta)}\left[\hat{\mathrm{S}}(\Theta,k,k_y)\mathrm{G}^*(\Theta,k_r)\right] \tag{21}$$

By (13):

$$\mathrm{P}(k_r\cos\theta,k_r\sin\theta,k_y)=\mathrm{IFT}_{1D}^{(\Theta)}\left[\hat{\mathrm{S}}(\Theta,k,k_y)\mathrm{G}^*(\Theta,k_r)\right], \tag{22}$$

where $k_x=k_r\cos\theta$ and $k_z=k_r\sin\theta$. The definition of these horizontal wavenumber components is crucial in the derivation of the reconstruction algorithm and have been improperly defined in past MSP-based algorithms [8]. $\hat{P}(\theta,k_r,k_y)$ will be a uniformly sampled function of θ and k_r and will need to be interpolated onto a uniform (k_x,k_z,k_y) grid via Stolt interpolation using the equation (8) and the following relations:

$$\theta=\tan^{-1}\left(\frac{k_z}{k_x}\right), \tag{23}$$

$$k=\frac{1}{2}\sqrt{k_x^2+k_y^2+k_z^2}. \tag{24}$$

The Stolt interpolation process will be denoted by the $\mathcal{S}[\bullet]$ operator, such that:

$$\mathrm{P}(k_x,k_y,k_z)=\mathcal{S}[\mathrm{P}(k_r\cos\theta,k_r\sin\theta,k_y)]. \tag{25}$$

Finally, the algorithm can be summarized by (26) and (27).

$$\mathrm{p}(x,y,z)=\mathrm{IFT}_{3D}^{(k_x,k_y,k_z)}\left[\mathrm{P}(k_x,k_y,k_z)\right], \tag{26}$$

$$\mathrm{P}(k_x,k_y,k_z)=\mathcal{S}\left[\mathrm{IFT}_{1D}^{(\Theta)}\left[\mathrm{FT}_{2D}^{(\theta,y)}\left[\hat{\mathrm{s}}(\theta,k,y)\right]\mathrm{G}^*(\Theta,k_r)\right]\right]. \tag{27}$$

From the above results, the complete 3-D image reconstruction algorithm is summarized below. To our knowledge, the key pairing of multistatic-to-monostatic conversion with a single pixel reconstruction algorithm has not been shown in prior literature. This novelty allows for an efficient imaging system by leveraging a MIMO array topology without increasing the computational complexity. In that way, the benefits of the MIMO virtual array can be achieved without the need for inefficient MIMO reconstruction algorithms. Rather, with the proposed algorithm, an efficient system can be easily implemented to quickly scan a target and immediately reproduce a high-resolution 3-D holographic image, as discussed in Section VII.

Efficient MIMO R-ISAR 3-D Holographic Imaging Algorithm

1) Gather the raw 4-D MIMO echo data as $\mathrm{s}(\theta,k,y_T,y_R)$.
2) Perform the phase compensation described in (3) to acquire $\hat{\mathrm{s}}(\theta,k,y)$, the 3-D monostatic equivalent.
3) Perform a 2-D FFT across the θ and y dimensions of the phase corrected data to obtain $\hat{S}(\Theta,k,k_y)$.
4) Generate the azimuth filter $\mathrm{g}(\theta,k_r)\triangleq e^{jk_rR_0\cos\theta}$ and implement an FFT across the θ dimension to compute the spectral azimuth filter $\mathrm{G}(\Theta,k_r)$
5) Multiply $\hat{\mathrm{S}}(\Theta,k,k_y)$ by the inverse filter $\mathrm{G}^*(\Theta,k,k_y)$ and perform an IFFT across the Θ domain to obtain $\mathrm{P}(k_r\cos\theta,k_r\sin\theta,k_y)$.
6) Apply Stolt interpolation using the relations in (8), (23), and (24) to transform the polar spatial spectral $\mathrm{P}(k_r\cos\theta,k_r\sin\theta,k_y)$ to the Cartesian $\mathrm{P}(k_x,k_y,k_z)$.
7) Finally, compute a 3-D IFFT across k_x, k_y, and k_z to recover the complex reflectivity function $\mathrm{p}(x,y,z)$.

IV. Discussion of Key Imaging Issues

A. Sampling Criteria

Akin to all sampling applications, spatial sampling in the R-ISAR regime must satisfy the spatial Nyquist theorem. Accordingly, the following sampling criteria must be satisfied for alias-free 3-D holographic image reconstruction as discussed in [7], [8], [13].

$$\Delta_k < \frac{\pi}{2R_T}, \tag{28}$$

$$\Delta_y < \frac{\lambda\sqrt{(D_y^S + D_y^T)^2/4 + R_0^2}}{2(D_y^S + D_y^T)}, \tag{29}$$

$$\Delta_\theta < \frac{\pi\sqrt{R_0^2 + R_T^2}}{2k_{max}R_0R_T}. \tag{30}$$

R_T is the maximum radius of the target scene, k_{max} is the maximum wavenumber, and D_y^T and D_y^S are the target and scan height, respectively.

B. Spatial Resolution

Another significant point of discussion for SAR imaging systems is spatial resolution. Vertical resolution is independent of the horizontal rotation and can be calculated using the effective aperture approach as shown in [13].

$$\delta_y \approx \frac{\lambda_c R_0}{2D_y^S} \tag{31}$$

λ_c is the wavelength of the center frequency. To derive the radial resolution, the problem is restricted to a 2-D horizontal plane, thereby removing the vertical element of the scan. Along this horizontal plane, the point spread function (PSF) can be computed analytically as [14]:

$$\text{PSF}(r,\theta) = k_{max}\frac{\text{J}_1(2k_{max}r)}{\pi r} - k_{min}\frac{\text{J}_1(2k_{min}r)}{\pi r} \tag{32}$$

From (32), the horizontal resolution can be deduced, where $\text{J}_1(\bullet)$ represents the first-order Bessel function and k_{min} is the minimum wavenumber.

$$\delta_R = \frac{2.4}{k_{max} + k_{min}} \tag{33}$$

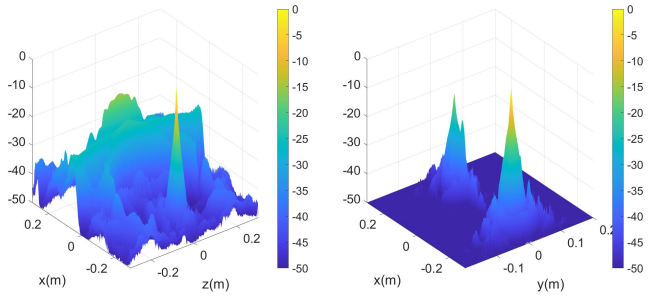
For both the vertical and horizontal resolutions, an ideal point reflector is employed for simplicity sake. However, for real-world applications involving real target scenes, this type of ideal spatial resolvability is rarely achieved [7]. Accordingly, these expressions serve as a lower limit on the empirical spatial resolution.

V. R-ISAR Simulations

To simulate the echo signal, targets are modeled as point reflectors using (2). All the simulations are done using the MIMO parameters shown in Table I where θ_{max} is the maximum rotation angle, N_y is the number of vertical captures, and Δ_y is the vertical spacing between MIMO captures.

A. Point Spread Function (PSF)

Using (2), the echo signal is simulated in MATLAB with a single point reflector located at $(-0.15, 0)$ in the x-z plane and with the parameters in Table I. Then, the image is reconstructed using the proposed algorithm described in Section III. 2-D slices of the 3-D PSF are examined in Fig. 2.

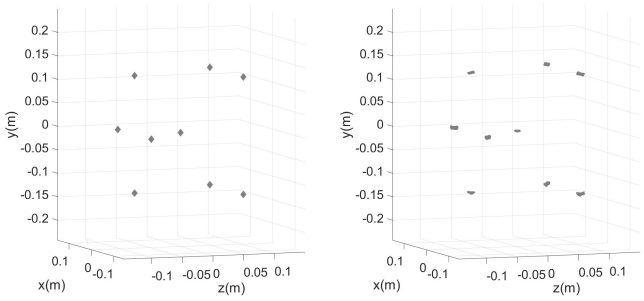


(a) (b)

Fig. 2: Point Spread Function (dB): R_0 = 0.25 m and D_y^S = 484.8 mm (a) 2-D x-z PSF, (b) 2-D x-y PSF

B. 3-D Points

Additionally, to verify the algorithm in simulation, a set of points in 3-D are generated and their echo signal is simulated. The algorithm again effectively reconstructs the images producing a nearly perfect duplicate of the input reflectivity function as shown in Fig. 3.



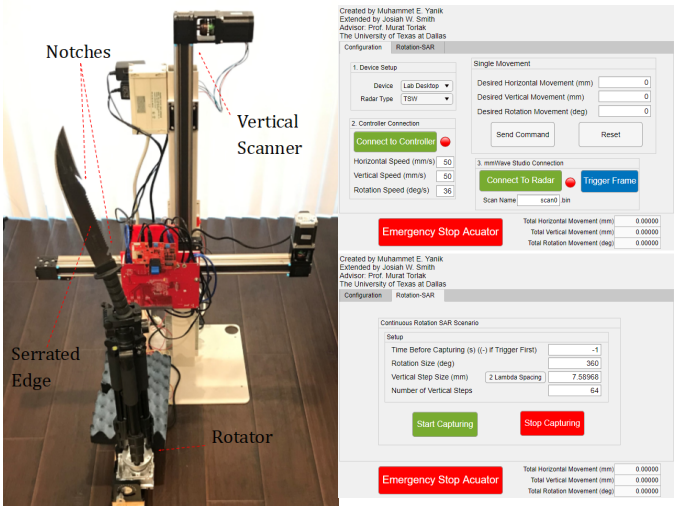
(a) (b)

Fig. 3: (a) 3-D input grid of points reflectivity function, and (b) 3-D reconstructed image.

With successful verification of the algorithm in simulation, a custom prototype R-ISAR scanner is built to experimentally capture data and test the algorithm's image quality on real echo data.

VI. Experimental Setup

A cylindrical aperture is synthesized by mechanically moving a linear MIMO array continuously along a vertical track pattern, and rotating the target as shown in Fig. 1. The system consists of Texas Instruments (TI) IWR1443-Boost, mmWave-Devpack, and TSW1400 mounted on a 2-D vertical and horizontal scanner as shown in Fig. 4a. For this application, only the vertical motion is used. The scanned object is mounted on the rotator. All mechanical motions are controlled by stepper drivers and embedded microcontrollers. The entire setup is controlled by a custom MATLAB graphical user interface (GUI), shown in Fig. 4b.



(a) (b)

Fig. 4: (a) Custom-built rotation-ISAR scanner, and (b) MATLAB GUI.

TABLE I: R-ISAR Radar Parameters

	R_0	Δ_θ	θ_{max}	N_y	D_y^S	Δ_y
MIMO	0.25 m	0.036°	360°	64	484.8 mm	2λ
SISO	0.25 m	0.036°	360°	512	484.8 mm	$\lambda/4$

VII. Imaging Results

The 2-D vertical and rotational scan is performed by the prototype scanner. The large knife shown in Fig. 4a is mounted to the rotator at an angle and scanned. Note the knife's notches and serrated edge.

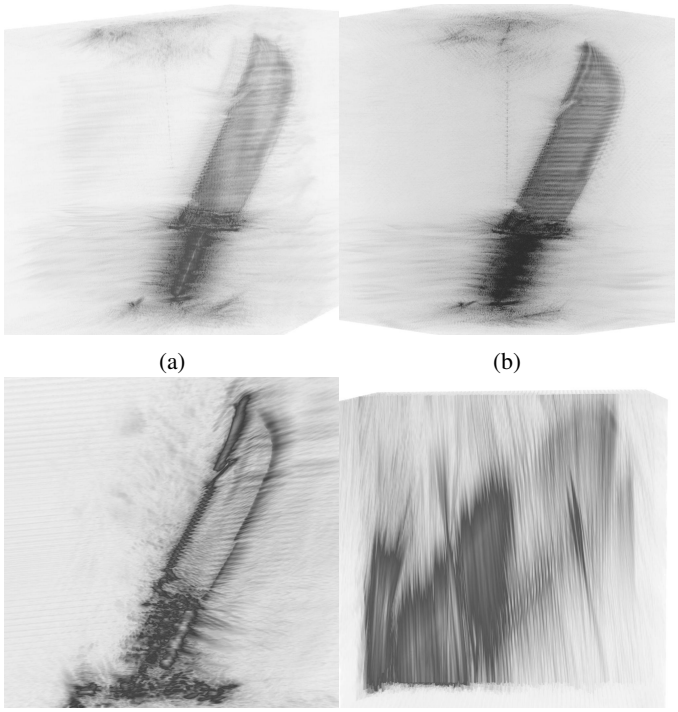
A. SISO Imaging Results

In the first experiment, a single transceiver pair is used to simulate a full-duplex SISO transceiver using the full 4 GHz bandwidth. Since the MIMO virtual array consists of 8 equally spaced virtual elements spanning 2λ, the SISO scan will have a vertical spacing of $\lambda/4$ and will require 512 vertical captures to replicate the MIMO scan, drastically increasing the scanning time. All other parameters will remain the same, as shown in Table I.

Neglecting the multistatic-to-monostatic conversion, the proposed algorithm is implemented on the SISO echo data to produce a holographic image, as shown in Fig. 5a. While the knife's intricacies are clearly visible in the high-resolution image, the entire scan took nearly two and a half hours to complete. Next, we exploit the MIMO virtual array to drastically reduce the scanning time. The key novelty of the proposed algorithm is the increase in scanning efficiency by leveraging a MIMO topology without increasing the computational complexity.

B. MIMO Imaging Results

Now, the knife is scanned again, this time using 2 transmitters and 4 receiver antennas on the TI IWR1443-Boost and again 4 GHz bandwidth. After calibration to remove instrument delay, the echo data is processed by the proposed R-ISAR MIMO 3-D holographic imaging algorithm and an image is produced. Again, a 3-D rendering is included below, in Fig. 5b. The MIMO scan only took less than twenty minutes to complete, a significant reduction in comparison to the SISO scan without degrading the image quality.



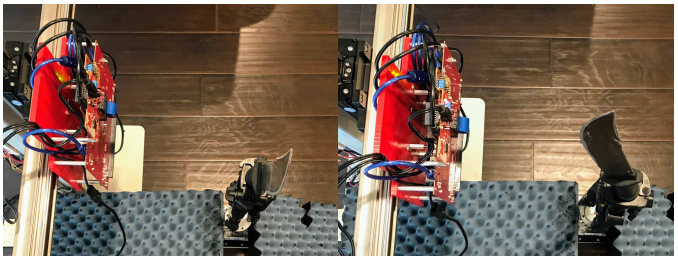
(a) (b) (c) (d)

Fig. 5: Comparison Reconstructed Images: (a) SISO R-ISAR 3-D MIP, (b) MIMO R-ISAR 3-D MIP, (c) MIMO R-SAR with knife parallel to scanning plane, (d) MIMO R-SAR with knife perpendicular to scanning plane

C. Rectilinear MIMO-SAR Comparison

The R-ISAR regime is more suitable for near-field target scanning than rectilinear (planar) SAR regime because the target is scanned by the radar from all sides. To demonstrate the advantage of R-ISAR over SAR, a rectilinear 2-D scan is performed across the x and y axes of the mechanical scanner, without rotating the knife, using the parameters in Table II, where D_S^x is the size of the horizontal scan. Two scenarios are proposed. First, the knife is scanned with its blade parallel to the scanning plane; then, it is scanned again with the blade perpendicular to the scanning plane. Both configurations are shown in Fig. 6. When the knife is parallel, the resulting image is a high quality reconstruction resembling the blade (Fig. 5c). But when it is perpendicular, the image of the blade is quite poor (Fig. 5d).

Examining all four images in Fig. 5 qualitatively, the image quality of the MIMO R-ISAR image (Fig. 5b) appears to be the same, if not better, than that of the SISO R-ISAR



(a) Knife Parallel (b) Knife Perpendicular

Fig. 6: Knife parallel (a) and perpendicular (b) to scanning plane for rectilinear SAR scans.

TABLE II: MIMO SAR Radar Parameters

D_x^S	Δ_x	N_y	D_y^S	Δ_y
450 mm	0.5 mm	64	484.8 mm	2λ

scan (Fig. 5a). For the MIMO-SAR (planar) scans, while the image quality of the knife for the perpendicular scan (5c) is comparable to the MIMO R-ISAR image, targets are not always ideally parallel to the scanning plane. When the knife is scanned perpendicular to the scanning plane (Fig. 5d), the quality of the reconstructed image of the blade is degraded substantially since the cross range resolution of the SAR regime is significantly less than its R-ISAR counterpart. For the R-ISAR regime, the image quality is independent of the orientation of the object with respect to the rotation axis since the target is rotated a full 360°. Therefore, the R-ISAR regime captures the scanned target both parallel and perpendicular at different times throughout the rotation. Lastly, considering time required for each scanning regime, as shown in table III, the R-ISAR mode offers the fastest scanning and best image performance. By using the algorithm proposed in this paper, we were able to drastically reduce the scanning time while maintaining the computational efficiency of the monostatic algorithms.

TABLE III: Scanning Times

SISO R-ISAR	MIMO R-ISAR	MIMO SAR
1032 s	8202 s	1740 s

VIII. Conclusion

In this paper, we developed an efficient MIMO rotational-ISAR 3-D holographic imaging system based on the single pixel polar formatting algorithm and multistatic-to-monostatic conversion. The algorithm successfully pairs the scanning efficiency of MIMO systems with the computational efficiency of monostatic reconstruction algorithms. Additionally, we developed a complete, robust 3-D imaging system consisting of a vertically scanned MIMO radar and a rotator to rotate the target object. Our system fully integrates scanning scenario setup, data collection and calibration, algorithm implementation, and image inspection for a complete, efficient 3-D holographic imaging platform. Using this prototype system, high-resolution images are captured to demonstrate the effectiveness of this system for 3-D scene reconstruction and verify the MIMO R-ISAR algorithm's performance in comparison to its SISO R-ISAR and MIMO-SAR counterparts. Additionally, the MIMO R-ISAR regime is shown to outperform the MIMO-SAR regime with improved range and cross-range resolution, deeming R-ISAR more suitable for near-field imaging applications. The algorithm and system demonstrate high-performance near-field imaging using MIMO rotational inverse synthetic aperture radar.

Acknowledgment

This work is supported by Semiconductor Research Corporation (SRC) task 2712.029 through The University of Texas at Dallas' Texas Analog Center of Excellence (TxACE).

References

[1] M. E. Yanik and M. Torlak, "Millimeter-wave near-field imaging with two-dimensional SAR data," in *Proc. SRC Techcon*, no. P093929, Austin, TX, USA, Sep. 2018.

[2] L. Qiao, Y. Wang, Z. Shen, Z. Zhao, and Z. Chen, "Compressive sensing for direct millimeter-wave holographic imaging," *Appl. Opt.*, vol. 54, no. 11, pp. 3280–3289, Apr 2015. [Online]. Available: http://ao.osa.org/abstract.cfm?URI=ao-54-11-3280

[3] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Near-field three-dimensional radar imaging techniques and applications," *Appl. Opt.*, vol. 49, no. 19, pp. E83–E93, Jul. 2010.

[4] Sheen, McMakin, and Hall, "Three-dimensional millimeter-wave imaging for concealed weapon detection," *IEEE Trans. on Microwave Theory and Techniques*, vol. 49, no. 9, pp. 1581–1592, Sep. 2001.

[5] X. Zhuge and A. G. Yarovoy, "Three-dimensional near-field mimo array imaging using range migration techniques," *IEEE Transactions on Image Processing*, vol. 21, no. 6, pp. 3026–3033, 2012.

[6] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Efficient terahertz wide-angle nufft-based inverse synthetic aperture imaging considering spherical wavefront," *Sensors (Basel, Switzerland)*, vol. 16, no. 12, p. 2120, Dec 2016, 27983618[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/27983618

[7] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "An efficient algorithm for mimo cylindrical millimeter-wave holographic 3-d imaging," *IEEE Transactions on Microwave Theory and Techniques*, vol. 66, no. 11, pp. 5065–5074, 2018.

[8] D. M. Sheen, D. L. McMakin, T. E. Hall, and R. H. Severtsen, "Real-time wideband cylindrical holographic surveillance system."

[9] J. Laviada, A. Arboleya-Arboleya, Y. Álvarez, B. González-Valdés, and F. Las-Heras, "Multiview three-dimensional reconstruction by millimetre-wave portable camera," *Scientific reports*, vol. 7, no. 1, pp. 6479–6479, Jul 2017, 28743908[pmid]. [Online]. Available: https://pubmed.ncbi.nlm.nih.gov/28743908

[10] M. E. Yanik, D. Wang, and M. Torlak, "3-d mimo-sar imaging using multi-chip cascaded millimeter-wave sensors," in *2019 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*, 2019, pp. 1–5.

[11] M. E. Yanik and M. Torlak, "Near-field MIMO-SAR millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31 801–31 819, Mar. 2019.

[12] J. Detlefsen, A. Dallinger, S. Huber, and S. Schelkshorn, "Effective reconstruction approaches to millimeter-wave imaging of humans," 01 2005.

[13] X. Zhuge and A. G. Yarovoy, "A sparse aperture mimo-sar-based uwb imaging system for concealed weapon detection," *IEEE Transactions on Geoscience and Remote Sensing*, vol. 49, no. 1, pp. 509–518, 2011.

[14] J. K. Gao, Y. L. Qin, B. Deng, H. Q. Wang, J. Li, and X. Li, "Terahertz wide-angle imaging and analysis on plane-wave criteria based on inverse synthetic aperture techniques," *Journal of Infrared, Millimeter, and Terahertz Waves*, vol. 37, no. 4, pp. 373–393, 2016. [Online]. Available: https://doi.org/10.1007/s10762-016-0249-x

IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 18, 2020, accepted January 5, 2021, date of publication January 13, 2021, date of current version January 20, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3051454

Improved Static Hand Gesture Classification on Deep Convolutional Neural Networks Using Novel Sterile Training Technique

JOSIAH W. SMITH, (Student Member, IEEE), SHIVA THIAGARAJAN, RICHARD WILLIS, YIORGOS MAKRIS, (Senior Member, IEEE), AND MURAT TORLAK, (Senior Member, IEEE)
Department of Electrical and Computer Engineering, The University of Texas at Dallas, Richardson, TX 75080, USA

Corresponding author: Josiah W. Smith (josiah.smith@utdallas.edu)

This work was supported in part by Texas Instruments through the Foundational Technology Research Centre and the Texas Analog Center of Excellence.

ABSTRACT In this paper, we investigate novel data collection and training techniques towards improving classification accuracy of non-moving (static) hand gestures using a convolutional neural network (CNN) and frequency-modulated-continuous-wave (FMCW) millimeter-wave (mmWave) radars. Recently, non-contact hand pose and static gesture recognition have received considerable attention in many applications ranging from human-computer interaction (HCI), augmented/virtual reality (AR/VR), and even therapeutic range of motion for medical applications. While most current solutions rely on optical or depth cameras, these methods require ideal lighting and temperature conditions. mmWave radar devices have recently emerged as a promising alternative offering low-cost system-on-chip sensors whose output signals contain precise spatial information even in non-ideal imaging conditions. Additionally, deep convolutional neural networks have been employed extensively in image recognition by learning both feature extraction and classification simultaneously. However, little work has been done towards static gesture recognition using mmWave radars and CNNs due to the difficulty involved in extracting meaningful features from the radar return signal, and the results are inferior compared with dynamic gesture classification. This article presents an efficient data collection approach and a novel technique for deep CNN training by introducing ''sterile'' images which aid in distinguishing distinct features among the static gestures and subsequently improve the classification accuracy. Applying the proposed data collection and training methods yields an increase in classification rate of static hand gestures from 85% to 93% and 90% to 95% for range and range-angle profiles, respectively.

INDEX TERMS Convolutional neural networks, deep learning, hand gesture recognition, millimeter-wave radar, sterile training.

I. INTRODUCTION

Accurately classifying human hand gestures has recently received significant attention as non-contact human-computer interaction (HCI) sensors become increasingly prevalent and desirable. Many efforts have been done towards classifying moving (dynamic) hand gestures and non-moving (static) hand gestures using optical cameras and many different classifiers [1]. Applications of static gesture classification include augmented/virtual reality (AR/VR) [2], human-computer interaction [3], and even medical applications for range of motion and therapeutic applications [4]. Such optical systems offer high-resolution two-dimensional (2-D) images but have innate drawbacks requiring specific lighting conditions and lacking depth information. Some solutions have investigated the use of an RGB-D depth camera [5], but these devices suffer under sunlight, restricting their usage to indoors only [2]. On the other hand, small form-factor millimeter-wave (mmWave) frequency-modulated-continuous-wave (FMCW) radar offers high-resolution depth information but does not have the cross-range resolution of an optical camera. mmWave radars are advantageous over optical solutions, due to the semi-penetrative nature of the electromagnetic (EM) at the wavelengths in the mmWave frequency range and independence from ambient temperature effects, allowing for fine measurements in non-ideal lighting and temperature

The associate editor coordinating the review of this manuscript and approving it for publication was Anubha Gupta.

environments including occlusion, fog, indoor/outdoor, etc. Additionally, FMCW mmWave radar allows for simultaneous gesture classification and localization. High-resolution spatial information reflected from a human hand is embedded in the FMCW return signal. However, due to the nature of the FMCW radar as a time-of-flight (ToF) sensor and hardware size limitations, an off-the-shelf radar device cannot reconstruct an image reminiscent of a human hand, meaningful to the human eye. Thus, a deep convolutional neural network (CNN) approach is commonly adopted to classify dynamic gestures from radar return signals, after some preprocessing [6]. Further, extracting meaningful features from the radar return signal is a key step towards accurately classifying hand gestures. As such, recent work on mmWave sensors for hand gesture recognition has been limited to dynamic hand gestures focusing on Doppler and micro-Doppler features [7]–[10] with little attention being paid to the static gesture case [11] due to the low classification rates in such applications. Prior research towards impulse-radio ultra-wideband (IR-UWB) [11]–[13] and Doppler radar-based gesture recognition [7]–[9] using CNNs has shown promising results. However, with the exception of [11], gesture recognition on IR-UWB and Doppler radar has been limited to dynamic, moving gestures. IR-UWB and Doppler sensors are capable of classifying dynamic gestures with ease as motion is easily visible in the Doppler spectrum, whereas both these approaches suffer for static gestures as the distinct features to each gesture class are much more difficult to extract from a stationary human hand. Doppler radar employs the Doppler shift principle to provide relative velocity information of gestures but does not provide spatial features necessary for classifying stationary, static gestures. Therefore, for the problem of static gesture recognition, Doppler radar does not perform well as the hand remains stationary, thus making features of the hand reflection quite difficult to extract from a Doppler radar return signal. On the other hand, IR-UWB sensors act similarly to mmWave FMCW radars and must overcome the same inherent difficulty of classifying the higher-dimensional hand pose from the lower-dimensional radar return signal as discussed later. In this paper, we introduce novel techniques for data collection and CNN training intended to overcome the aforementioned limitations of mmWave sensors for static gesture recognition by providing distinct, meaningful features of each hand gesture thus aiding the CNN learning process.

The rest of this article is formatted as follows. In Section II, we briefly overview the FMCW radar signal model and key concepts helpful in developing an intuition for the static gesture classification problem. In Section III, the measurement setup is discussed and the novelty of "sterile" data for FMCW radar is introduced. Section IV overviews some basics of deep convolutional neural networks and highlights the network architecture employed in our implementation. In Section V, classification results are shown and discussed, followed finally by conclusions.

II. FMCW RADAR SIGNAL MODEL

Understanding the FMCW radar signal model provides several key insights into the difficulty of the static hand gesture recognition problem space. The frequency-modulated-continuous-wave signal model is well explored in the literature [14] and briefly overviewed here to provide some insight into the inherent challenges of static gesture recognition using mmWave FMCW sensors.

A. FMCW BEAT SIGNAL

We will begin by considering a single bistatic FMCW transceiver, whose transmitter and receiver are positioned at the points $(0,y_T,Z_0)$ and $(0,y_R,Z_0)$ in (x,y,z) space, respectively, and one stationary ideal point reflector in the scene with reflectivity σ located at the point (x_0,y_0,z_0). The radar transceiver is positioned on the x'-y' plane located at $z = Z_0$ from the point target.

First, the FMCW device generates what is known as a chirp signal, which can be modeled as a complex sinusoidal whose frequency increases linearly with time as

$$m(t) = e^{j2\pi(f_0 t + \frac{1}{2}Kt^2)}, \quad 0 \le t \le T, \tag{1}$$

where f_0 is the instantaneous frequency at the time $t = 0$, K is the chirp slope, and T is the chirp duration in fast time. The chirp bandwidth can easily be computed using $B = KT$ [15].

The chirp signal $m(t)$ is transmitted by the transmit antenna, reflected off of the ideal point reflector, and returned to the receive antenna as a scaled and time-delayed version of the transmitted signal. Taking round-trip amplitude decay into account, the received signal can be modeled as

$$\hat{m}(t) = \sigma \frac{m(t-\tau)}{R_T R_R} = \frac{\sigma}{R_T R_R} e^{j2\pi(f_0(t-\tau)+\frac{1}{2}K(t-\tau)^2)}, \tag{2}$$

where τ is the round-trip time delay [16] and the values R_T and R_R (see Fig. 1) can be computed by

$$R_T = \sqrt{x_0^2 + (y_0 - y_T)^2 + (z_0 - Z_0)^2}, \tag{3}$$

$$R_R = \sqrt{x_0^2 + (y_0 - y_R)^2 + (z_0 - Z_0)^2}. \tag{4}$$

Therefore, the round trip time delay τ can be computed by

$$\tau = \frac{R_T + R_R}{c}, \tag{5}$$

where c is the speed of light.

Now, the received signal $\hat{m}(t)$ is demodulated with the transmitted signal $m(t)$ yielding what is known as the IF signal or FMCW beat signal, written as

$$s(t) = \frac{\sigma}{R_T R_R} e^{j2\pi(f_0\tau + K\tau - \frac{1}{2}K\tau^2)} \tag{6}$$

The last phase term of (6) is called the residual video phase (RVP) term and is known to be negligible [17]. Finally, the beat signal can be simplified to the expression

$$s(y_T, y_R, k) = \frac{\sigma}{R_T R_R} e^{jk(R_T + R_R)}, \tag{7}$$

where $k = 2\pi f/c$ is the wavenumber corresponding to the instantaneous frequency $f = f_0 + Kt$ for $t \in [0, T]$.

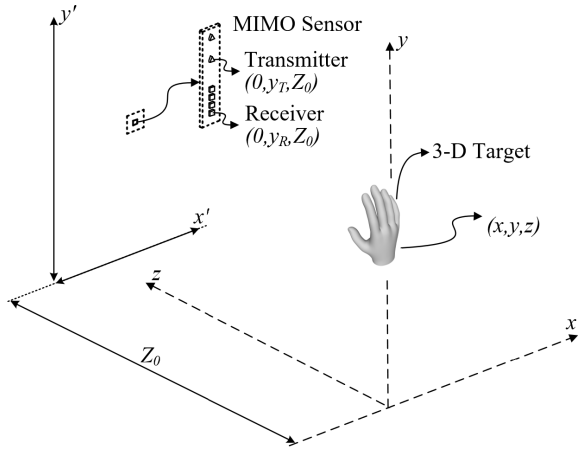


FIGURE 1. A MIMO radar sensor with transmitter and receiver antenna elements located at $(0,y_T,Z_0)$ and $(0,y_R,Z_0)$, respectively captures the return signal from a three-dimensional (3-D) target whose reflectivity function is $\sigma(x,y,z)$.

B. MULTISTATIC-TO-MONOSTATIC CONVERSION

The result in (7) shows the FMCW beat signal from a single point reflector using a multistatic antenna array where the transmitter and receiver are not co-located. To ease the subsequent signal processing, it is desirable to approximate this multistatic echo signal to a monostatic version. This approximation already has been explored in the literature and we will simply use the result derived in [17], [18]. The multistatic-to-monostatic conversion is known as a simple phase adjustment applied to each transceiver pair as

$$\hat{s}(y',k) = s(y_T, y_R, k)e^{-jk\frac{d_y^2}{4Z_0}}, \tag{8}$$

where d_y is the small separation between the transmitter and receiver and Z_0 is an approximate distance from the radar to the target.

Now, the monostatic approximation yields a beat signal whose virtual antenna positions are at the midpoint of each of the transceiver pairs. Taking y' as these virtual antenna locations, the virtual monostatic signal can be approximated by a simplified version of (7) as

$$\hat{s}(t) \approx \frac{\sigma}{R_0^2}e^{j2kR_0}, \tag{9}$$

where R_0 is the distance between each virtual antenna element and the point reflector and is expressed as

$$R_0 = \sqrt{x_0^2 + (y_0 - y')^2 + (z_0 - Z_0)^2}. \tag{10}$$

Now, the range of the target is clearly embedded in the frequency of the beat signal.

C. FMCW RANGE-ANGLE ANALYSIS

Considering the ideal point reflector described previously and a uniform linear monostatic array along the y-axis, the range and range-angle profiles can be computed and used to localize the point reflector. First, as evident in (9) and (11), the frequency of the beat signal corresponds directly to the range of the target. Thus, the range profile can be generated by performing a fast-Fourier transform (FFT) along the k-domain of the beat signal. The minimum resolvable distance between two targets, or range resolution, can be easily computed as $\Delta z_{min} = c/(2B)$ [19].

Similarly, as discussed in [20], the angular profile of the target scene can be computed by performing an FFT across the spatial y domain. To avoid aliasing in the angle domain, by Nyquist theorem, the maximum theoretical distance between elements is $\lambda/4$, where λ is the wavelength. However, even after applying both range and angle FFTs, the resulting range-angle profile only describes the intensity in two dimensions and is limited by the number of antenna elements and bandwidth.

In this work, we will employ and compare both range and range-angle analysis to preprocess the echo signals before using them for CNN training or classification.

D. MODELING A DISTRIBUTED TARGET

For gesture recognition, a human hand can be mathematically modeled as a distributed target consisting of continuously varying reflectivity across space. Understanding how the radar captures such target scenes provides an intuition into the difficulty of the hand gesture recognition problem.

Assuming a simple linear multistatic array along the y-axis, such as the depiction in Fig. 1, after the aforementioned conversion, the return signal from a distributed target can be modeled as the superposition of the echo signals from each of the target coordinates scaled by the target's reflectivity function $\sigma(x,y,z)$. The beat signal from each virtual monostatic transceiver at the positions y' can be expressed as

$$s(y',k) = \iiint \frac{\sigma(x,y,z)}{R^2}e^{j2kR}dxdydz. \tag{11}$$

where R is the radial distance from each virtual monostatic element located at the positions y' to each point in the distributed target domain as

$$R = \sqrt{x^2 + (y - y')^2 + (z - Z_0)^2}. \tag{12}$$

If samples are taken throughout the x'-y' plane, the reflectivity function can be reconstructed by inverting (11); however, for an application such as hand gesture recognition, the transceiver elements only span a small space along the y'-axis. This model provides insight into the simultaneous plausibility and difficulty of the static gesture recognition problem on FMCW radar.

Embedded in the beat signal are high-resolution spatial features describing the shape of the target or static gesture being performed, meaning different hand poses or static gestures have distinct echo signals unique to that gesture. However, the target scene, or hand, cannot be analytically reconstructed as a three-dimensional (3-D) image and used to easily classify the gestures using traditional optical image approaches. Thus, classifying static hand gestures involves attempting to learn a 3-D pattern (the hand pose in three dimensions) from 2-D radar data.

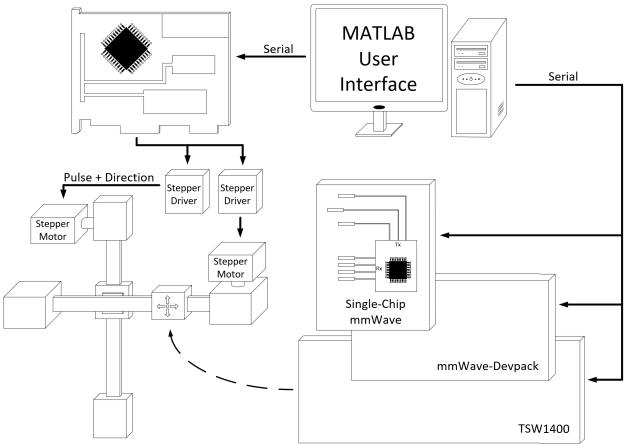


FIGURE 2. Two dimensional *x*-*y* rectangular scanner system diagram.

Further, another issue inherent to the hand gesture problem is the small radar cross-section (RCS) of the human hand resulting in a low signal-to-noise-ratio (SNR), as discussed later. Even with a large amount of data, since the RCS of the hand is low, the features unique to each gesture class are not pronounced. As a result, the CNN has difficulty discerning meaningful features for each gesture. Our proposed method using ''sterile'' data aims to overcome this deficiency in the training data and will be discussed in detail in the next section.

III. MEASUREMENT SETUP

For any problem using the supervised learning approach to deep learning, the availability of meaningful and diverse data is crucial to building an accurate and well-generalized model.

A. MECHANICAL SCANNER

To efficiently gather data from many perspectives, we first design a 2-D mechanical scanning system capable of positioning the radar anywhere within a 0.5 m $\times$ 0.5 m square, as shown in Fig. 2. The entire system is controlled by a custom-built MATLAB graphical user interface (GUI) hosted by a desktop computer. An AMC4030 motion controller receives commands over serial and controls the stepper motors via stepper drivers, accurately moving the radar to the desired location. Each stepper motor is mounted to a linear belt-driven rail with a usable length of 0.5 m. The radar under test is an IWR1443BOOST, which is an off-the-shelf automotive radar from Texas Instruments (TI). As shown in the system diagram, the radar board is oriented with its MIMO array aligned vertically. The IWR1443BOOST has 3 transmit (Tx) antennas and 4 receive (Rx) antennas, but this work will exclusively use the 2 Tx and 4 Rx antennas which form a linear MIMO array whose virtual elements are separated by $\lambda/4$. The operating frequency is 77 GHz and the bandwidth is 4 GHz. The radar board is attached to a booster, the TI mmWave-Devpack, and a TI TSW1400 data capture card. All three radar boards are controlled via TI mmWave Studio, which receives commands from the MATLAB GUI.

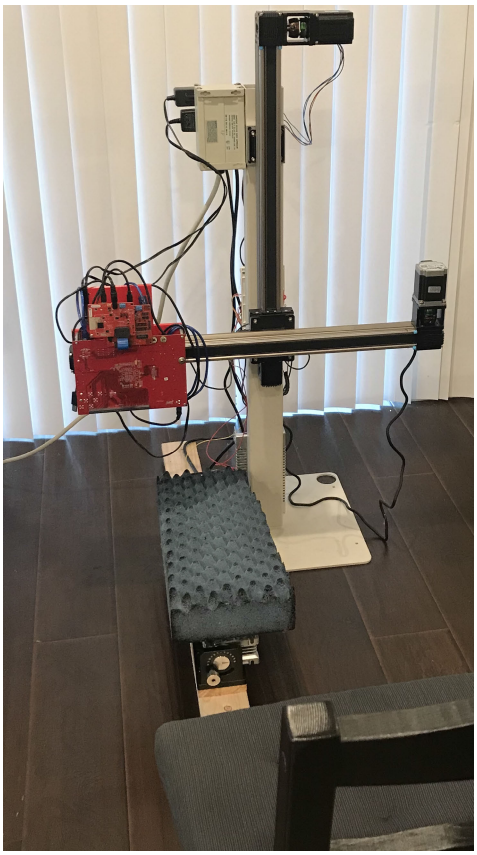
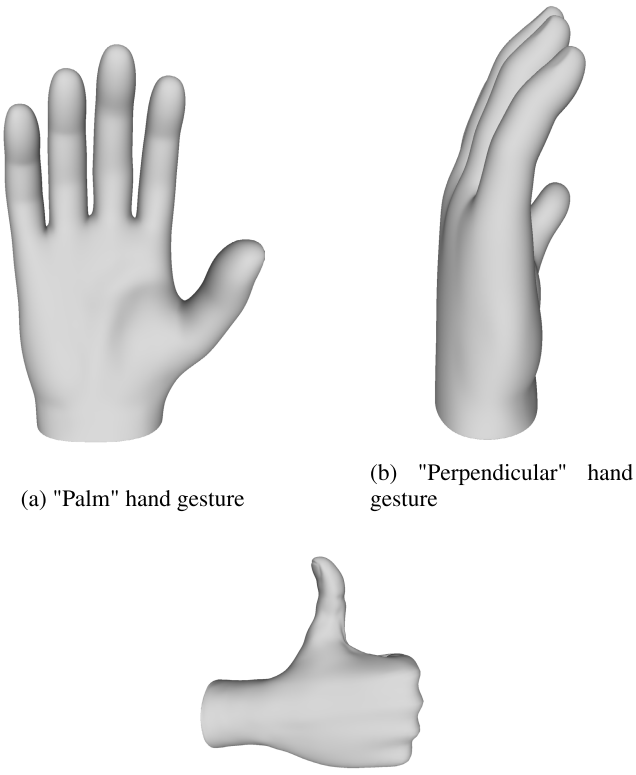


FIGURE 3. Two dimensional *x*-*y* rectangular scanner system with chair for test subject to sit in.

As such, the user has full control over the radar position, settings, timing, etc. directly from the custom GUI.

The novel mechanical scanning system allows for the efficient capture of static hand gestures from many locations, as shown in Fig. 3. A test subject simply keeps their hand in the correct pose and position and the radar is scanned both horizontally and vertically capturing many different perspectives of the hand gesture.

For the rest of this article, the scanning range will be limited to a square with sides of 0.25 m, and the subject's hand will be placed between 0.25 m and 0.55 m from the radar. The subject will hold their hand out in front of them performing the gesture with their hand kept away from their torso so as to avoid occluding the hand in the torso peak's side-lobes. Further, the user will sit in a chair located 1 m from the radar while performing the gestures. Multiple test subjects are used to collect data varying in height, weight, torso size, arm length, hand size, and gender to diversify the dataset. Since the radar data are captured at locations throughout the x'-y' plane, even though the subject does not move their hand, the dataset will consist of many unique ''views'' of the hand as if the hand is positioned at many locations relative to the radar. To the authors' knowledge, this article is the first attempt to use a 2-D scanner to collect static hand gesture data on mmWave radar. The mechanical scanner is employed to collect a diverse dataset consisting of multiple perspectives



(a) "Palm" hand gesture

(b) "Perpendicular" hand gesture

(c) "Thumbs-up" hand gesture

FIGURE 4. **Three static hand gestures from the perspective of the radar.**

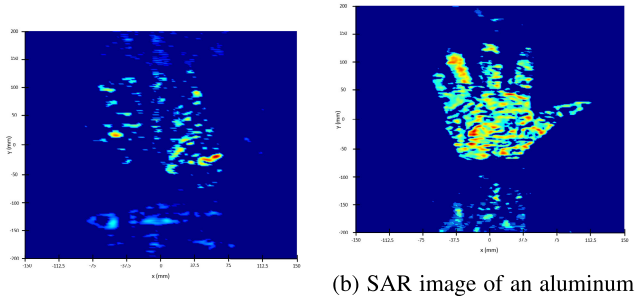
of the hand gestures; however, the problem of human hand reflectivity and feature prominence remain. The novel data collection technique is next extended in attempt to overcome these issues.

B. CHALLENGES

We proceed to test the proposed system using three distinct static hand gestures which we label ''palm'', ''perpendicular'', and ''thumbs-up,'' as shown in Fig. 4. For the ''palm'' gesture, the user places their hand with the palm facing the radar. The ''perpendicular'' gesture involves the subject's hand oriented perpendicular with the x'-y' plane and thumb facing away from the radar. And, the ''thumbs-up'' gesture requires the subject to face the back of their hand towards the radar with all fingers abducted except the thumb which points upward.

As mentioned previously, the RCS of the human hand is problematically small in comparison to noise and propagation effects. Comparing the range profiles of the different gestures, the differences are mostly indistinguishable to the human eye. Even though a peak exists in the range FFT at the distance corresponding to the human hand, the features of the gesture reflected back to the radar are not sharply defined and are centered at different places on the human hand.

To demonstrate this phenomenon, a synthetic radar aperture (SAR) approach is temporarily adopted to reconstruct



(a) SAR image of a real human hand at 0 25 m from the radar

(b) SAR image of an aluminum cutout hand at 0.25 m from the radar

FIGURE 5. **Comparison of the reconstructed SAR images from the real human hand and the aluminum cutout of the human hand demonstrating the low RCS of the human hand. For consistency, both the human hand and the aluminum cutout are placed on top of a tripod, which contributes the reflections visible beneath the hand.**

an image of the human hand using the methods described in [19], [21]. It is important to note that the images shown in Fig. 5 are not the data used to train and validate the CNN. These images require all the data from the entire horizontal and vertical scan, which takes approximately 5 minutes to complete. The data used to train the network are discussed in greater detail later.

The reconstructed image of the human hand, Fig. 5a, shows a poor image of the hand due to the low RCS. For comparison, a SAR image is also reconstructed using an aluminum cutout in the shape of the hand attached to demonstrate an ideal hand target, Fig. 5b. For the remainder of this article, we call data collected using aluminum cutouts ''sterile'' data. This empirical analysis uncovers the innate difficulty in classifying hand gestures from the radar beat signal. Even combining thousands of radar return signals to construct the SAR image, the hand is barely visible and the gesture is difficult to recognize. From these images, we can infer that the features from a human hand contained in a single beat signal reflected are not pronounced and have a quite low magnitude compared to the surroundings, noise, etc. On the contrary, from Fig. 5b, the aluminum cutout demonstrates a high SNR, meaning the features of the gesture are much more prominent and consistent for each static gesture. The novel data collection strategy proposed in this article consists of capturing data from many perspectives using a 2-D mechanical scanner from both ''real'' human hands and ''sterile'' aluminum cutouts, in attempt to improve classification accuracy.

From these observations, we pose several key questions this article aims to answer. Does the radar return signal from an aluminum cutout of a static gesture contain more pronounced, meaningful, and consistent features uniquely describing each gesture compared to the return signal from a human hand? If so, can these ''sterile'' radar data captured from aluminum cutouts be used to improve the accuracy of a CNN classifier? Specifically, will a training set consisting of both human hand data and ''sterile'' data provide more easily learnable features to the CNN?

IV. DEEP CONVOLUTIONAL NEURAL NETWORKS

Before answering those exploring those questions further, we first will overview the proposed classifier. In data-driven detection problems, there are two fundamental steps to constructing a robust classifier: feature extraction and classification. Many methods have been applied to extract features from raw data including handcrafted features [10], [22], linear predictive coding [23], empirical mode decomposition [24], principal component analysis [25], and more. Similarly, many different classification techniques have been adopted such as k-nearest-neighbors (KNN) [26], support vector machines [10], [11], dynamic Bayesian networks [27], etc. However, in recent years, the preferred approach for many classification and regression problems is the deep convolutional neural network [9]. The concept of deep learning combines feature extraction and classification into a singular step. Now, the feature extraction and classification are simultaneously modeled as a single optimization problem. CNNs have been widely used for image classification and are popularly employed for dynamic gesture recognition on Doppler radar [6], [7]. Unique from most conventional machine learning algorithms, CNNs adopt a multi-layer approach with interconnected neurons meant to imitate the human brain. Further, due to recent advancements in parallel computing, specifically in graphics processing units (GPUs), training CNNs with complex architectures consisting of many layers has become increasingly feasible.

The fundamental building blocks of a CNN are convolution layers and nonlinear activation functions. The convolution layers extract features by convolving an array of weights over the input image. The weights are updated every iteration by the back-propagation algorithm used in conjunction with stochastic gradient descent to maximize the classification rate and minimize the loss. After the convolution layer, a nonlinear activation function is applied. Most deep CNNs employ a Rectified Linear Unit (ReLU) defined as $f(x) = \max(0, x)$ over the traditional sigmoid function for improved results [28]. By using a nonlinear activation function, the network is able to learn the highly nonlinear complex relationships between the inputs and outputs.

After convolution and activation, pooling layers are often used to downsample the data by either the average or maximum of a local pool. Convolution, ReLU, and pooling layers are connected to form a complex network of neurons and are finally followed by a fully connected layer, which reduces the dimensionality to the known number of classes, and subsequent general perceptron for classification.

A. PREPROCESSING OF INPUT IMAGES

Input data are gathered from the radar and undergo preprocessing before being used for network training and validation. First, the multistatic-to-monostatic conversion in (8) is applied to the complex-valued, MIMO beat signal described in equation (7). Then either the range or range-angle analysis described in Section II-C is performed. In the next section, we compare the results from using only range analysis to those using range-angle analysis. The complex-valued image is of size $8\text{x}N_R$, for the range analysis case, or $M_A\text{x}N_R$, for both range and angle analysis, where M_A is the number of angle FFT bins and N_R is the number of range FFT bins.

Since minute variations in the hand reflectivity are contained in the phase of the radar beat signal, retaining the amplitude and phase of the complex-valued signal is essential for accurate classification. Some work has been done towards complex-valued implementations of CNNs for radar problems such as SAR image classification [29] and enhanced SAR imaging [30], however, we consider an alternative approach to classifying the complex radar return signals. Rather than simply taking the magnitude of each image pixel, the real and imaginary parts of the range or range-angle data sample are layered, forming images of size $8 \times N_R \times 2$ or $M_A \times N_R \times 2$. Now, inherent relations between the magnitude and phase of the radar data are not lost, improving the classification rate.

Additionally, most deep CNN implementations employ an input normalization for each channel for numerical robustness. In this case, however, normalizing the real and imaginary layers effectively ruins the phase interdependence. As such, prior to separation into real and imaginary layers, the complex-valued image is normalized to zero-mean and unit variance. Then, no normalization is applied to the real-valued 3-D arrays.

V. CLASSIFICATION RESULTS

In this section, we use empirical results to affirmatively answer the questions posed in Section III-B demonstrating for effectiveness of our proposed ''sterile'' radar data collection techniques for improving static hand gesture classification using mmWave radar. Supplementing training data with synthetically generated data for convolutional neural networks has proven effective for numerous deep learning problems [31]–[34]. However to our knowledge, this work is the first to use synthetic ''sterile'' hands, in the form of aluminum cutouts, captured by mmWave radar, to improve the classification rate of static hand gestures.

A. TRAINING DATA

As discussed in Section II-C, a 1-D range FFT or 2-D range-angle FFT is applied to each captured beat signal prior to training. For the remainder of this article, we will refer to these datasets as the ''range'' and ''range-angle'' datasets, respectively. Each dataset consists of 80000 samples for each of the three static hand gesture classes of which 40000 are from human hands and 40000 are from aluminum cutouts. These samples are rapidly captured at various locations throughout the 2-D x'-y' plane by moving the radar mounted to the mechanical scanner. Both the test subjects' hands and the aluminum cutouts vary in size and shape pursuant of improving classifier robustness. Eight participants are selected with varying age, height, weight, hand size and shape, arm length, torso size, and gender.

To construct the aluminum cutouts, a simple procedure is adopted. Hands of different sizes and shapes are selected from online sources. These shapes are cutout from a thin aluminum sheet and attached to an equivalently shaped cardboard cutout, allowing for a simple training process for sterile data collection. Similarly, eight aluminum cutouts are prepared for each gesture class and used to collect the data. Each aluminum cutout does not directly correspond to test subject hand; rather, the aluminum cutout sizes and shapes are selected independently. For the range data, FFT is performed across the k-domain yielding the range profile. The region wherein the hand is expected to be placed is selected at a size of 64 range bins. Similarly, for the range-angle dataset, the range FFT is performed followed by an angle FFT of size 16. Once the data are preprocessed, the range dataset images are of size $64 \times 8 \times 2$ and the range-angle images are of size $64 \times 16 \times 2$.

B. NETWORK ARCHITECTURE AND TRAINING

The networks used to classify the hand gesture vary based on the preprocessing applied to the dataset. For the range dataset, convolutional layers with kernel sizes of 13×2 each with 16 filters are each followed by a Rectified Linear Unit. These are connected in series followed by a fully connected layer with 3 output neurons, a softmax layer, and a final classification layer using the cross-entropy loss function. The range-angle dataset employs a network with the same architecture changing only the size of the convolutional layers to 13×4 to account for the larger image sizes. Key to both networks are the complex-valued layering and network architectures. Considering the real and imaginary parts of the radar range data as distinct layers of the image allows the network to identify pixel-to-pixel relationships as well as layer-to-layer relationships, which correspond to phase information of each complex-valued pixel. The architectures of both networks are chosen after close inspection of the feature sizes in the observation image domain in both range and channel/angle in addition to extensive testing to optimize real-time implementation efficiency and classification rate. Both network architectures are shown in Fig. 6.

For training, each dataset is split into 90% for training and 10% for validation, where all the ''sterile'' data is contained in the training dataset and the validation dataset consists exclusively of randomly selected real hand gestures. Of the training data, 55.6% are captures of the aluminum cutouts and 44.4% are from real human hands. In this way, the sterile data is used to supplement the training dataset but is not included in the validation dataset. To ensure consistency, we set aside the randomly selected captures of the validation dataset to be reused for each experiment, numbering 8000 in total.

C. IMPROVED CLASSIFICATION WITH ''Sterile'' DATA

To compare against a control, we first train two networks using only the real human hand data. For these networks, we use the 8000 set aside captures as the validation dataset, making the split between training to validation 80% to 20%.

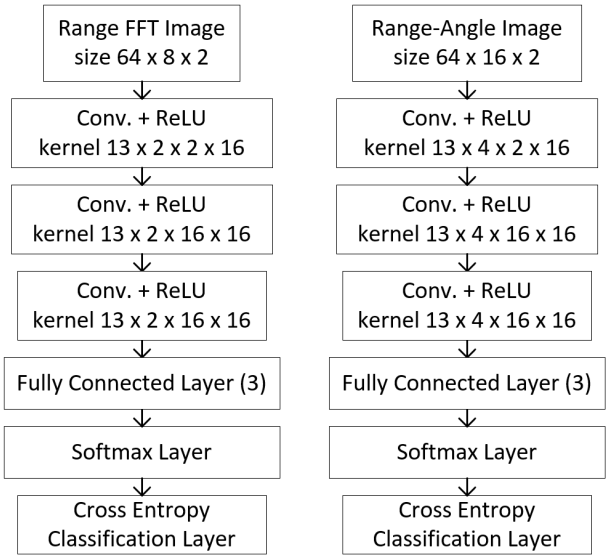


FIGURE 6. The network architecture for the Range FFT CNN and Range-Angle FFT CNN.

TABLE 1. Comparison of classification rate between networks trained with only human hand data (Human Only) and networks trained using sterile data to supplement the real human hand data (Combined).

	Human Only	Combined
Range	84.9%	93.1%
Range-Angle	90.2%	95.4%

After training each network with only real human hand data, the range CNN and range angle CNN have classification rates of 84.9% and 90.2%, respectively. These networks are named ''Human Only'' in Table 1 since they are trained with only range and range-angle profiles from human hands.

Next, two new networks are trained using the complete datasets, consisting of real human hand data supplemented by ''sterile'' data from aluminum cutouts. These networks are dubbed ''Combined'' since they are trained with both real and ''sterile'' images. It is important to note that the ''Combined'' networks are validated with the same validation data as the ''Human Only.'' The only difference is the training dataset used for each network. Once trained, the networks corroborate our hypotheses on training with ''sterile'' data as the classification rates improve to 93.1% and 95.4% for the range and range-angle datasets, respectively.

These results promote an affirmative answer to the questions posed earlier. Namely, the network trained on the dataset consisting of both human hand data and ''sterile'' data is able to learn meaningful features in classifying human hand data more effectively than the network trained using only non-''sterile'' data. Further, our results explicitly demonstrate an increase in classification accuracy by employing the ''sterile'' radar data collection scheme proposed in this article.

Finally, compared to past work in the literature, our proposed method improves upon gesture recognition by using sterile data while offering a solution to the difficult classification problem of static gestures under three-dimensional spatial translation. Kim *et al.* [11] employ a time-domain gesture recognition approach on ultra-wideband impulse-radio radar. Reference [11] considers two scenarios separately: (1) six gestures using human hands 15 cm away from the transceiver and (2) three plaster model gestures rotated at 10° increments. For scenario (1), the hand is kept at a constant position for all captures. Both training and testing are performed using human hand data resulting in a classification rate of 91% using a CNN. In scenario (2), plaster models of each gesture are captured from different perspectives by rotating the plaster model. For this scenario, Kim *et al.* record classification accuracies of above 90% for three gestures, validating using data from the plaster model. Comparatively, our method yields a more robust classifier by including both real human hand reflections and ''sterile'' reflections in the training processes and validating with only human hand data. Rather than creating two distinct classifiers for human and sterile data separately, as discussed in [11], the technique proposed in this paper unites human and sterile data to construct a robust classifier. Further, our approach investigates more diverse scenarios by capturing data from multiple test subjects at many locations.

Extensive work has been done towards dynamic gesture recognition on mmWave radar, Doppler radar, and IR-UWB sensors [6]–[10], [12], [13]; however, this is an entirely separate problem from the problem addressed in this paper as the dynamic gesture case considers only motion. This reduces the dimensionality of the classification to temporal motion features, whereas static gesture recognition on mmWave radar involves classification of a three-dimensional structure using lower dimensional data, as discussed in Section II-D. Thus, our model is trained for the more difficult problem of static gesture classification under spatial translation and demonstrates superior classification accuracy to prior static gesture classification work [11].

VI. CONCLUSION

In this article, we investigated novel data collection and training techniques for improving the classification of static hand gestures using mmWave FMCW radar and convolutional neural networks. A novel data collection technique for static hand gestures is proposed consisting of a radar mounted on a two-dimensional mechanical scanner allowing the efficient collection of large, diverse radar datasets. Then, we examine the innate challenges of static hand gesture recognition and observe the low radar cross-section of static hand gestures, compared to an ideal aluminum cutout of the same shape. From this observation, we hypothesize that if a convolutional neural network is trained with data from both a real human hand and ''sterile'' aluminum cutout, the resulting network will outperform a network trained on human data alone because the features unique to each static gesture are more pronounced in the ''sterile'' data and will be thus easier for the CNN to learn. From this hypothesis, we extend the data collection approach to capture radar range and range-angle profiles of both human hands and aluminum cutouts for use in CNN training.

Three static (non-moving) hand gestures are considered applying both range and range-angle preprocessing. Using deep CNNs and data from human hands only, the classification accuracies for range and range-angle preprocessed data are 85% and 90% respectively. However, using the same data for validation and only changing the training data as described by our hypothesis, the classification rates improve, respectively, to 93% and 95%. The increase in accuracy demonstrates the improvement introduced by the novel ''sterile'' radar data technique never before examined in the literature. Further, the model developed in this article outperforms prior work on static gesture recognition on a more challenging classification problem [11]. Since the CNN model relies on the availability of a large amount of meaningful data, such ''sterile'' and synthetic data acquisition and generation techniques are likely to increase as they have been proven suitable for many classification and regression problems. In future work, we plan to extend this premise to capture more ''sterile'' static gestures and even ''sterile'' dynamic gestures to improve CNN accuracy and robustness.

REFERENCES

[1] J. Baek, J. Kim, and E. Kim, ''Comparison study of different feature classifiers for hand posture classification,'' in *Proc. 13th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2013, pp. 683–687.

[2] Y.-J. Son and O. Choi, ''Image-based hand pose classification using faster R-CNN,'' in *Proc. 17th Int. Conf. Control, Autom. Syst. (ICCAS)*, Oct. 2017, pp. 1569–1573.

[3] M. Matilainen, P. Sangi, J. Holappa, and O. Silven, ''OUHANDS database for hand detection and pose recognition,'' in *Proc. 6th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Dec. 2016, pp. 1–5.

[4] A. Anaz, M. Skubic, J. Bridgeman, and D. M. Brogan, ''Classification of therapeutic hand poses using convolutional neural networks,'' in *Proc. 40th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc. (EMBC)*, Jul. 2018, pp. 3874–3877.

[5] J.-C. Lin and C.-M. Huang, ''3D hand posture tracking with depth gradient estimation on a RGB-D camera,'' in *Proc. IEEE Int. Symp. Consum. Electron. (ISCE)*, Jun. 2013, pp. 109–110.

[6] Y. Kim and B. Toomajian, ''Application of Doppler radar for the recognition of hand gestures using optimized deep convolutional neural networks,'' in *Proc. 11th Eur. Conf. Antennas Propag. (EUCAP)*, Mar. 2017, pp. 1258–1260.

[7] B. Dekker, S. Jacobs, A. S. Kossen, M. C. Kruithof, A. G. Huizing, and M. Geurts, ''Gesture recognition with a low power FMCW radar and a deep convolutional neural network,'' in *Proc. Eur. Radar Conf. (EURAD)*, Oct. 2017, pp. 163–166.

[8] J. S. Suh, S. Ryu, B. Han, J. Choi, J.-H. Kim, and S. Hong, ''24 GHz FMCW radar system for real-time hand gesture recognition using LSTM,'' in *Proc. Asia–Pacific Microw. Conf. (APMC)*, Nov. 2018, pp. 860–862.

[9] Y. Kim and B. Toomajian, ''Hand gesture recognition using micro-Doppler signatures with convolutional neural network,'' *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[10] S. Zhang, G. Li, M. Ritchie, F. Fioranelli, and H. Griffiths, ''Dynamic hand gesture classification based on radar micro-Doppler signatures,'' in *Proc. CIE Int. Conf. Radar (RADAR)*, Oct. 2016, pp. 1–4.

[11] S. Y. Kim, H. G. Han, J. W. Kim, S. Lee, and T. W. Kim, ''A hand gesture recognition sensor using reflected impulses,'' *IEEE Sensors J.*, vol. 17, no. 10, pp. 2975–2976, May 2017.

[12] S. K. Leem, F. Khan, and S. H. Cho, "Detecting mid-air gestures for digit writing with radio sensors and a CNN," *IEEE Trans. Instrum. Meas.*, vol. 69, no. 4, pp. 1066–1081, Apr. 2020.
[13] J. Park and S. H. Cho, "IR-UWB radar sensor for human gesture recognition by using machine learning," in *Proc. IEEE 18th Int. Conf. High Perform. Comput. Commun.; IEEE 14th Int. Conf. Smart City; IEEE 2nd Int. Conf. Data Sci. Syst. (HPCC/SmartCity/DSS)*, Dec. 2016, pp. 1246–1249.
[14] J. W. Smith, M. E. Yanik, and M. Torlak, "Near-field MIMO-ISAR millimeter-wave imaging," in *Proc. IEEE Radar Conf. (RadarConf)*, Sep. 2020, pp. 1–6.
[15] M. E. Yanik and M. Torlak, "Near-field 2-D SAR imaging by millimeter-wave radar for concealed item detection," in *Proc. IEEE Radio Wireless Symp. (RWS)*, Jan. 2019, pp. 1–4.
[16] M. E. Yanik and M. Torlak, "Millimeter-wave near-field imaging with two-dimensional SAR data," in *Proc. SRC Techcon*, Austin, TX, USA, Sep. 2018, pp. 1–6.
[17] M. E. Yanik and M. Torlak, "Near-field MIMO-SAR millimeter-wave imaging with sparsely sampled aperture data," *IEEE Access*, vol. 7, pp. 31801–31819, 2019.
[18] M. E. Yanik, D. Wang, and M. Torlak, "3-D MIMO-SAR imaging using multi-chip cascaded millimeter-wave sensors," in *Proc. IEEE Global Conf. Signal Inf. Process. (GlobalSIP)*, Nov. 2019, pp. 1–5.
[19] D. Sheen, D. Mcmakin, and T. Hall, "Near-field three-dimensional radar imaging techniques and applications," *Appl. Opt.*, vol. 49, no. 19, p. E83, Jul. 2010.
[20] S. Rao. *mmWave Sensors*. Texas Instruments. Accessed: Dec. 10, 2020. [Online]. Available: http://www.ti.com/sensors/mmwave/overview.html
[21] M. E. Yanik, D. Wang, and M. Torlak, "Development and demonstration of MIMO-SAR mmWave imaging testbeds," *IEEE Access*, vol. 8, pp. 126019–126038, 2020.
[22] Y. Kim and H. Ling, "Human activity classification based on micro-Doppler signatures using a support vector machine," *IEEE Trans. Geosci. Remote Sens.*, vol. 47, no. 5, pp. 1328–1337, May 2009.
[23] R. J. Javier and Y. Kim, "Application of linear predictive coding for human activity classification based on micro-Doppler signatures," *IEEE Geosci. Remote Sens. Lett.*, vol. 11, no. 10, pp. 1831–1834, Oct. 2014.
[24] D. P. Fairchild and R. M. Narayanan, "Classification of human motions using empirical mode decomposition of human micro-Doppler signatures," *IET Radar, Sonar Navigat.*, vol. 8, no. 5, pp. 425–434, Jun. 2014.
[25] Y. Kim, "Detection of eye blinking using Doppler sensor with principal component analysis," *IEEE Antennas Wireless Propag. Lett.*, vol. 14, pp. 123–126, 2015.
[26] Q. Wan, Y. Li, C. Li, and R. Pal, "Gesture recognition for smart home applications using portable radar sensors," in *Proc. 36th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2014, pp. 6414–6417.
[27] V. Pavlovic, B. J. Frey, and T. S. Huang, "Time-series classification using mixed-state dynamic Bayesian networks," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, Jun. 1999, pp. 609–615.
[28] X. Glorot, A. Bordes, and Y. Bengio, "Deep sparse rectifier neural networks," in *Proc. 14th Int. Conf. Artif. Intell. Statist.*, Jun. 2011, pp. 315–323.
[29] Z. Zhang, H. Wang, F. Xu, and Y.-Q. Jin, "Complex-valued convolutional neural network and its application in polarimetric SAR image classification," *IEEE Trans. Geosci. Remote Sens.*, vol. 55, no. 12, pp. 7177–7188, Dec. 2017.
[30] J. Gao, B. Deng, Y. Qin, H. Wang, and X. Li, "Enhanced radar imaging using a complex-valued convolutional neural network," *IEEE Geosci. Remote Sens. Lett.*, vol. 16, no. 1, pp. 35–39, Jan. 2019.
[31] E. Richardson, M. Sela, and R. Kimmel, "3D face reconstruction by learning from synthetic data," in *Proc. 4th Int. Conf. 3D Vis. (3DV)*, Oct. 2016, pp. 460–469.
[32] X. Wu, L. Liang, Y. Shi, and S. Fomel, "FaultSeg3D: Using synthetic data sets to train an end-to-end convolutional neural network for 3D seismic fault segmentation," *Geophysics*, vol. 84, no. 3, pp. IM35–IM45, May 2019.
[33] V. Allken, N. O. Handegard, S. Rosen, T. Schreyeck, T. Mahiout, and K. Malde, "Fish species identification using a convolutional neural network trained on synthetic data," *ICES J. Mar. Sci.*, vol. 76, no. 1, pp. 342–349, Jan. 2019.
[34] T. Björklund, A. Fiandrotti, M. Annarumma, G. Francini, and E. Magli, "Automatic license plate recognition with convolutional neural networks trained on synthetic data," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

JOSIAH W. SMITH (Student Member, IEEE) was born in Denver, CO, USA, in 1997. He received the B.S. degree *(summa cum laude)* in electrical engineering from The University of Texas at Dallas, in 2019, where he is currently pursuing the Ph.D. degree in electrical engineering specializing in communications engineering. During the summer of 2020, he developed real-time human–computer interaction algorithms for millimeter-wave (mmWave) radar with imec, USA. His current research interests include new regime radar imaging algorithm development, ultrawideband radar imaging algorithms, terahertz radars, radar perception, computer vision, machine learning, millimeter-wave sensing, and phased array signal processing. He received the Texas Instruments Analog Excellence Graduate Fellowship, in August 2019.

SHIVA THIAGARAJAN received the B.Tech. degree in electrical and electronics engineering from SRM University, Chennai, India, in 2016, and the M.S. degree in electrical engineering from The University of Texas at Dallas, in 2018, where he is currently pursuing the Ph.D. degree in electrical engineering specializing in design optimization and alternative IC design flow methodologies for integrated circuits. His research interests include exploring the synergy between IC designs and machine learning and developing EDA tools for improving design robustness and test methodologies. Apart from IC design, developing machine learning models for object recognition, radar imaging, and graph algorithms form other aspects of his research.

RICHARD WILLIS was born in Knoxville, TN, USA, in 1998. He received the B.S. degree *(summa cum laude)* in electrical engineering with a minor in computer science from The University of Texas at Dallas (UT Dallas), in 2020. At UT Dallas, he engaged in the development of algorithms for real-time human–computer interaction algorithms for millimeter-wave (mmWave) radar. After graduating in the Summer of 2020, he entered the finance industry as a Quantitative Analyst.

YIORGOS MAKRIS (Senior Member, IEEE) received the Diploma degree in computer engineering from the University of Patras, Greece, in 1995, and the M.S. and Ph.D. degrees in computer engineering from the University of California at San Diego, in 1998 and 2001, respectively. After spending a decade on the faculty of Yale University, he joined UT Dallas, where he is currently a Professor of electrical and computer engineering, leading the Trusted and RELiable Architectures (TRELA) Research Laboratory, and the Safety, Security and Healthcare Thrust Leader of the Texas Analog Center of Excellence (TxACE). His research interests include applications of machine learning and statistical analysis in the development of trusted and reliable integrated circuits and systems, with particular emphasis in the analog/RF domain. He serves as an Associate Editor for the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He has served as an Associate Editor for the IEEE INFORMATION FORENSICS AND SECURITY and the *IEEE Design and Test of Computers Periodical* and a Guest Editor for the IEEE TRANSACTIONS ON COMPUTERS and the IEEE TRANSACTIONS ON COMPUTER-AIDED DESIGN OF INTEGRATED CIRCUITS AND SYSTEMS. He was a recipient of the 2006 Sheffield Distinguished Teaching Award, best paper awards from the 2013 IEEE/ACM Design Automation and Test in Europe (DATE 2013) Conference and the 2015 IEEE VLSI Test Symposium (VTS 2015), and Best Hardware Demonstration Awards from the 2016 and the 2018 IEEE Hardware-Oriented Security and Trust Symposia (HOST 2016 and HOST 2018).

MURAT TORLAK (Senior Member, IEEE) received the M.S. and Ph.D. degrees in electrical engineering from The University of Texas at Austin, in 1995 and 1999, respectively. Since August 1999, he has been with the Department of Electrical and Computer Engineering, The University of Texas, where he has been promoted to the rank of a Full Professor. He is currently the Rotating Program Director at the U.S. National Science Foundation (NSF). His current research interests include experimental verification of wireless networking systems, cognitive radios, millimeter-wave automotive radars, millimeter-wave imaging systems, and interference mitigation in radio telescopes. He was the General Chair of the Symposium on Millimeter Wave Imaging and Communications, in 2013, and the IEEE GlobalSIP Conference. He has served as an Associate Editor for the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, from 2008 to 2013. He is a Guest Co-Editor of the IEEE JOURNAL OF SELECTED TOPICS IN SIGNAL PROCESSING—Special Issue on Recent Advances in Automotive Radar Signal Processing.