MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare

SIZHE AN and UMIT Y. OGRAS, University of Wisconsin-Madison, USA

Rehabilitation is a crucial process for patients suffering from motor disorders. The current practice is performing rehabilitation exercises under clinical expert supervision. New approaches are needed to allow patients to perform prescribed exercises at their homes and alleviate commuting requirements, expert shortages, and healthcare costs. Human joint estimation is a substantial component of these programs since it offers valuable visualization and feedback based on body movements. Camera-based systems have been popular for capturing joint motion. However, they have high-cost, raise serious privacy concerns, and require strict lighting and placement settings. We propose a millimeter-wave (mmWave)-based assistive rehabilitation system (MARS) for motor disorders to address these challenges. MARS provides a low-cost solution with a competitive object localization and detection accuracy. It first maps the 5D time-series point cloud from mmWave to a lower dimension. Then, it uses a convolution neural network (CNN) to estimate the accurate location of human joints. MARS can reconstruct 19 human joints and their skeleton from the point cloud generated by mmWave radar. We evaluate MARS using ten specific rehabilitation movements performed by four human subjects involving all body parts and obtain an average mean absolute error of 5.87 cm for all joint positions. To the best of our knowledge, this is the first rehabilitation movements dataset using mmWave point cloud. MARS is evaluated on the Nvidia Jetson Xavier-NX board. Model inference takes only 64 μs and consumes 442 μJ energy. These results demonstrate the practicality of MARS on low-power edge devices.

CCS Concepts: • **Networks → Cyber-physical networks**; • **Human-centered computing → Ubiquitous and mobile computing**; • **Computing methodologies → Reconstruction;**

Additional Key Words and Phrases: Human pose estimation, point cloud, millimeter wave, smart healthcare

ACM Reference format:
Sizhe An and Umit Y. Ogras. 2021. MARS: mmWave-based Assistive Rehabilitation System for Smart Healthcare. *ACM Trans. Embedd. Comput. Syst.* 20, 5s, Article 72 (September 2021), 22 pages.
https://doi.org/10.1145/3477003

1 INTRODUCTION

Rehabilitation is the process of recovering a patient's health condition to its normal state after a period of illness. The sequelae of central nervous system disorders, such as Parkinson's disease (PD) and cerebrovascular diseases (e.g., stroke), afflict more than 10 million people worldwide. According to recent studies, patients can recover up to 91% functional ability if they start the rehabilitation

This article appears as part of the ESWEEK-TECS special issue and was presented in the International Conference on Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2021.
This research was funded in part by NSF CAREER award CNS-2114499.
Authors' address: S. An, and U. Y. Ogras, University of Wisconsin-Madison, Department of Electrical and Computer Engineering, Madison, WI, 53706, USA; emails: {sizhe.an, uogras}@wisc.edu.

1539-9087/2021/09-ART72 $15.00
https://doi.org/10.1145/3477003

within three months of the stroke [22]. Similarly, PD patients must follow regular rehabilitation treatments to maximize their functional ability and minimize secondary complications [3]. There is also a strong interplay between mental well-being and maintaining physical activity [14]. These examples demonstrate the importance of rehabilitation treatment to regain patients' quality of life.

The current mainstream rehabilitation treatment involves a physical therapist who supervises the patients in person. The supervision aims to guide the patients to perform specific movement exercises and give feedback to ensure their correctness. Individualized attention from an expert is certainly favorable, but it also incurs a high cost due to critical dependence on experts, dedicated infrastructure, and patients' commute. *Indeed, the Covid-19 pandemic experience has further demonstrated the growing importance of developing alternatives to in-person care.* Home-based rehabilitation systems are needed to address this urgent need. These systems must allow patients to perform prescribed movement exercises at home while receiving feedback. In this way, they can complement the therapists by enabling daily practices between clinic visits, which can be weeks apart. During these practices, they can monitor whether the patients perform the movements correctly and adjust the intensity. For example, when a patient lifts the arm or the thigh, the system must tell if it is high enough and guide patients to improve the movements.

Home-based patient monitoring systems use predominantly two approaches: wearable sensors- and video-based systems [6, 30, 46]. One of the main advantages of wearable systems is their independence from environmental factors. For instance, it does not matter whether the patients perform the rehabilitation exercise indoors or outdoors as long as they wear the sensors. Also, sensor measurements can be very accurate when appropriately placed. However, recent studies show that frequent charging requirements and discomfort hinder the users from using the wearables [10, 35]. Moreover, multiple sensors are required to capture full-body motion, e.g., account for wrist, arm, and legs simultaneously. The video-based systems tackle some of the drawbacks of wearable systems. They usually use an RGB video camera, depth camera, or other motion capture systems, such as Microsoft Kinect [27] and Intel RealSense sensors [20]. The user only needs to perform some actions in front of a camera instead of wearing multiple sensors. Besides, video-based methods can reconstruct multiple essential body parts, such as the neck, wrist, shoulder, and ankle [12]. Then, they can model real-time skeleton movement by connecting these points [12, 34]. However, video-based systems also face critical challenges that limit their practicality. First, they require a strict environment setting, such as lighting and camera placement, which significantly affects accuracy. Second, privacy is a more complex issue since many users do not want to share their camera access and videos. Consequently, the challenges discussed in this paragraph limit the use of video-based rehabilitation assistive systems at home.

With the recent advances in mmWave technology, Radio Frequency (RF) imaging has emerged as a promising technique that can address the limitations of wearable and video-based rehabilitation systems [48]. Small form-factor and low-power mmWave radars have become commercially available [41]. These devices provide a high-resolution 3D point cloud representation, which can be processed locally using edge artificial intelligence (AI) algorithms to reconstruct human motion. Since they generate and transmit RF signals towards the target, they can maintain a robust operation under poor lighting and weather conditions. They also address privacy concerns since mmWave radar signals do not involve any video images or facial information. *However, existing mmWave radar techniques have primarily been limited to object detection and target localization since it represents the scene with a reflection point cloud instead of the true color image* [24, 26].

This paper proposes a novel real-time mmWave-based Assistive Rehabilitation System (MARS) to monitor patient movements in home environments accurately and provide real-time feedback. MARS tracks the patient movement using a low-cost mmWave radar [41]. Unlike prior work that uses the point cloud for activity recognition and localization [24–26, 36], our novel pre-processing

algorithms and convolutional neural network (CNN) design convert the radar point cloud to 3D joint coordinates. *This unique capability enables MARS to produce real-time skeleton movements without using any video images or facial information.* Hence, its output is compatible with more expensive and complex video-based systems. Furthermore, MARS supports angle and speed estimations of limbs as well as posture correction. We evaluated MARS empirically by performing experiments with *70 minutes of exercise data (40,083 frames) from ten popular rehabilitation movements*. We also collected reference data using Microsoft Kinect V2 sensor [27] during these experiments. Experimental results show that MARS accurately estimates the 3D coordinates of 19 joints with only 5.87 cm mean absolute error (MAE). For more details about comparative results, please refer to Section 5.2. MARS also offers joint angle and velocity estimation as the feedback of the rehabilitation movements. The average MAE of MARS in the knee and the elbow angle estimation is 6° and 12°, respectively. Finally, we implement the proposed approach on the Nvidia Jetson Xavier-NX board [29]. Our experiments show that MARS can process well over 9,000 frames per second with less than 500 μJ energy consumption per frame. Hence, it can be used reliably for home-based rehabilitation systems.

In summary, the major contributions of this paper are as follows:

- A low-cost and low-power mmWave-based assistive rehabilitation system that accurately reconstructs 19 human joints and skeleton movements using mmWave point cloud data,
- A novel pre-processing method that transforms the raw 5D time-series point cloud with irregular length and random order to a 3D 5-channel stacked feature map; a CNN for processing the proposed feature map to 3D spatial coordinates of human joints,
- A *first-of-its-kind* rehabilitation movement dataset using mmWave point cloud, including 70 minutes of ten distinct rehabilitation movements performed by four human subjects with 19 human joints data and 40,083 labeled frames and their video demonstrations to the public [37],
- Experimental evaluations show, on average, about 5 cm localization error in 3D space, and 6° error for the knee angle, and 12° error for the elbow angle.

2 RELATED WORK

This section discusses the related work systematically under assistive healthcare systems, human pose tracking, and mmWave imaging categories.

2.1 Healthcare Assistive System using Internet of Things (IoT)

Successful rehabilitation is a crucial step for the patients to improve their self-care ability, re-participate in social life, and raise their quality of life. The IoT technology has become a promising solution to offer long-term, holistic, and accurate health monitoring owing to its rapid development. This technology can alleviate the caregiver burden and provide valuable information to the clinical experts to support their decision-making for rehabilitation suggestions.

Wearable devices dominate earlier research in healthcare assistive systems due to their increasing affordability and popularity. Weiss et al. [47] analyze PD patients' movements using a triaxial accelerometer and a triaxial gyroscope. Specifically, the transition between turning and sitting in patients with Parkinson's disease is the focus of this study. Abbate et al. [2] aim for long-term monitoring and fall detection in nursing homes by using accelerometers and electroencephalograph (EEG). The paper also involves ergonomics for designing the health monitoring system. A garment-based system using a strain-sensor to facilitate rehabilitation was presented in 2009 [16, 17]. This system provides the patients with real-time feedback based on wearable sensors embedded in the garment. In [39], Bhomer et al. developed a sound assistive rehabilitation

system. The sound reflects the movement of the subject as the pitch or volume of a tune, manipulated by a stretch fabric sensor. However, these two systems focus only on the upper limb and trunk. Besides, they do not provide any joint and skeleton information of a body and lack quantitative results. In summary, wearable-based assistive healthcare systems usually consist of multiple sensors. They mainly target one part of the body since broader coverage requires more sensors. Lack of interpretation ability is also a significant shortcoming of these systems since they rarely give human joint or skeleton information.

Another mainstream technique used in healthcare assistive system research is computer vision (CV). CV offers an accurate representation of the real world using true-color images or videos. The primary goal of using vision-based approaches in rehabilitation systems is to help patients improve motor functions and overcome challenges by monitoring their daily actions and activities instructed by the doctors. RGB video camera [8, 9], depth camera [7, 23, 45], and motion capture system [9, 45] are the major devices have been used in this area. In [8], Ar et al. present Home-based Physical Therapy Exercises (HPTE) dataset which targets therapy actions. The Kinect camera is used in this study to provide video and depth streams to the user. One of the outputs of this approach is the binary image that indicates the body shape. They give eight shoulder and exercise movements but without any joint or skeleton information. In 2015, researchers developed a system and released a dataset named EmoPain that has both body joint information and face videos [9]. The object of EmoPain is to relate the pain to the emotion in the rehabilitation systems. These systems use RGB cameras often has limitation due to environmental noise, and lens distortion [34]. A dataset named AHA-3D was released in 2018 [7]. This dataset contains 79 skeleton videos, each consisting of one exercise repeated 1-3 runs. It is recorded with both young and elderly subjects using a Kinect v2 sensor. Similarly, a more recent study [45] in 2018 presents a dataset called UI-PRMD with 10 subjects doing the common physical rehabilitation exercises using the Vicon and Kinect systems. It has advantages over others because of its performance metrics for the rehabilitation exercises instructed by clinical experts.

In the computer vision area, researchers have focused on human pose estimation since 2005 [31]. This study proposes a framework that can detect ten distinct body parts using rectangular templates from RGB images. In [19], He et al. present Mask R-CNN, which can reconstruct skeleton from RGB images using K masks by leveraging ResNet architecture. It first detects K different key points then connects them. Mask R-CNN has become popular due to its fast processing time and accurate estimation. At almost the same time, Cao et al. proposed OpenPose [12], a real-time human pose estimation techniques that can detect human body, face, and foot key points together for the first time. OpenPose also has become one of the popular benchmarks due to its decent performance and the easy-to-use open-source package. Besides the RGB video-based approach, Microsoft Kinect and Kinect V2 [34] provide depth cameras to extract the human joints information. Kinect uses an RGB and infra-red camera, while Kinect V2 uses a Time of Flight (ToF) camera to capture the information. The Kinect family has become one of the popular ways to obtain the ground truth label for training due to its convenience, low cost, and accurate performance [7, 33, 49].

2.2 mmWave Radar Imaging

Localization [24] and multiple kinds of classification tasks [25, 26, 36] are the fundamental applications of mmWave radar. The work in [24] assumes an environment with many static mmWave devices referred to as anchor points placed in known locations (four corners of a room) and a mobile node surrounded by these anchor points. Using the flight time and arrival angles derived by the transmitted and received signals, the system can determine the mobile node's localization. Relying on multiple devices degrades the practicality of this approach. Singh et al. [36] present a human activity recognition approach using a 77 GHz TI IWR1443 mmWave radar. They achieve

above 90% accuracy with deep learning classifiers recognize five different activities: boxing, jumping, jumping jacks, squats, and walking. Similarly, Liu et al. [25], and Meng et al. [26] perform gesture recognition and gait recognition using the point cloud data generated by IWR1443. However, these tasks only focus on classification tasks with up to ten classes.

Human skeleton reconstruction, a more challenging task, is first considered in RF-Capture[5]. RF-Capture first outputs the coarse human body parts using FMCW signals from 5.4 to 7.2 GHz. It paves the way for further study using mmWave, but it does not provide accurate human joint points all over the body. In 2018, researchers from the same group proposed RF-Pose3D [49], a technique that reconstructs up to 14 body parts, including head, neck, shoulders, elbows, wrists, hip, knees, and feet. This work first uses 12 camera nodes to record RGB-based video then obtain label key points from OpenPose. At the same time, FMCW signals at a few GHz are used to generate the RF heatmap. They then train a region proposal network (RPN) zooms in on RF data and a CNN with ResNet architecture to extract the 3D skeleton from the region of interest. For key point localization, the average errors in x, y, z axes are 4.2, 4.0, and 4.9cm, respectively. Besides being limited to 14 joints, this work does not leverage the mmWave radar's ability to obtaining a high-quality point cloud. Thus, it requires a much more complex NN architecture with high computation cost. Moreover, multiple cameras and bulky FMCW signal generating systems hinder the practicality of the approach. Most recently, Sengupta et al. propose mmPose [33], an approach that predicts over 15 joints. mmPose constructs the skeleton from point clouds by using two IWR1443 radar devices and a forked-CNN architecture. It reports 3.2, 2.7, and 7.5 cm localization errors in x, y, z axes, respectively. Besides requiring two radars, mmPose uses a large CNN model (with twice as many parameters as MARS) and incurs a high computation cost since it first projects the point cloud data into two different planes instead of using the 3D representation.

In contrast to the previous studies, we propose MARS: a low-cost, low-power mmWave based *assistive rehabilitation system* for motor disorders. We leverage high-quality point clouds generated *by only one* TI IWR1443 76-81 GHz radar device to reconstruct 19 critical human joints. Our novel pre-processing and feature map generation algorithms *enable robust estimation with low computational overhead. More importantly, MARS targets ten complex rehabilitation movements instead of relying on simple activities, such as walking. We will release the first labeled dataset that includes both mmWave and Kinect V2 data and video demonstrations for these movements [37].*

3 BACKGROUND ON MMWAVE RADAR AND OVERVIEW

Frequency Modulated Continuous Wave (FMCW) mmWave radar has recently attracted significant attention, especially in automotive and industrial applications. The fundamental component of FMCW is a chirp signal, which is a sinusoid wave whose frequency increases linearly with time [32, 42, 44]. Due to this characterization, a chirp signal is typically displayed by a linear frequency versus time plot, as illustrated in the top left part of Figure 1. The chirp signal is uniquely defined by its start frequency (f_c), duration (T_c), and bandwidth (B). The bandwidth to duration ratio gives the chirp slope (S), i.e., the rate at which the signal frequency increases ($S = B/T_c$).

The FMCW radar synthesizes a sequence of chirp signals to form a frame. For instance, Figure 1 illustrates a frame with N back-to-back chirp signals. It transmits the chirp frame using a transmitter (TX) antenna. If any object is in the vicinity, it reflects off the chirp frame. Then, the FMCW radar receives the reflected signals at the receiver (RX) antennas. Note that both TX- and RX-signals are chirps with different instantaneous frequencies and phases. A mixer module in the radar processes these signals to produce an *intermediate frequency* (*IF*) signal [44], which is another sinusoid with the following instantaneous frequency (f_{IF}) and phase (ϕ_{IF}):

$$f_{IF} = f_{TX} - f_{RX}, \qquad \phi_{IF} = \phi_{TX} - \phi_{RX} \tag{1}$$

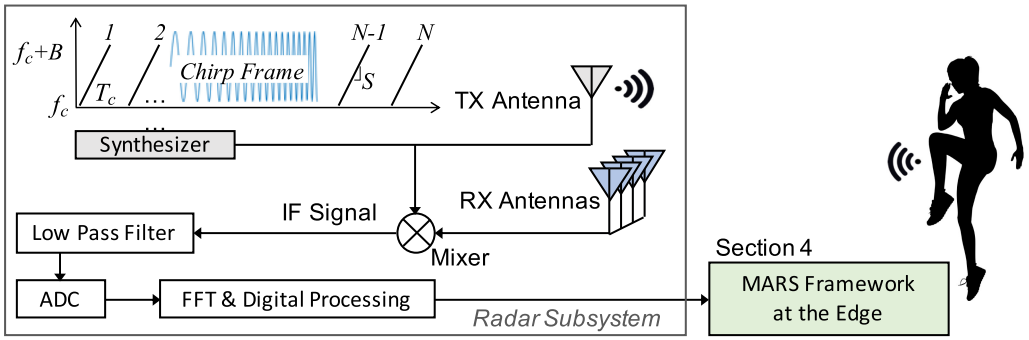

Fig. 1. Overview of the mmWave imaging system. The radar transmits chirp frames. Then, it mixes the reflected and transmitted frames to produce 5D point cloud data, which is then processed by MARS.

Suppose only one object reflects the chirp frame with distance d from the radar. The round-trip delay of the received signal can be found as $\tau = 2d/c$, where c is the speed of light. Since the received signal is a replica of the transmitted frames delayed by τ, the frequency of the IF signal will be $St - S(t - \tau) = S\tau$. That is, the IF signal has a single tone when only one object reflects the chirp signal. We can find the frequency of this tone as $S\tau = 2dS/c$. When the chirp frame is reflected from multiple objects or different parts of the body, the mixer will produce an IF signal with multiple tones, a.k.a., beat frequencies. FMCW radar chips extract the IF signal tones by computing frequency spectrum using the fast Fourier transform (FFT), as depicted in Figure 1. As in the single object case, the frequency of each tone is proportional to the distance of the corresponding object. The IF signal is processed in the digital domain to map the tones into range bins using a *range FFT* process [32]. Note that the range resolution is inversely proportional to the chirp bandwidth:

$$d_{res} = \frac{c}{2B} \tag{2}$$

where c is the speed of light and B is the chirp bandwidth.

Another essential metric besides the range is the velocity of the detected objects. FMCW radars compute the velocity using the phase changes in the IF signal across multiple chirps. This process converts small displacements of the object to a phase difference in the IF signal. As in the range detection case, there may be multiple objects with equal distance from the radar but with different relative velocities. The chirps in the transmitted and received frames (N chirps in Figure 1) are processed by a second FFT, called *Doppler-FFT* [32], to resolve the velocity of different objects. After this step, the radar can produce a range-Doppler heat map to detect object velocities. The velocity resolution of the radar is inversely proportional to the frame time as:

$$v_{res} = \frac{\lambda}{2NT_c} \tag{3}$$

where λ is the wavelength, N is the number of chirps, and T_c is the time between two chirps. For instance, v_{max} is 39 m/s given T_c is 25 μs, which is significantly faster than human motion. Finally, the FMCW radar filters out the noise interference using a *noise elimination* algorithm, such as the built-in constant false alarm rate (CFAR) [28], used in this work.

The last metric estimated by the radar is the angle of arrival (AoA). AoA is defined as the angle of a reflected signal with the horizontal plane [11]. Angle estimation requires at least two RX antennas and it is calculated by $\theta_{res} = \lambda / N_{RX} N_{TX} d cos(\theta)$, where λ is the wavelength, N_{RX} is the number of receiver antennas, N_{TX} is the number of transmitter antennas, d is the distance

Table 1. List of Major Parameters and Variables Related to mmWave and their Values in this Work

Symbol	Description	Values	Symbol	Description	Values
f_c	Starting frequency	77 GHz	θ_{res}	Angle resolution	9.55°
T_c	Chirp duration	32 μs	N_{RX}	No. of RX antennas	4
B	Bandwidth	3.20 GHz	N_P	Maximum points detectable per frame	64
S	Slope of chirp	100 MHz/μs	f_{IF}	Frequency of IF signal	NA
N	No. of chirps per frame	96	ϕ_{IF}	Phase of IF signal	NA
d_{res}	Range resolution	4.69 cm	τ	RX signals time difference	NA
v_{max}	Maximum Velocity	5.69 m/s	D_i	the i^{th} point's Doppler velocity	NA
v_{res}	Velocity resolution	0.35 m/s	I_i	the i^{th} point's reflection intensity	NA
x_i, y_i, z_i	3D coordinates of the i^{th} point	NA	p_i	Point representation of the i^{th} point	NA
N_{TX}	No. of TX antennas	3			

"NA" means that the corresponding is not a fixed parameter but a variable for each point.

between two consecutive receiver antenna, and $cos(\theta)$ is the cosine of the angle between two receivers. Note that the resolution is often quoted assuming that $d = \frac{\lambda}{2}$ and $\theta = 0$, such that $\theta_{res} = \frac{2}{N_{RX}N_{TX}}$ (radians) [15]. The radar chip estimates the angle of arrival by using the phase change in the 2D-FFT peak caused by the different distances from the object to each antenna. This FFT is referred to as the *angle FFT*, which outputs the azimuth angle divided by elevation angle.

Finally, the radar receives multiple RX signals back for all the chirps the TX antennas sends. Each object (or body part) that reflects an RX signal is referred to as a point. For each point p_i in the frame, the radar chip calculates its 3D coordinates by using the result of range FFT after the noise elimination algorithm. Moreover, it computes the Doppler velocity and reflection intensity. Multiple points within each frame form the *point cloud*, which is formatted as follows:

$$p_i = \{x_i, y_i, z_i, D_i, I_i\}, i \in [0, N_P] \tag{4}$$

where x_i, y_i, z_i represents the spatial coordinates of the point, D_i denotes the Doppler velocity, I_i denotes the signal intensity, and N_P denotes the total number of points in this frame. Note that N_P will be zero, i.e., there will not be any points when no object is detected. On the contrary, the number of detected points can exceed the radar chip's capacity if too many signals are reflected. Therefore, radar chips limit the maximum value N_P can take.

The radar parameters used in this study are shown in Table 1. In our work, the first 64 reflected points are processed, i.e., $N_P = 64$. The IWR1443 radar sensor is configured with three TX antennas and four RX antennas. The frame duration and sampling rate are set to 100 ms and 2.49 Msps, respectively. With these parameters, the radar has a maximum detection range of 3.37 m, maximum detection velocity of 5.69 m/s, a range resolution of 4.69 cm, and velocity resolution of 0.35 m/s. For more mmWave radar details, we refer the readers to recent tutorials [15, 32, 42].

4 MARS: MMWAVE-BASED ASSISTIVE REHABILITATION SYSTEM

This section presents the proposed MARS framework that processes the raw mmWave point cloud data to provide rehabilitation feedback to the user. To provide accurate and relevant feedback, MARS tracks the following fundamental attributes in real-time: The 3D position (x, y, z coordinates) and velocity (along x, y, z dimensions) of 19 joints, four key angles, as listed in Table 2. Furthermore, it provides correction feedback on ten commonly used postures shown in the last row of Table 2. MARS accomplishes these tasks by following the following steps outlined in Figure 2:

(1) Use an FMCW radar to collect point cloud data, as described in Section 3,
(2) Pre-process the point cloud to construct robust and delay-invariant features (Section 4.1),
(3) Infer 3D joint positions using the new features and a CNN architecture (Section 4.2),

Table 2. MARS Provides 3D Joint Positions and the Velocity of 19 Joints listed in the first Row

3D joint position estimation	SpineBase, SpineMid, Neck, Head, SpineShoulder, ShoulderLeft, ElbowLeft, WristLeft, ShoulderRight, ElbowRight, WristRight, HipLeft, KneeLeft, AnkleLeft, FootLeft, HipRight, KneeRight, AnkleRight, FootRight
3D joint velocity estimation	
Angle estimation	Left elbow, Right elbow, Left knee, Right knee
Posture correction feedback	Left upper limb extension, Right upper limb extension, Both upper limbs extension, Left front lunge, Right front lunge, Squat, Left side lunge, Right side lunge, Left limb extension, Right limb extension.

It also estimates the angles in the second row and provides posture correction feedback for ten movements in the last row.

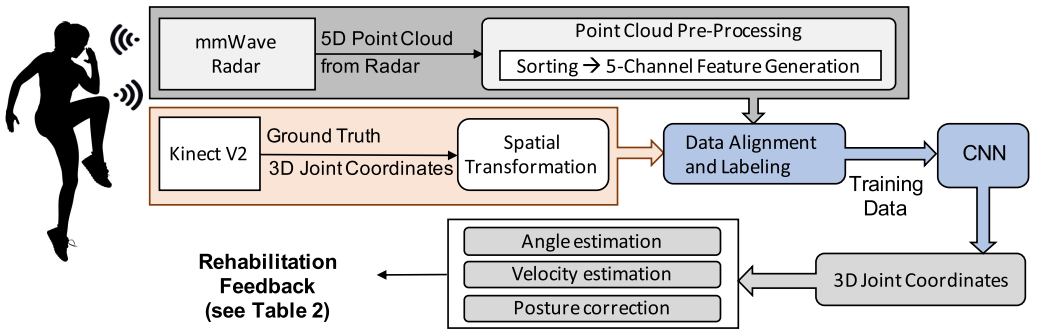

Fig. 2. Overview of proposed MARS framework and its interaction with radar and Kinect V2.

(4) Produce user feedback by converting the 3D joint positions to joint velocity and angle estimations (Section 4.3).

4.1 Point Cloud Pre-Processing

4.1.1 Input Data: Challenges and Reformatting. The primary input to MARS is a point cloud arranged as five-dimensional (5D) time-series data. Each point consists of the x, y, z coordinates of the points that reflected the TX-signal (x, y, z), Doppler velocity D, and the reflection intensity I, as described in Section 3. The FMCW radar stores the first N_P points to form a data frame (in this work $N_P = 64$, as shown in Table 1). If fewer than N_P body parts reflect the chirp signals, fewer than 64 points will be received. In these cases, the rest of the frame is padded with zeros to obtain a uniform size ($N_P \times 5$) input frames.

Figure 3 depicts a sample input frame from different perspectives. The triangle marker represents the radar location, which is also set at the origin $(0, 0, 0)$. Figure 3(a) shows that point positions in 3D, while the other plots show their projections to 2D coordinates. Similarly, Figure 3(a) illustrates the Doppler velocity, which indicates the relative velocity from the detected point to the radar. Finally, the colors in the figures represent the energy intensity of the reflected signals.

4.1.2 Feature Generation for CNN. Note that the reflected chirp signals arrive at the radar in random order due to slight variations in the body posture and round-trip delay, as illustrated in

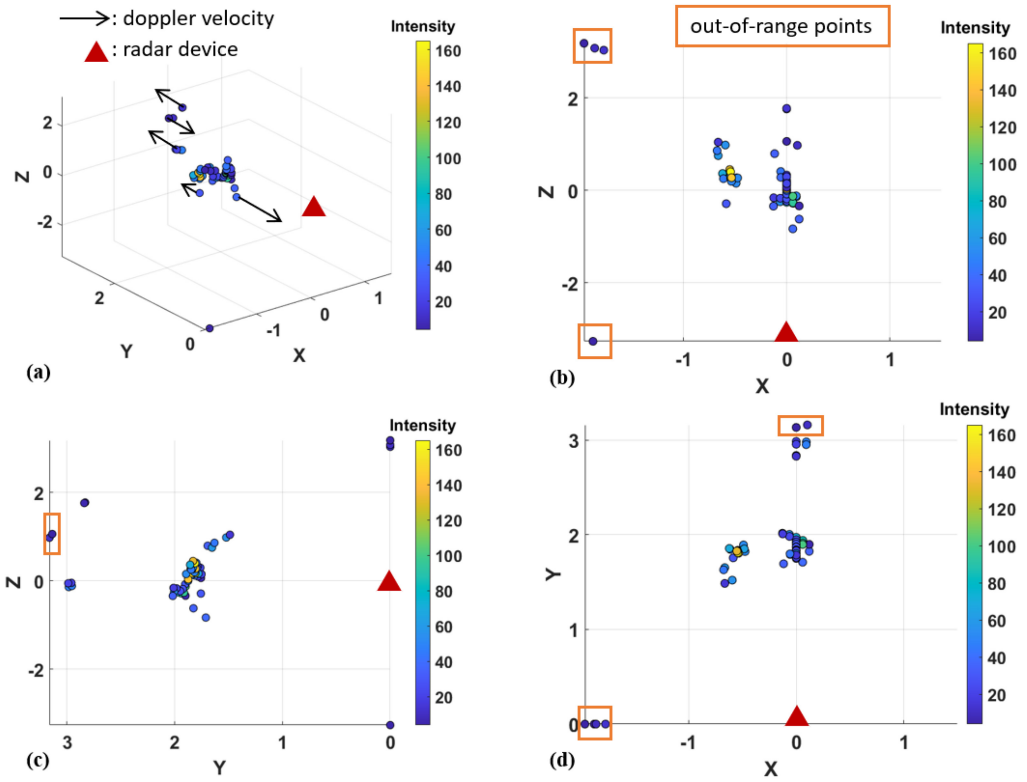

Fig. 3. mmWave point cloud representation for one frame. (a), (b), (c), and (d) shows the 3D view, front view, side view, and top view, respectively.

Figure 4(a). Therefore, the order of the points in a frame is random, i.e., the same point may be in different positions across consecutive frames. Although CNNs have a decent shift, scaling, and rotation invariance, random data ordering poses a challenge to CNN design. Furthermore, the input data must have a fixed shape. To address these issues, we propose a pre-processing algorithm that consists of sorting and matrix transformation. We sort the points in each frame in ascending order of x, y, and z coordinates. The points are sorted first based on their x coordinates in ascending order. At the second level, the points with the same x coordinate are sorted by their y coordinates. Finally, for the data points with the same x and y coordinates, we sort them by z coordinates in ascending order, as illustrated in Figure 4(b). Note that this sorting does not change the distances between the points since we only change the order of the inputs to the CNN.

After the sorting phase, the dimension of the input features is 64×5, i.e., the input data is arranged as a column vector with 64 rows each with 5 features. The transformation converts 64 rows into an 8x8 square matrix in the row-major order, similar to images commonly used in CNNs. Thus, the transformation reshapes the same data while preserving the values from the 64×5 matrix to an $8 \times 8 \times 5$ data structure. Since there are five dimensions (x, y, z coordinates, Doppler velocity, reflection intensity), we end up with five channels, each with an 8×8 feature map.

4.1.3 Handling Out-of-range "ghost images". mmWave radar imaging can sometimes generate a point called "ghost image" that is outside the range of interest [4]. In assistive rehabilitation systems, the user is standing within a fixed distance away from the radar sensor. Hence, we divide

(a)

# Obj	X	Y	Z	Doppler	Intensity
11	-0.06	1.81	0.35	0.36	13
11	-0.12	1.86	0.22	0.36	15
11	-0.12	1.86	0.18	0.36	14
11	-0.18	1.91	0.22	0.36	10
11	0.61	1.88	0.89	0.36	10
11	0.00	1.80	0.21	-0.36	9
11	0.00	1.84	0.17	-0.36	13
11	-0.12	1.87	0.08	-0.36	16
11	0.00	1.80	0.04	-0.36	14
11	-0.12	1.93	0.05	-0.36	11

(b)

# Obj	X	Y	Z	Doppler	Intensity
11	-0.18	1.91	0.22	0.36	10
11	-0.12	1.86	0.18	0.36	14
11	-0.12	1.86	0.22	0.36	15
11	-0.12	1.87	0.08	-0.36	16
11	-0.12	1.93	0.05	-0.36	11
11	-0.06	1.81	0.35	0.36	13
11	0.00	1.80	0.04	-0.36	14
11	0.00	1.80	0.21	-0.36	9
11	0.00	1.84	0.17	-0.36	13
11	0.61	1.88	0.89	0.36	10

Fig. 4. Input data (a) before and (b) after sorting.

the generated point cloud into two classes: *in-range* and *out-of-range*. The in-range point cloud is defined by lower and upper bounds on each dimension. In our implementation, we use the following ranges: $x \in \{-1\ m, 1\ m\}$ (horizontal width), $y \in \{0, 3\ m\}$ (depth), and $z \in \{-1\ m, 1\ m\}$ (vertical height). The points within these boundaries are considered in-range, while others are marked as out-of-range points. The out-of-range points are highlighted by rectangles in Figure 3 for illustration. The out-of-range points (i.e., ghost images) are inevitable in real application scenarios due to scattering. Therefore, MARS marks and includes out-of-range points in training and inference. Section 5.3.2 presents quantitative results and discusses the implications of this choice.

4.2 CNN Architecture Design

The next step is converting the feature maps depicted in Figure 3 into actual 3D joint positions. This challenging task is accomplished using a CNN architecture that outputs the x, y, and z coordinates of 19 joints, as illustrated in Figure 5. The input layer of the CNN takes the stacked 5-channel feature map as the input. Two consecutive convolution layers follow the input layer with 16 and 32 channels, respectively. After performing the convolutions, the data is passed to a flattening layer that generates the input vector for the fully connected (FC) layers. The first FC layer is with 512 neurons. The final output of CNN contains 57 neurons, which stand for 3D coordinates for the 19 joints. All activation functions are Relu except for the final FC layer, where we use linear activation. Finally, four dropout layers with probabilities of 0.3 and 0.4 are employed after the convolution layers and the fully connected layers to avoid excessive dependency on specific neurons.

The design choice of including Batch Normalization (BN) and max-pooling or leaving them out is vital in developing a CNN. The BN layer is commonly used to avoid significant data distribution changes after each mini-batch called "internal covariate shift." The max-pooling layer is used to maintain the feature invariance after image translation, rotation, and scaling by taking the maximum of a particular region. It also reduces model parameters while avoiding overfitting and improving the generalization ability of the model. MARS implements BN layers after the second convolution layer and the fully connected layers. We choose not to have a max-pooling layer since it loses local information, which is essential for the spatial coordinates regression task. We discuss these choices and present a comprehensive quantitative evaluation in Section 5.3.3.

4.2.1 Ground Truth and Loss Function. Training the proposed CNN architecture requires the ground truth, i.e., the reference positions of the target joint positions. In this work, we use a Kinect V2 sensor [27] to capture the reference coordinates. The Kinect sensor and the mmWave radar are placed on the same table, next to each other. This placement does not lead to spatial offsets in the

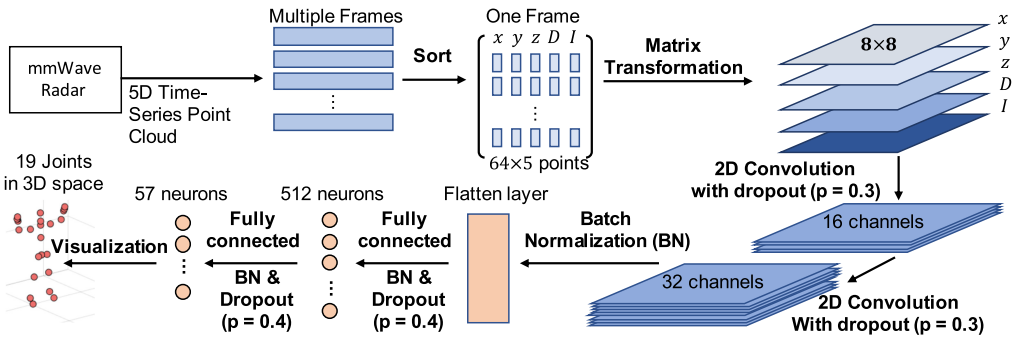

Fig. 5. Point cloud pre-processing and CNN architecture.

y and z-axis between two sensors since their y and z coordinates are identical. However, there is a spatial offset. Firstly, the x-axis in the Kinect sensor's reference frame is inverted with respect to the mmWave radars x-axis. Thus, we take the additive inverse of all x-axis values during the pre-processing stage. Secondly, there is still a 3 cm offset in the x-axis since the sensors are placed next to each other. We do not manually calibrate this offset since the CNNs have decent shift-invariance. CNN itself can learn the spatial offset between the mmWave sensor and the Kinect sensor. The Kinect sensor's sampling rate is fixed at 30 Hz, while the radar's frame duration is 100 ms. Hence, we align the radar and the Kinect sensor data frame by frame. The frame alignment is achieved by connecting both devices to the same laptop and timestamping the data frames from each device. We find the closest timestamp in the Kinect sensor for each radar data frame and pair it with the radar data as its label.[1]

For a given data frame, let x_i, y_i, and z_i be the *reference coordinates of joint* i, $1 \leq i \leq N_J$ from the Kinect sensor, where N_J is the number of tracked joints. Similarly, let the corresponding estimates from MARS be $\hat{x}_i$, $\hat{y}_i$, and $\hat{z}_i$, respectively. We define the loss function as the mean squared error (MSE) between the reference positions and the estimations as follows:

$$Loss_{coor} = \frac{\sum_{i=1}^{N_J}(x_i - \hat{x}_i)^2 + \sum_{i=1}^{N_J}(y_i - \hat{y}_i)^2 + \sum_{i=1}^{N_J}(z_i - \hat{z}_i)^2}{3N_J} \tag{5}$$

An illustration of MARS estimating 19 human joints from the mmWave radar is shown in Figure 6. From left to right, the subfigure represents the point cloud generated by radar, estimation from MARS, and the ground truth from the Kinect V2 sensor. We observe that MARS reconstructs 19 human joints accurately. We show only five of the ten rehabilitation movements due to space limitation. The remaining five movements look very similar since they are mirrored versions of the same movements. *A live demo can be found on our GitHub page, where the dataset is released* [37].

4.3 Rehabilitation Movement Feedback to User

4.3.1 Velocity Estimation. The CNN presented in Section 4.2 produces the 3D coordinates of 19 joints listed in Table 2. The next step is deriving the velocity of these joints. We find each joint's velocity by dividing its distance between two consecutive frames by the frame duration. One example is shown in Figure 7(d) for the squat movement. We first find the complete squatting frame (shown in solid line in Figure 7(d)). Then, the corresponding positions in the previous frame are found. Finally, the ratio of the distance between two consecutive frames and frame duration gives the joint velocities. The ground truth velocity is derived using consecutive ground truth 3D

[1]The time difference between a data-label pair is less than 5 ms by using the proposed time alignment method.

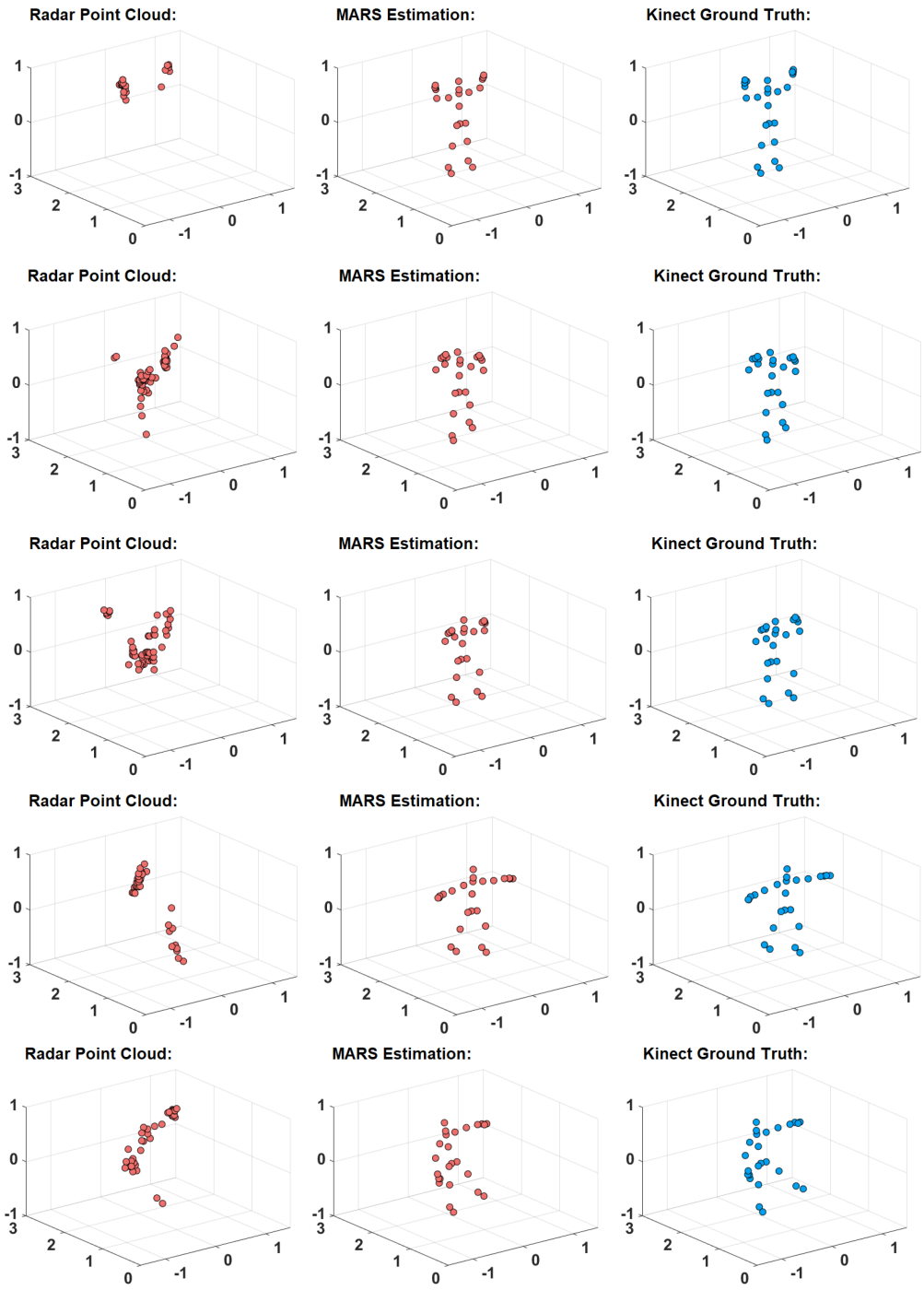

Fig. 6. Demo of MARS reconstructing human joints from point cloud for *both upper limb extension, right front lunge, squat, left side lunge, and left limb extension*. From left to right, it shows radar point cloud, MARS estimation, and ground truth, respectively. *The accuracy of the estimations are analyzed in Section 5.*

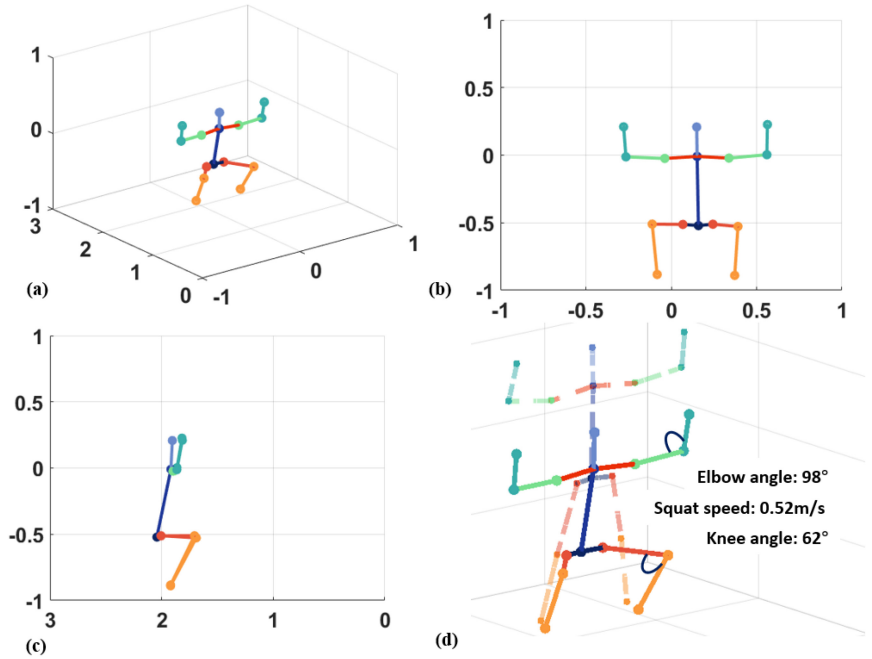

Fig. 7. Angle estimation by MARS during *squat* movements. (a), (b), (c), (d) shows the 3D view, front view, side view, and a zoomed in version with the estimated angles and speed.

coordinates reported by the Kinect sensor and their sampling times. For the squat example, the spinebase joint's velocity is used for evaluating the squat speed. In general, users can observe every joint's velocity when they perform different movements and adjust their pace accordingly.

4.3.2 Joint Angle Estimation. The joint coordinates found by the CNN are also used to find the angles between critical joints. This work focuses on the four most commonly used joint angles: right and left elbow angles, right and left knee angles, as listed in Table 2. The elbow angle is found using the shoulder, elbow, and wrist positions, as illustrated in Figure 7. We first calculate the skeleton length between the shoulder and elbow and the length between the elbow and wrist using their 3D coordinates. Then, the angle is obtained by using triangulation from the law of cosines. We follow the same procedures to calculate the knee angle using the hip, knee, and ankle positions. The ground truth angle is computed using the ground truth 3D coordinates reported by the Kinect sensor. As an example, Figure 7(d) illustrates a squat movement. We observe that the elbow and knee angles are 98° and 62°, respectively, for the complete squat.

4.3.3 Posture Correction. Therapists can define *specific rehabilitation movements* for a given user, such as the squat movement illustrated in Figure 7. Then, the correctness of a movement can easily be defined by setting acceptable ranges for relevant joints' velocities and angles. For example, the knee angles are essential for the squat movement. Thus, the user can set acceptable ranges for the knee angles, such as 55°–65° when the legs are stretched the most.

While the user performs the movements, MARS tracks the knee angles, as described in Section 4.3.2. Then, it compares the joint positions and angles to the acceptable ranges specified by the user (e.g., +/− 10% around ideal positions). The reconstructed skeleton is shown to the user with a transparent dash-line before the user's movement satisfies the acceptable range target, as

shown in Figure 7(d). The skeleton visualization becomes a solid line after the joint coordinates, angle, and velocities reach the set goals. The system can also support a sound played or a visual cue as feedback that indicates successful completion of the exercise.

5 EXPERIMENTAL EVALUATIONS

5.1 Experimental Setup and Dataset

mmWave radar: The radar processing is performed on Texas Instruments (TI) IWR1443 Boost mmWave radar [41]. We use a Matlab Runtime implementation from TI [43] for the data acquisition. The detailed configuration and radar parameters are summarized in Table 1. The device is connected to a laptop through the UART interface. It starts acquiring the data from the Matlab Runtime using a frame duration of 100 ms. Note that the frame duration can be set to different values for different applications. Due to the bandwidth limitation, the least frame duration we can set is 33.3 ms, equivalent to the 30 Hz sampling rate. We chose 100 ms (i.e., 10 Hz sampling rate) since it is enough for measuring human movement (the frequency of most voluntary human movements spans from 0.6 to 8 Hz [18]). The average power consumption of IWR1443 mmWave radar tested at power terminals is 2.1 W [40]

Kinect V2 sensor [27]: The ground truth reference is obtained using Microsoft Kinect V2. Both Kinect and radar are placed on a 1 m tall table while the subjects perform the instructed movements two meters away from the table. The Kinect V2 sensor is connected to a laptop through the USB port using an adaptor. It captures images with a 30 Hz sampling rate. Then, the images are processed using Matlab to identify the 3D coordinates of 19 human joints listed in Table 2. These positions are used as the labels during training and reference points for testing, as described in Section 4.2. The Kinect reference system requires a 12V 2.67A power adapter to work.

Hardware measurements: We implemented the proposed MARS framework, including all the pre-processing steps and the proposed CNN, on the Nvidia Jetson Xavier NX Development Kit [29]. The execution time and power measurements are presented in Section 5.5.

Open-source training and test datasets: We collected training and test data through user-subject studies, following an official protocol approved by our institution's IRB board. Each subject performed the ten movements listed in Table 2 (five of them are illustrated in Figure 6). This set of movements enables us to evaluate both the upper and lower body joints and associated angles. Each user performed each movement for two minutes, i.e., approximately 20 minutes of data is collected in total. As a result, we obtained close to 10,000 data frames per user. Each frame contains data for 19 joints. Furthermore, the Kinect V2 reference data points have three dimensions, while the data points from the radar have five dimensions (3D coordinates, Doppler velocity, and reflection intensity). Hence, our reference data set from Kinect and radar contain close to 570K ($10{,}000 \times 19 \times 3$) and 950K ($10{,}000 \times 19 \times 5$) points for each subject, respectively. We emphasize that this is a large-scale dataset w ith a comparable size to other similar studies. More importantly, it is *the first rehabilitation movement dataset using mmWave point cloud with well-labeled joints.* Since home-based assistive rehabilitation systems, like MARS, are user-specific, evaluations even on one user are representative. Regardless, we repeated the evaluations with four different users to obtain a total of 2.28 million reference data points from Kinect V2 and 3.81 million data points from mmWave data. We plan to release this dataset to the public through Github [37] together with the existing demo.

CNN training details: We implemented the proposed CNN using Tensorflow 2.2.0 [1] with Keras 2.3.4 [13]. We use the Adam [21] as the optimizer with an initial learning rate of 0.001. The CNN is trained with a batch size of 128 for 150 epochs, where the validation loss converges at 0.01. Aiming

Table 3. Average Localization Error for 19 Human Joints Position

	X (Horizontal) (cm)		Y (Depth) (cm)		Z (Vertical) (cm)		Average (cm)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
SpineBase	5.67	8.22	3.55	4.96	5.96	7.83	5.06	7.00
SpineMid	6.16	8.90	3.07	3.97	6.80	8.94	5.34	7.27
Neck	6.78	9.80	3.39	4.32	7.58	9.97	5.92	8.03
Head	7.37	10.57	3.69	4.69	8.20	10.68	6.42	8.65
ShoulderLeft	6.92	9.91	3.39	4.39	6.88	9.00	5.73	7.77
ElbowLeft	7.52	10.23	4.37	6.00	8.19	10.74	6.69	8.99
WristLeft	10.34	13.76	5.07	6.80	13.57	18.14	9.66	12.90
ShoulderRight	6.75	9.69	3.78	5.05	7.04	9.21	5.86	7.98
ElbowRight	7.96	10.71	4.73	6.74	8.41	10.93	7.03	9.46
WristRight	10.74	14.18	5.22	7.26	14.14	18.68	10.03	13.37
HipLeft	5.63	8.13	3.56	4.99	5.84	7.67	5.01	6.93
KneeLeft	5.56	8.10	4.09	5.63	3.25	4.53	4.30	6.09
AnkleLeft	6.27	8.83	4.29	6.18	2.47	4.49	4.34	6.50
FootLeft	6.59	9.36	4.84	7.01	3.04	5.14	4.82	7.17
HipRight	5.55	8.04	3.66	5.06	5.91	7.77	5.04	6.96
KneeRight	6.11	8.53	4.38	5.83	3.60	5.33	4.70	6.57
AnkleRight	6.92	9.42	4.43	6.10	2.74	5.37	4.69	6.96
FootRight	7.38	9.99	4.57	6.51	3.22	5.81	5.05	7.44
SpineShoulder	6.62	9.57	3.22	4.10	7.40	9.72	5.75	7.80
MARS 19 points Avg.	6.99	9.79	4.07	5.56	6.54	8.94	5.87	8.10

The results in this table are for 20% test data.

for a personalized model, we split the data time-wise for training, validation, and testing. First, each movement data is divided into 60% (24,066 frames)-20% (8,033 frames)-20% (7,984 frames). Then, we take the first 60% of it for training, the next 20% for validation, and the last 20% for testing. We choose to use the 60%-20%-20% ratio instead of the fixed-length since some data is not exactly two minutes. These frames add up to 2.28 million data points from Kinect V2 and 3.81 million data points from radar, as described under the dataset. The training is performed on AMD Ryzen^{TM} 7 3800X 8-Core 3.9 GHz and Nvidia RTX2080 with 8 GB of graphics memory.

5.2 Accuracy of 3D Joint Position Estimation

Table 3 shows the detailed localization error for the 19 human joint points, illustrated in Figure 6. We use the mean absolute error (MAE), and root mean squared error (RMSE) metrics to evaluate MARS. To eliminate the system errors, we train ten different models and take the average. This methodology is applied to all quantitative results reported in this paper. The average MAE for all 19 joints is 6.99, 4.07, 6.54 cm for x-, y-, and z-axes, respectively. Similarly, the average RMSE of x-, y-, and z-axes are 9.79, 5.56, and 8.94 cm, respectively. In general, the x- and z-axes have larger errors than the y-axis since our movements involve intensive horizontal and vertical displacement of all body parts. In contrast, the error along the y- axis is minimal (3.07 cm–5.22 cm) due to the smaller displacement in depth.

The mean average absolute error of most joints is smaller than 8 cm. The most notable exceptions are the right and left wrist joints. An intuitive explanation is that joints related to hands need a higher resolution to localize. Since the mmWave radar's range resolution is 4.69 cm@3.20 GHz as mentioned in Section 3, it is challenging for the model to reconstruct these points. Since estimating

Table 4. Comparison of Average Localization Error for 19 Human Joints Position Between Models Trained with Different Point Cloud Range

	X (Horizontal) (cm)		Y (Depth) (cm)		Z (Vertical) (cm)		Average (cm)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
With "in-range" point cloud	7.75	10.73	4.18	5.79	7.11	9.63	6.35	8.72
With all point cloud	6.99	9.79	4.07	5.56	6.54	8.94	5.87	8.10

The results in this table are for 20% test data.

human pose from mmWave point cloud is a relatively new research area, there are only a few studies to compare with [33]. MARS achieves 5% lower error with only half model parameters than the method proposed in [33], as explained in Section 5.3.2. By searching similar research areas, we note that the accuracy of MARS is competitive with the human pose estimation techniques [38]. Hence, MARS can provide reliable user feedback in home-based rehabilitation systems.

5.3 Ablation Study

We performed extensive ablation studies to demonstrate the necessity of each component in MARS and justify the design choice adopted by MARS.

5.3.1 Using Out-of-range Point Clouds During Training. As described in Section 4.1.3, some radar data frames may contain out-of-range points, also referred to as ghost images. It is possible to train MARS by *including* or *excluding* the out-of-range points, which constitute about 2% of all frames. To evaluate each choice's effectiveness, we first train the model with the frames that only contain the in-range point clouds (i.e., *out-of-range points are excluded*). Then, we obtain a different CNN model by using *all frames*. We observe that the model trained with all point clouds performs slightly better than the one trained with only in-range point clouds, as shown in Table 4. The out-of-range point clouds add noise to the inputs, making the CNN more robust. Furthermore, out-of-range points are inevitable during real use cases. Therefore, we conclude that they should be included in the training data.

5.3.2 Different Feature Channels and Projection for CNN. As discussed in Section 4.1.2, we have an 8×8 feature map for each channel, including x, y, z coordinates, Doppler velocity, and reflection intensity. We can combine and stack these feature maps in different ways to obtain a stacked feature map for the CNN. To find out the best option, we train different CNNs with four different stacked feature maps and refer to the CNNs as *Configuration-1*, *Configuration-2*, *Configuration-3*, and *Configuration-4*. *Configuration-1* represents the CNN trained with feature maps only stacked with x, y, z three channels. *Configuration-2* represents the CNN trained with feature maps stacked with x, y, z, and Doppler velocity, four channels. *Configuration-3* represents the CNN trained with feature maps stacked with x, y, z, and reflection intensity, four channels. Finally, *Configuration-4* represents the CNN trained with feature maps stacked with x, y, z, Doppler velocity, and reflection intensity, five channels.

We observe that the *Configuration-1* model has the worst performance due to a lack of Doppler velocity and reflection intensity information, as shown in Table 5. *Configuration-2* and *Configuration-3* has slightly better performance since Doppler velocity or intensity information is introduced. *Configuration-4* performs the best since the 5-channel feature maps contain all the information, including x, y, z with both Doppler and intensity. Note that because of weight sharing in CNN, adding channels in input only increases negligible parameters in the model, as shown in Table 5. We then apply *Configuration-4* in MARS.

Table 5. Comparison of Average Localization Error for 19 Human Joints Position Across Models Trained with Different Feature Channels

	X (Horizontal) (cm)		Y (Depth) (cm)		Z (Vertical) (cm)		Average (cm)		
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE	No. of parameters
Configuration-1	7.37	10.37	4.64	6.48	7.06	9.77	6.36	8.87	1,094,827
Configuration-2	7.33	10.20	4.37	6.02	7.00	9.52	6.23	8.58	1,094,971
Configuration-3	6.94	9.80	4.46	6.08	6.62	9.10	6.01	8.33	1,094,971
Configuration-4	6.99	9.79	4.07	5.56	6.54	8.94	5.87	8.10	1,095,115
mmPose[33]	6.80	10.21	4.79	6.67	6.94	9.86	6.18	8.91	2,281,739

The results in this table are for 20% test data.

A recent prior study, mmPose [33], projects the point cloud to two different planes as features and then concatenates them. However, the projection step increases the number of parameters hence increases the computation cost. Decomposing features into different projections increases the model parameters linearly since we need to perform the convolution multiple times for each decomposed feature map and then concatenate them. To analyze the effect of projections, we implement the mmPose model [33] and compare it with MARS. To make a fair comparison, we reduce mmPose's feature map size from 16×16 to 8×8 since the maximum number of points per frame N_J is 64 in our dataset. As shown in Table 5, the CNN used in MARS has 1,095,115 parameters, which is half of 2,281,739 in mmPose. Moreover, the MAE of the 3-axis localization error of MARS is 5.87 cm, lower than 6.18 cm of mmPose. The result shows that MARS feature generation reduces the model complexity while obtaining higher performance. We also emphasize that mmPose requires two radars, while MARS uses only one radar, making it more practical and easier to use. Furthermore, MARS handles complex rehabilitation movements, whereas mmPose is developed to analyze joint movements during walking.

5.3.3 CNN Architecture Design. Section 4.2 presented the use of BN and max-pooling concepts in CNN architectures. This section justifies incorporating or excluding BN and max-pooling by training different models with or without them. We first train the model without BN and max-pooling as the baseline. Then, we train another model called "Baseline with BN", which adds a BN layer after each convolution layer and fully connected layer. Similarly, we train another model called "Baseline with max-pooling", which adds a max-pooling after the convolution layers. Finally, we train a model called "Baseline with both", which adds a max-pooling layer after each BN layer except the final BN layer after the fully connected layer. We observe that the "Baseline with BN" gives the best result and "Baseline with both" gives the worst, similar to the baseline, as shown in Table 6. BN successfully avoids the internal covariate shift. Max-pooling is not a good option because our model maps mmWave points to the joints point such that this task is essentially a mapping regression problem. Max-pooling introduces information loss when taking the local maximum of the features such that the model cannot leverage every joint's coordinates accurately. We then decide to keep only BN in MARS.

5.3.4 Training with User-specific or Aggregate Data. Our dataset contains close to 10,000 frames per user, which alone is sufficient to train custom user-specific models. This section moves one step forward to analyze the ability of MARS to generalize to multiple users, considering that several people in the same household can use a shared setup.

To this end, we investigate the performance between the model trained with individual users and all users. Using the same CNN architecture, we first train the models with individual user data then train a model with all users. The first four rows in Table 7 summarize the MAE and RMSE

Table 6. Comparison of Average Localization Error for 19 Human Joints Positions Position Across Models Trained using CNN Architecture with Different Components (Batch Normalization and Max-pooling)

	X (Horizontal) (cm)		Y (Depth) (cm)		Z (Vertical) (cm)		Average (cm)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Baseline	7.52	10.55	4.84	6.65	7.36	10.05	6.57	9.08
Baseline with BN	6.99	9.79	4.07	5.56	6.54	8.94	5.87	8.10
Baseline with max-pooling	7.63	10.38	4.65	6.64	7.15	9.57	6.48	8.86
Baseline with both	7.44	10.20	4.45	6.09	7.22	9.81	6.37	8.70

The results in this table are for 20% test data.

Table 7. Comparison of Average Localization Error for 19 Human Joints Position between Models Trained with Individual user and All Users

	X (Horizontal) (cm)		Y (Depth) (cm)		Z (Vertical) (cm)		Average (cm)	
	MAE	RMSE	MAE	RMSE	MAE	RMSE	MAE	RMSE
Subject 1 (M)	8.01	10.78	5.23	6.70	6.56	9.08	6.60	8.85
Subject 2 (M)	7.80	10.31	4.95	6.57	5.37	7.41	6.04	8.10
Subject 3 (M)	8.25	10.80	4.82	6.42	6.13	8.07	6.40	8.43
Subject 4 (F)	7.58	10.34	5.08	6.89	5.70	7.45	6.12	8.23
Avg. of all subjects	7.91	10.56	5.02	6.65	5.94	8.00	6.29	8.40
All subjects	6.99	9.79	4.07	5.56	6.54	8.94	5.87	8.10

The results in this table are for 20% test data.

for the test data when the CNN is trained for a single user. We observe that the maximum MAEs for the x-, y-, z-axes are 8.25 cm, 5.23 cm, and 6.56 cm, respectively. The fifth row shows that the MAE and RMSE average across all subjects. The corresponding average MAE and RMSE across all subjects and dimensions are 6.29 cm and 8.40 cm, respectively.

The last row in Table 7 shows the MAE and RMSE when a single model is trained using data from all users. The resulting modeling errors are very similar to the performances of user-specific models for each subject. We also note that the model trained for all users has a slightly higher MAE of 6.54 cm along the z-axis than x-axes. This behavior is attributed to the height differences between all our subjects (160 cm-192 cm) since the z-axis represents the vertical dimension. Overall, these results show that multiple people in the same household can easily use a shared MARS profile. We also note that multiple user profiles can also be used depending on the user preference.

5.4 Joint Angle Estimation

The angle estimation is essentially a nonlinear transformation of the coordinate estimations. Therefore, the estimation error in the joint angle is related to the localization error trend. The average MAE of MARS in estimating left elbow angle, right elbow angle, left knee angle, and right knee angles are 12°, 13°, 7°, 6°, respectively. We observe that the elbow angles have a higher error than the knee angles, as summarized in Table 8. The higher error stems from using the WristLeft and WristRight joint positions to calculate the elbow angles. Since these two joints have higher estimation errors, as discussed in Section 5.2, they increase the error in the elbow angle estimates. Moving the users closer to the radar might reduce this error since the radar can detect more hands-related points, but the system may also lose the entire body's aspect. Further improvement of elbow angle estimates will be considered in our future work.

Table 8. MAE of MARS Joint Angle Estimation

MAE of Left Elbow	MAE of Right Elbow	MAE of Left Knee	MAE of Right Knee
12°	13°	7°	6°

The results in this table are for 20% test data.

Table 9. Power and latency results for model inference of MARS on Jetson Xavier NX

	Config. 1	Config. 2	Config. 3	Config. 4	Config. 5
Online CPU	2	2	4	4	6
Max CPU frequency (MHz)	1900	1500	1400	1200	1400
Max GPU frequency (MHz)	1100	800	1100	800	1100
Total time (second)	2.5	3.2	3.7	3.9	4.1
CPU-GPU power (mW)	3921.4	2366.8	2075.7	1968.1	1950.6
Total power (mW)	6865.2	5211.4	4854.4	4723.1	4700.2
Total energy (J)	17.3	16.5	17.9	18.2	19.4
Time per frame (μs)	64.4	81.1	94.4	98.9	105.6
Energy per frame (μJ)	442.3	422.9	458.3	467.5	496.4

The upper part is the hardware configurations.

5.5 Power and Execution Time Analysis

MARS's ability to run on hardware within acceptable power and execution time is crucial for its practicality on low-power edge devices. To evaluate this ability, we implemented MARS on Nvidia Jetson Xavier NX Development Kit [29]. The board has a 6-core ARM CPU, 384 Nvidia CUDA cores, and 48 tensor processing units.

We focus on real-time model inference since the training is usually done using more powerful computing resources, and inference is more meaningful during rehabilitation exercises. The computing power of edge devices varies widely. To do a comprehensive study considering most use-cases, we set five different hardware configurations with different numbers of active CPU cores and maximum CPU/GPU operating frequencies, as shown in the upper part of Table 9. We sort five configurations in descending order of computation power. For different configurations, the total inference time for all 40,083 frames ranges from 2.5 s to 4.1 s, as shown in Table 9. The total CPU and GPU power consumption decreases from 3921.4 mW to 1950.6 mW as we move from Config. 1 to Config. 5. The corresponding total power consumption decreases from 6865.2 mW to 4700.2 mW. We find the average inference time and energy consumption by dividing these measurements into the total number of frames (40,083). The average frame processing time ranges from 64.4 μs to 105.6 μs, which shows that MARS can process well over 9,000 frames per second with less than 500 μJ energy consumption per frame. These results show that MARS provides a high-performance and energy-efficient solution for reliable and privacy-preserving home-based rehabilitation.

6 CONCLUSIONS AND FUTURE WORK

This paper presented a mmWave-based assistive rehabilitation system, called MARS, for smart healthcare. MARS can reconstruct up to 19 human joints and human skeleton in 3D space using mmWave radar without raising privacy concerns and requiring strict lighting settings. Moreover, MARS provides the users with 19 joints velocity estimations, four critical angle estimations, and ten commonly used rehabilitation posture correction feedback. It incorporates point cloud preprocessing, a CNN that outputs joint positions, and rehabilitation movement feedback to the user. It first maps the 5D time-series mmWave point cloud to a 5-channel feature map, then outputs 3D

joint positions. It finally provides joint velocity, angle estimations, and posture correction feedback. We evaluate MARS extensively using a newly produced dataset with 2.28 million reference data points from Kinect V2 and 3.81 million data points from mmWave radar. Our experimental evaluations show an average MAE of 5.87 cm for the 3D joints position estimation of MARS. Extensive ablation studies demonstrate the necessity of each component of MARS. Model inference takes only 64 μs and consumes 442 μJ energy on the Nvidia Jetson Xavier-NX board. These results show the practicality of the proposed technique running real-time on low-power edge devices. MARS paves the way for the assistive rehabilitation system based on mmWave. A demo of MARS and training/validation/test datasets are released on our GitHub page [37].

This work is one of the first steps towards a practical home-based rehabilitation system. We envision a scenario where MARS will be first trained before deployment for different target groups (e.g., age, height, and gender) with available users. Then, it can be further calibrated for new users using a mechanism similar to the existing feedback system. The proposed framework and released dataset aim at stimulating research in this area and contribute to an eventual practical solution.

REFERENCES

[1] Martín Abadi et al. 2015. TensorFlow: Large-Scale Machine Learning on Heterogeneous Systems. http://tensorflow.org/ Software available from tensorflow.org.

[2] Stefano Abbate, Marco Avvenuti, and Janet Light. 2014. Usability study of a wireless monitoring system among Alzheimer's disease elderly population. *International Journal of Telemedicine and Applications* 2014, Article 617495 (2014), 8 pages.

[3] Giovanni Abbruzzese, Roberta Marchese, Laura Avanzino, and Elisa Pelosin. 2016. Rehabilitation for parkinson's disease: Current outlook and future challenges. *Parkinsonism & Related Disorders* 22 (2016), S60–S64.

[4] Abdi T. Abdalla, Mohammad T. Alkhodary, and Ali H. Muqaibel. 2018. Multipath ghosts in through-the-wall radar imaging: challenges and solutions. *ETRI Journal* 40, 3 (2018), 376–388.

[5] Fadel Adib, Chen-Yu Hsu, Hongzi Mao, Dina Katabi, and Frédo Durand. 2015. Capturing the human figure through a wall. *ACM Trans. Graph.* 34, 6 (Oct. 2015).

[6] Md Atiqur Rahman Ahad, Anindya Das Antar, and Omar Shahid. 2019. Vision-based action understanding for assistive healthcare: A short review. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR) Workshops*. 1–11.

[7] João Antunes, Alexandre Bernardino, Asim Smailagic, and Daniel P. Siewiorek. 2018. AHA-3D: A labelled dataset for senior fitness exercise recognition and segmentation from 3D skeletal data. In *Prof. of The British Machine Vision Conference (BMVC)*. 332.

[8] Ilktan Ar and Yusuf Sinan Akgul. 2014. A computerized recognition system for the home-based physiotherapy exercises using an RGBD camera. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 22, 6 (2014), 1160–1171.

[9] Min S. H. Aung et al. 2015. The automatic detection of chronic pain-related expression: requirements, challenges and the multimodal EmoPain dataset. *IEEE Transactions on Affective Computing* 7, 4 (2015), 435–451.

[10] Ganapati Bhat, Ranadeep Deb, and Umit Y. Ogras. 2019. OpenHealth: open-source platform for wearable health monitoring. *IEEE Design & Test* 36, 5 (2019), 27–34.

[11] M. Bondarenko and Vadym I. Slyusar. 2011. Influence of jitter in ADC on precision of direction-finding by digital antenna arrays. *Radioelectronics and Communications Systems* 54, 8 (2011), 436–445.

[12] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2017. Realtime multi-person 2d pose estimation using part affinity fields. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*. 7291–7299.

[13] François Chollet et al. 2015. Keras. https://keras.io.

[14] K. Czarnecki et al. 2012. Functional movement disorders: Successful treatment with a physical therapy rehabilitation protocol. *Parkinsonism & Related Disorders* 18, 3 (2012), 247–251.

[15] Vivek Dham. 2017. Programming chirp parameters in ti radar devices. *Application Report SWRA553, Texas Instruments* (2017).

[16] Toni Giorgino, Paolo Tormene, Federico Lorussi, Danilo De Rossi, and Silvana Quaglini. 2009. Sensor evaluation for wearable strain gauges in neurological rehabilitation. *IEEE Transactions on Neural Systems and Rehabilitation Engineering* 17, 4 (2009), 409–415.

[17] Toni Giorgino, Paolo Tormene, Giorgio Maggioni, Caterina Pistarini, and Silvana Quaglini. 2009. Wireless support to poststroke rehabilitation: Myheart's neurological rehabilitation concept. *IEEE Transactions on Information Technology in Biomedicine* 13, 6 (2009), 1012–1018.

[18] A. Godfrey, R. Conway, D. Meagher, and G. ÓLaighin. 2008. Direct measurement of human movement by accelerometry. *Medical Engineering & Physics* 30, 10 (2008), 1364–1386.
[19] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. 2017. Mask r-cnn. In *in Proc. of IEEE Intl. Conf. on Computer Vision*. 2961–2969.
[20] Intel. 2014. Realsense sensor. https://www.intelrealsense.com/. accessed 29 Sep. 2020.
[21] Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980* (2014).
[22] Kyoung Bo Lee et al. 2015. Six-month functional recovery of stroke patients: A multi-time-point study. *Intl. Journal of Rehabilitation Research. Internationale Zeitschrift fur Rehabilitationsforschung. Revue internationale de recherches de readaptation* 38, 2 (2015), 173.
[23] Daniel Leightley, John Darby, Baihua Li, Jamie S. McPhee, and Moi Hoon Yap. 2013. Human activity recognition for physical rehabilitation. In *2013 IEEE International Conference on Systems, Man, and Cybernetics*. IEEE, 261–266.
[24] Filip Lemic et al. 2016. Localization as a feature of mmWave communication. In *Proc. of Intl. Wireless Communications and Mobile Computing Conference*. 1033–1038.
[25] Haipeng Liu et al. 2020. Real-time arm gesture recognition in smart home scenarios via millimeter wave sensing. *Proc. of ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 4, 4 (2020), 1–28.
[26] Zhen Meng et al. 2020. Gait recognition for co-existing multiple people using millimeter wave sensing. In *Proc. of AAAI Conference on Artificial Intelligence*, Vol. 34. 849–856.
[27] Microsoft. 2014. Kinect sensor. https://developer.microsoft.com/en-us/windows/kinect/. accessed 29 Sep. 2020.
[28] Gary Minkler and Jing Minkler. 1990. CFAR: The principles of automatic radar detection in clutter. *NASA STI/Recon Technical Report A* 90 (1990), 23371.
[29] Nvidia. 2014. Jetson Xavier NX Developer Kit. https://developer.nvidia.com/embedded/jetson-xavier-nx-devkit. accessed 29 Sep. 2020.
[30] Shyamal Patel, Hyung Park, Paolo Bonato, Leighton Chan, and Mary Rodgers. 2012. A review of wearable sensors and systems with application in rehabilitation. *Journal of Neuroengineering and Rehabilitation* 9, 1 (2012), 1–17.
[31] Deva Ramanan, David A. Forsyth, and Andrew Zisserman. 2005. Strike a pose: Tracking people by finding stylized poses. In *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, Vol. 1. IEEE, 271–278.
[32] Sandeep Rao. 2017. Introduction to mmWave sensing: FMCW radars. *Texas Instruments (TI) mmWave Training Series* (2017).
[33] Arindam Sengupta, Feng Jin, Renyuan Zhang, and Siyang Cao. 2020. Mm-pose: Real-time human skeletal posture estimation using mmWave radars and CNNs. *IEEE Sensors Journal* 20, 17 (2020), 10032–10044.
[34] Ling Shao, Jungong Han, Dong Xu, and Jamie Shotton. 2013. Computer vision for RGB-D sensors: Kinect and its applications [special issue intro]. *IEEE Transactions on Cybernetics* 43, 5 (2013), 1314–1317.
[35] Ana Ligia Silva de Lima, Tim Hahn, Luc J. W. Evers, Nienke M. De Vries, Eli Cohen, Michal Afek, Lauren Bataille, Margaret Daeschler, Kasper Claes, Babak Boroojerdi, et al. 2017. Feasibility of large-scale deployment of multiple wearable sensors in Parkinson's disease. *PLoS One* 12, 12 (2017), e0189161.
[36] Akash Deep Singh, Sandeep Singh Sandha, Luis Garcia, and Mani Srivastava. 2019. Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar. In *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*. 51–56.
[37] Sizhe An. 2021. MARS. https://github.com/SizheAn/MARS. accessed 8 Jul. 2021.
[38] Piotr Szczuko. 2019. Deep neural networks for human pose estimation from a very low resolution depth image. *Multimedia Tools and Applications* 78, 20 (2019), 29357–29377.
[39] Martijn Ten Bhömer, Oscar Tomico, and Caroline Hummels. 2013. Vigour: smart textile services to support rehabilitation. *Nordes* 1, 5 (2013).
[40] Texas Instruments. 2014. Datasheet. https://www.ti.com/lit/ds/symlink/iwr1443.pdf. accessed 8 Apr. 2021.
[41] Texas Instruments. 2014. IWR1443BOOST. https://www.ti.com/tool/IWR1443BOOST. accessed 29 Sep. 2020.
[42] Texas Instruments. 2014. mmWavetutorial. https://www.ti.com/lit/pdf/swra553. accessed 29 Sep. 2020.
[43] Texas Instruments. 2018. Zone Occupancy. https://www.ti.com/lit/pdf/tiduea7. accessed 8 Apr. 2021.
[44] Texas Instruments. 2020. mmWavefundamentals. https://www.ti.com/lit/spyy005. accessed 8 Apr. 2021.
[45] Aleksandar Vakanski, Hyung-pil Jun, David Paul, and Russell Baker. 2018. A data set of human body movements for physical rehabilitation exercises. *Data* 3, 1 (2018), 2.
[46] Qi Wang, Panos Markopoulos, Bin Yu, Wei Chen, and Annick Timmermans. 2017. Interactive wearable systems for upper body rehabilitation: A systematic review. *Journal of Neuroengineering and Rehabilitation* 14, 1 (2017), 1–21.
[47] Aner Weiss, Talia Herman, Anat Mirelman, Shirley Shema Shiratzky, Nir Giladi, Lisa L. Barnes, David A. Bennett, Aron S. Buchman, and Jeffrey M. Hausdorff. 2019. The transition between turning and sitting in patients with Parkinson's disease: A wearable device detects an unexpected sequence of events. *Gait & Posture* 67 (2019), 224–229.

[48] Zhicheng Yang, Parth H. Pathak, Yunze Zeng, Xixi Liran, and Prasant Mohapatra. 2016. Monitoring vital signs using millimeter wave. In *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*. 211–220.

[49] Mingmin Zhao, Yonglong Tian, Hang Zhao, Mohammad Abu Alsheikh, Tianhong Li, Rumen Hristov, Zachary Kabelac, Dina Katabi, and Antonio Torralba. 2018. RF-based 3D skeletons. In *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*. 267–281.

Received April 2021; revised June 2021; accepted July 2021