2024 IEEE 44th International Conference on Distributed Computing Systems (ICDCS) | 979-8-3503-8605-9/24/$31.00 ©2024 IEEE | DOI: 10.1109/ICDCS60910.2024.00102

mmHand: 3D Hand Pose Estimation Leveraging mmWave Signals

Hao Kong[*], Haoxin Lyu[†], Jiadi Yu[†§], Linghe Kong[†], Junlin Yang[†], Yanzhi Ren[‡], Hongbo Liu[‡], and Yi-Chao Chen[†]
[*]Shanghai University, Shanghai, China
Email: haokong@shu.edu.cn
[†]Shanghai Jiao Tong University, Shanghai, China
Email: {a1151501486, jiadiyu, linghe.kong, junlinyang, yichao}@sjtu.edu.cn
[‡]University of Electronic Science and Technology of China, Chengdu, China
Emai: {renyanzhi05, hongbo.liu}@uestc.edu.cn
[§]Coppresonding Author

***Abstract*—Hand pose estimation is a key support for a variety of interactive applications including user interface control, sign language understanding, virtual reality modeling, etc. Existing approaches mainly exploit wearable devices such as gloves or bracelets to estimate hand poses, which may introduce high deploying costs and intrusive user experience. Others rely on vision technologies whereas they could face complicated illuminations and privacy leakage. In this paper, we present a millimeter wave (mmWave) signal-based 3D hand pose estimation system, *mmHand*, which utilizes a mmWave radar to generate 3D hand skeletons and reconstruct 3D hand meshes. *mmHand* first leverages mmWave signals to sense a hand and pre-process the signals. Then, *mmHand* extracts spatial and temporal features using a designed attention-based hourglass network (*mmSpaceNet*) and Long Short-Term Memory (LSTM), respectively. Based on the extracted features, *mmHand* further regresses hand joints in 3D space to generate 3D hand skeletons. Finally, 3D hand meshes that continuously describe hand poses with detailed surfaces are reconstructed through a hand Model with Articulated and Non-rigid defOrmations (MANO). Extensive experiments demonstrate that *mmHand* can accurately generate 3D hand skeletons with 18.3mm mean per joint position error and 95.1% of correct key points, which indicates the effectiveness of *mmHand* on hand pose estimation.**

***Index Terms*—Millimeter wave sensing, hand pose estimation, hand joint regression, hand mesh reconstruction**

I. Introduction

As an important expression carrier, human hands can express human's rich personal needs, intentions and operations through various gestures. Hence, hand-based interaction is one of the most natural and pervasive ways to connect humans and machines together. Hand pose estimation is a kind of technique that estimates and models hand poses under various hand gestures and motions continuously, which provides the fundamental knowledge for machines to acquire interactive content. Together with the explosive growth of Internet of Things (IoT) applications, hand pose estimation has become a critical support in a lot of interaction scenarios including user interface controls, sign language understanding, virtual reality (VR), augmented reality (AR), interactive games, etc.

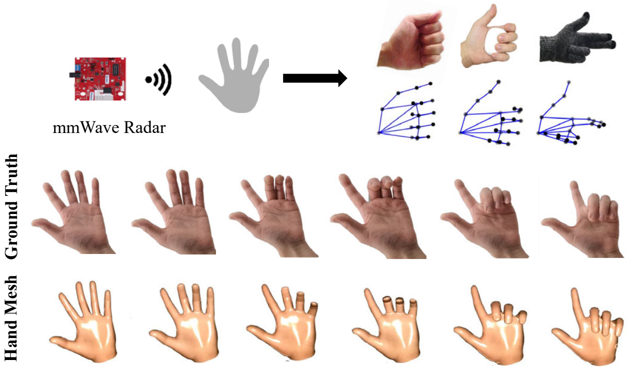

Fig. 1: Illustration of *mmHand*, which utilizes a mmWave radar to realize 3D hand pose estimation.

To meet the widespread need for interaction scenarios, academic researchers and industrial practitioners have proposed some hand pose estimation approaches. Among them, wearable-based solutions are extensive technologies that corner the market. Wearable devices such as data gloves [1] and bracelets [2] can capture highly accurate hand shapes and motions with sensitive hand-attached sensors, which act as professional hand pose estimation solutions in specialized application scenarios such as surgical operations. However, wearable devices are usually dedicated tools and bring about high deployment costs. In addition, active wearing devices may induce intrusive user experience and limit usage scenarios. Different from wearable-based approaches, leveraging images or videos to estimate hand poses also attracts a lot of attention due to the advancement of computer vision technologies. Vision-based approaches usually rely on large-scale vision datasets and deep neural networks to generate hand skeletons or synthesize hand meshes [3, 4, 5, 6, 7]. However, they strictly depend on illumination conditions of the environment and suffer from none line-of-sight scenarios. Besides, vision-based approaches may expose individual privacy hidden in the background. Therefore, a low-cost, passive, and nonintrusive solution is highly desired in hand pose estimation market.

Recently, radio frequency (RF) signals have been exploited for sensing besides communications, among which the millimeter wave (mmWave) signal is one of the most popu-

2575-8411/24/$31.00 ©2024 IEEE
DOI 10.1109/ICDCS60910.2024.00102

lar. mmWave stands out because of its fine-grained sensing capability from short wavelength and large bandwidth, which have yielded many sensing applications ranging from automotive-based object detection [8], ego-motion estimation [9], to human-centric activity recognition [10], indoor localization [11], vital sign monitoring [12], etc. Although mmWave signals have been successfully exploited in gesture recognition [13, 14, 15], most of them can only output predefined categories of gestures but cannot express dynamic 3D hand poses. A recent study [16] exploits mmWave signals to sense human forearm and therefore infer finger motions, but it ignores the shape of hand palms and cannot render realistic hand meshes. Besides, the forearm is required to always face the radar to track finger motions, which significantly limits the performance when users rotate arms. Moreover, since mmWave signals do not directly capture depth information of the hand, the depth information of the hand is inferred and the method generates pseudo-3D hand skeletons. The sensing capability of mmWave signals and high demand for 3D hand pose estimation motivate us to leverage mmWave signals to build a nonintrusive and realistic 3D hand pose estimation system. The system is robust to various lighting conditions and works in a privacy-preserving and nonintrusive manner, which can be easily deployed in real-world scenarios to enable a broad array of interactive applications, such as user interface control, VR modeling, etc. To achieve 3D hand pose estimation based on mmWave signals, we face several challenges in practice. First, since the motion of a hand is usually sophisticated and flexible, we need to robustly capture the subtle motion of hands leveraging a commercial off-the-shelf (COTS) mmWave radar. Second, due to the limited resolution and error-prone nature of mmWave signals, we need to effectively extract the multi-scale features of human hand from mmWave signals. Third, to enable practical hand estimation applications, the system should generate dynamic 3D hand skeletons and reconstruct realistic 3D hand meshes.

In this paper, we propose a nonintrusive 3D hand pose estimation system, *mmHand*, which generates 3D hand skeletons and reconstructs hand meshes continuously leveraging a COTS mmWave radar. *mmHand* leverages mmWave signals to sense a user's hand and pre-process the signals. Then, *mmHand* extracts spatial features of the hand using a designed attention-based hourglass network, *mmSpaceNet*, and further extracts temporal features of the hand using Long Short-Term Memory (LSTM). Based on the extracted features, *mmHand* regresses hand joints in 3D space to generate 3D hand skeletons with the combination of 3D loss and kinetics loss. Finally, *mmHand* reconstructs 3D hand meshes that continuously describe hand poses with more detailed surfaces through a model named hand Model with Articulated and Non-rigid defOrmations(MANO). We evaluate the performance of *mmHand* by conducting extensive experiments in real-world scenarios. The results show that *mmHand* can effectively estimate different hand poses. An illustration of the *mmHand* system is shown in Fig. 1.

We highlight our main contributions as follows:

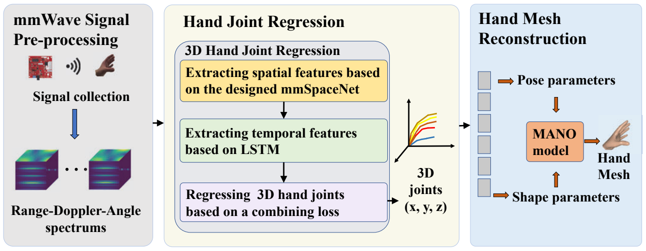
mmWave Signal Pre-processing
Signal collection
Range-Doppler-Angle spectrums
Hand Joint Regression
3D Hand Joint Regression
Extracting spatial features based on the designed mmSpaceNet
Extracting temporal features based on LSTM
Regressing 3D hand joints based on a combining loss
3D joints (x, y, z)
Hand Mesh Reconstruction
Pose parameters
MANO model
Hand Mesh
Shape parameters

Fig. 2: Architecture of *mmHand*.

- We propose a nonintrusive 3D hand pose estimation system, *mmHand*, which leverages mmWave signals to generate 3D hand skeletons and reconstruct 3D hand meshes that continuously describe hand poses.
- We design a deep learning model to effectively extract multi-scale spatial and temporal features of human hands from mmWave signals for hand pose estimation.
- We regress 21 hand joints in 3D space to generate dynamic hand skeletons, and further reconstruct realistic and continuous 3D hand meshes.
- We conduct extensive experiments involving 10 participants in real-world scenarios. The results show that *mmHand* effectively reconstructs 3D hand meshes, and achieves 18.3mm mean per joint position error and 95.1% of correct keypoints in 3D space on hand joint estimation.

II. System Overview

To realize hand pose estimation in complex and real-world scenarios, we build *mmHand*, which leverages a commercial mmWave radar to continuously sense the motion and shape of hands for 3D hand pose estimation. Fig. 2 shows the architecture of *mmHand*, which consists of following modules.

mmWave Signal Pre-processing. In this module, *mmHand* leverages mmWave signals to sense a user's hand and pre-process the signals. The mmWave radar first receives the signals reflected by the hand. Then, *mmHand* initially derives distance, velocity, and angle information through a series of FFT operations. The pre-processing of mmWave signals provides crucial insight into the posture and motion of the hand, which is the basis of hand pose estimation subsequently.

Hand Joint Regression. In this module, *mmHand* generates 3D hand skeletons from the pre-processed mmWave signals based on a designed deep learning model. Specifically, *mmHand* first extracts multi-scale spatial features that describe the posture of the hand using a designed attention-based hourglass network called *mmSpaceNet*. Then, *mmHand* extracts temporal features that describe the motion of the hand using a Long Short-Term Memory(LSTM)-based temporal model. Finally, *mmHand* regresses hand joints in 3D space and generates 3D hand skeletons based on the extracted features with the combination of 3D loss and kinetics loss.

Hand Mesh Reconstruction. In this module, *mmHand* further reconstructs 3D hand meshes using a universal parametric hand model MANO (hand Model with Articulated and Non-rigid defOrmations). By fitting the pose and shape parameters of MANO based on the regressed 3D hand joints, *mmHand*

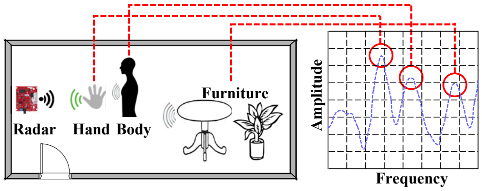

Fig. 3: Illustration of sensing human hand and other objects using mmWave signals.

generates realistic 3D meshes of the hand, which realizes dynamic hand pose estimation leveraging mmWave signals.

III. Signal Pre-Processing

To sense the posture and motion of a hand, *mmHand* first uses frequency-modulated continuous wave (FMCW) techniques on mmWave signals to measure the range, velocity, and angle of the sensing target. Specifically, a commercial-off-the-shelf (COTS) mmWave radar transmits chirp signals with linearly increasing frequencies at transmit antennas. Upon reflection from objects in the environment, the signals are captured by the receive antennas of the radar. The transmitted and received signals are then mixed in the mixer of the radar and yield intermediate frequency (IF) signals, which is expressed as

$$x_{IF}(t) = A_r \cdot e^{j2\pi[f_0 + \frac{B}{T_c}t - \frac{B}{2T_c}\tau(r,c)]}, \tag{1}$$

where f_0 is the start frequency of a chirp, B is the signal bandwidth, T_c is the chirp duration, A_r is the amplitude coefficient representing the attenuation of mmWave signals, and $\tau(r,c)$ is the delay of the received signals relative to the transmitted signals, which is determined by the propagation speed c of mmWave signals and the distance r between the object and radar.

After obtaining the IF signals, *mmHand* performs a series of pre-processing steps to calculate range, velocity and angle of a hand, respectively. The range r between objects and the radar can be denoted as $r = \frac{cfT_c}{2B}$, where f is the frequency of IF signals. However, mmWave signals may contain environmental noises, which could affect accurate sensing of human hand. Thus, we first need to eliminate environmental interference from the received mmWave signals. Since the range r is directly proportional to the frequency f, there are different frequencies on IF signals corresponding to the hand and other objects in the environment, such as the human body, furniture, etc. As shown in Fig. 3, due to different distances from the radar, the hand, the human body, and the furniture correspond to different peaks in the spectrum, and the hand is always located in the first dominant peaks because hand is usually closest to the radar in gesture interactions. To remove environmental interferences, *mmHand* filters the raw mmWave signals through an 8-order bandpass Butterworth filter and preserves signals related to the hand. Finally, by performing range-FFT on the mmWave signals, *Range-Spectrums* that describe the object in range dimension are obtained.

To calculate the velocity v of an object, the FMCW radar transmits two chirps with an interval of T_c. After performing range-FFT, the two signals peak at the same position in *Range-Spectrums* but they have a phase difference $\Delta\phi$ which corresponds to the movement of the object in vT_c. Thus, the velocity v can be calculated by $v = \frac{\lambda\Delta\phi}{4\pi T_c}$, where λ is the wavelength of signals. After performing Doppler-FFT, *Doppler-Spectrums* are obtained.

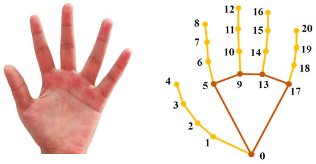

Fig. 4: The 21-hand-joint model used in *mmHand*.

To estimate the angle of arrival (AOA), at least two receive antennas are required. The distance difference Δd between an object and the two receive antennas causes a phase difference $\Delta\phi$ at the peak of range-FFT, which can be denoted as $\Delta\phi = \frac{2\pi\Delta d}{\lambda}$. According to geometric relationship, Δd is denoted as $\Delta d = lsin(\theta)$, where l is the distance between two receive antennas, and θ is the AOA. Hence, θ can be denoted as $\theta = sin^{-1}(\frac{\lambda\Delta\phi}{2\pi l})$. To locate the object in space, *mmHand* uses TDM-MIMO technologies to calculate two kinds of AOA, i.e., azimuth and elevation. The four receive antennas are always activated, while the three transmit antennas are activated alternately in sequence, generating virtual antenna arrays to measure azimuth and elevation simultaneously. Then, *mmHand* performs angle-FFT on the signals to obtain *Azimuth-Spectrums* and *Elevation-Spectrums*. However, the frequency resolution of the angular spectrum obtained by the traditional fast Fourier transform is insufficient. Since the hand only appears within a range of ±30 ° relative to the azimuth and elevation angles of the radar, *mmHand* uses zoom-FFT with a refinement factor of 2 in angle-FFT, which improves the accuracy of angle estimation. Finally, *Azimuth-Spectrums* and *Elevation-Spectrums* are obtained.

After signal pre-processing, *mmHand* constructs a four-dimensional matrix containing all spectrums, i.e. *Range-Spectrums*, *Doppler-Spectrums*, *Azimuth-Spectrum* and *Elevation-Spectrums*, which is called *Radar Cube*. The *Radar Cube* contains the range, angle and velocity information of the sensed hand.

IV. Hand Joint Regression

To represent a human hand, we employ a widely-used 21-hand-joint model, which comprises a wrist joint, 16 finger joints, and 4 fingertip joints, as illustrated in Fig. 4. For hand pose estimation based on mmWave signals, we propose a deep neural network in *mmHand* that regresses the position of the 21 hand joints in 3D space to generate 3D hand skeletons.

The input of the designed deep learning model is the *Radar Cube* (**RC**), which is generated through the pre-processed signals. **RC** is a four-dimensional matrix denoted by **RC** $\in \mathbb{R}^{F\times V\times D\times A}$, where F is the number of frames, V is the number of velocity bins, D is the number of distance bins, and A is the number of angle bins. Theoretically, inputting each frame of **RC** into the neural network can generate 3D

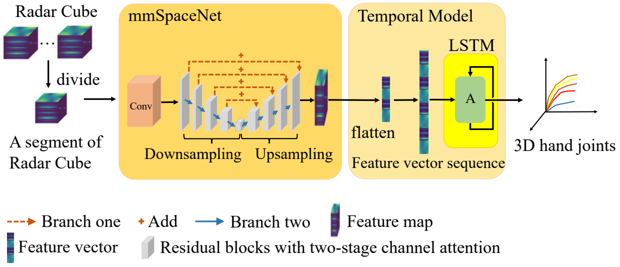

Fig. 5: The architecture of 3D hand joint regression in *mmHand*.

hand skeletons corresponding to the frame at that moment. However, the features extracted from a single frame cannot cover all the features of a hand at a certain instant, leading to an inaccurate hand joint regression. Thus, we use several consecutive frames to form a segment of *Radar Cube*, i.e. $\mathbf{X} \in \mathbb{R}^{st \times V \times D \times A}$, where st is the number of consecutive frames contained in each segment, and $\mathbf{X}$ is used as a single input to the network. The radar cube provides more details about hand motions at a certain instant and enhances the robustness of the input.

To achieve hand pose estimation using mmWave signals, it is necessary for *mmHand* to sense the spatial distribution of hands in 3D space and describe hand motion in time dimension. Hence, *mmHand* first extracts spatial and temporal features of a hand from mmWave signals, and further regresses 21 hand joints in 3D space to generate 3D hand skeletons. Fig. 5 shows the architecture of hand joint regression in *mmHand*.

A. Hand Feature Extraction

Extracting Spatial Features based on *mmSpaceNet*. Since a human hand has a relatively small reflection area and the postures of a hand are diversified, the reflected mmWave signals have a low reflection intensity and remain similar between different postures. To effectively extract spatial features of a hand, we design an attention-based hourglass network, *mmSpaceNet*, which combines shallow features with deep features to characterize a human hand from different granularities in space. As Fig. 5 shows, *mmSpaceNet* is composed of attention residual blocks and each block has two branches. One branch adjusts the number of channels without changing the size of the feature map using 1×1 convolutional layer to preserve the features of the current level. The other branch first uses convolutional layers for downsampling to extract high-dimensional and fine-grained features, and then uses deconvolutional layers for upsampling to obtain high-resolution feature maps. We adopt a two-stage channel attention mechanism as well as a spatial attention mechanism in all residual blocks, which enhances *mmSpaceNet*'s ability to extract key features.

Two-Stage Channel Attention. In order to enhance the feature extraction capability of each residual block, we propose a two-stage channel attention mechanism that combines traditional channel attention mechanisms[17, 18] with the characteristics of mmWave signals. Fig. 6 shows a diagrammatic sketch of the two-stage attention mechanism.

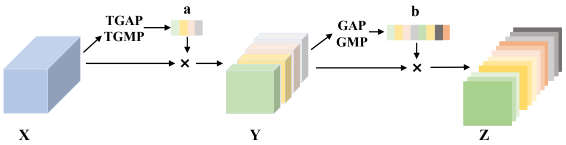

Fig. 6: Two-Stage channel attention mechanism.

Each segment of *Radar Cube* $\mathbf{X}$ can be considered as a combination of st three-dimensional matrices, where $\mathbf{X} = [X_1, X_2, ..., X_{st}] \in \mathbb{R}^{st \times V \times D \times A}$. For each $X_i \in \mathbb{R}^{V \times D \times A}$, $i = 1, 2, ...st$, the first channel attention mechanism is applied, which is represented by

$$a_i = \sigma(\mathrm{Conv}_1(\mathrm{TGAP}(X_i) + \mathrm{TGMP}(X_i))), \quad (2)$$
$$Y_i = a_i X_i, \quad (3)$$

where a_i represents the weight of the i-th frame channel, σ is the sigmoid function, Conv_1 represents a block with two convolutional layers, TGAP means the Three-dimensional Global Average Pooling, and TGMP means the Three-dimensional Global Max Pooling. Then, X_i is multiplied by the corresponding weight a_i and the output Y_i of each frame channel is obtained. After the first stage, the original input $\mathbf{X}$ is converted to $\mathbf{Y} = [Y_1, Y_2, ..., Y_{st}] \in \mathbb{R}^{st \times V \times D \times A}$ which is weighted on frame channel.

Then, we apply the second stage attention mechanism. A Global Max Pooling (GMP) and a Global Average Pooling (GAP) are performed on each velocity channel. The results of the two poolings on each channel are concatenated as channel features to preserve more information. After that, a fully connected layer FC is used to encode all channel features into a weight vector, which is multiplied with the original input $\mathbf{Y}$. This process can be expressed as

$$b_i = \sigma(\mathrm{FC}([\mathrm{GAP}(Y_i), \mathrm{GMP}(Y_i)]), \quad (4)$$
$$Z_i = b_i Y_i. \quad (5)$$

After that, the input $\mathbf{Y}$ is converted to $\mathbf{Z} = [Z_1, Z_2, ..., Z_{st}] \in \mathbb{R}^{st \times V \times D \times A}$ which is weighted on velocity channel.

3D Spatial Attention. The mmWave Radar equally receives and processes the reflected signals from all distances and directions. However, in hand pose estimation, all positions are not equally important. We focus more on finger joints and fingertips, which correspond to certain areas on the *Range-angle Spectrums*. In order to enable the network to learn the difference in *Range-angle Spectrums*, we utilize a spatial attention mechanism on the output $\mathbf{Z}$ after performing the two-stage channel attention mechanism, which can be expressed as

$$C_i = \sigma(\mathrm{Conv}_2([\mathrm{MEAN}(Z_i), \mathrm{MAX}(Z_i)])), \quad (6)$$
$$W_i = C_i Z_i, \quad (7)$$

where MEAN represents the mean of all feature maps of the form D×A computed along the velocity dimension, MAX represents the maximum value computed along the velocity dimension across all feature maps, and Conv_2 is convolutional layer used for adjusting the number of channels. After performing the spatial attention mechanism, the input $\mathbf{Z}$ is

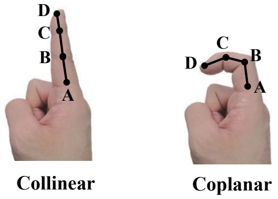

Fig. 7: Two geometric relationships of finger joints.

converted to $\mathbf{W} = [W_1, W_2, ..., W_{st}] \in \mathbb{R}^{st \times V \times D \times A}$. Compared to the initial input $\mathbf{X}$, $\mathbf{W}$ distinguishes the importance of different frame channels and velocity channels, while focusing on the key areas in *Range-angle Spectrums*.

With the above processing, a feature map that contains multiscale spatial features of the hand is obtained from mmWave signals. The feature map is used as the input of the temporal module for further extracting temporal features.

Extracting Temporal Features based on LSTM. To achieve dynamic hand pose estimation, we further design a temporal model to extract temporal features, as shown in Fig. 5. The frames of mmWave are highly correlated between adjacent frames. To reconstruct continuous motion of a hand, we introduce LSTM into the temporal model for extracting temporal features.

Specifically, each segment of *Radar Cube* $\mathbf{X}$ is processed in *mmSpaceNet* to generate a global feature map. In addition, the global feature map is flattened at the temporal model. Then, every single input of the network generates a feature vector. All feature vectors form a vector sequence as an input to LSTM for extracting temporal features.

B. Regressing 3D Hand Joints based on a Combined Loss

After extracting spatial and temporal features, *mmHand* further regresses 21 hand joints in 3D space through a combined loss to generate 3D hand skeletons. The combined loss function L_{total} is expressed as

$$L_{total} = \beta \times L_{3D} + \gamma \times L_{kine}, \tag{8}$$

where β, γ is the weight corresponding to each loss. L_{3D} is the 3D hand joint loss, which is represented by $L_{3D} = \sum_{i=0}^{20} ||h_i^{pred} - h_i^{gt}||_2$, where $h_i^{gt} = (x_i, y_i, z_i), i = 0, 1, \ldots\ldots, 20$, is the ground truth, and h_i^{pred} is the result of i-th joint predicted by the proposed network.

L_{kine} is the hand kinematic loss inspired by [19], which divides finger bending into four situations and imposes constraints on each. We simplify the relationship between finger joints to two geometric categories, i.e., collinear and coplanar, and constrain them separately. Generally, a hand is an object with segmented rigidity characteristics. Each phalange is a rigid body, and phalanges are articulated with each other through joints, allowing the hand to perform various motions. We use A, B, C, and D to represent three phalanges and one fingertip, where A is the finger root. Each joint has its 3D coordinates. When the finger is straightened, the four joints are collinear. When the finger is bent, the four joints are non-collinear, but they are still coplanar. Fig. 7 shows two types of situations. L_{kine} can be denoted as $L_{kine} = \lambda L_{cop} + (1 - \lambda) L_{col}$. λ is equal to 1 in collinear cases while 0 in coplanar cases. L_{col} represents the collinear loss and L_{cop} represents the coplanar loss. For collinear cases, the length between the phalanges satisfies $||B - A|| + ||C - B|| + ||D - C|| < (\phi + 1)||D - A||$, where ϕ is set to 0.01. Meanwhile, the angle between the vector corresponding to each phalange and the direction vector $\mathbf{e}$ of the finger should be small enough, which means $t < \cos(\overrightarrow{AB}, \mathbf{e}_d) < 1$. t is a number close enough to 1, which is set to 0.99 in *mmHand*. This also holds for $\overrightarrow{BC}$ and $\overrightarrow{CD}$. Therefore, the loss of collinear cases is expressed as

$$\begin{aligned} L_{col} &= \max(||AB|| + ||BC|| + ||CD|| - 1.01||AD||, 0) \\ &+ \max\{p - \frac{\overrightarrow{AB} \cdot \mathbf{e}_d}{||\overrightarrow{AB}||}, 0\} + \max\{p - \frac{\overrightarrow{BC} \cdot \mathbf{e}_d}{||\overrightarrow{BC}||}, 0\} \\ &+ \max\{p - \frac{\overrightarrow{CD} \cdot \mathbf{e}_d}{||\overrightarrow{CD}||}, 0\}. \end{aligned} \tag{9}$$

For coplanar cases, the direction vector of each phalange is orthogonal to the plane normal vector $\mathbf{e}_n$. Hence, the loss of coplanar cases can be expressed as $L_{cop} = \overrightarrow{AB} \cdot \mathbf{e}_n + \overrightarrow{BC} \cdot \mathbf{e}_n + \overrightarrow{CD} \cdot \mathbf{e}_n$.

Under the supervision of the combined loss, *mmHand* obtains the 3D position of 21 hand joints using fully-connected layers, which realizes 3D hand skeleton generation.

V. Mesh Reconstruction

With the 3D hand skeletons of 21 joints, we further generate realistic 3D hand meshes. 3D hand meshes present a hand with more geometric details and finer expressiveness, which can realistically exhibit the posture and motion of the hand.

Hand Model. With the development of 3D scanning technology, parametric hand models have emerged to attain more authenticity and accuracy in hand pose estimation [20]. Parametric hand models are based on the anatomical structure and kinematics principles of a human hand, modeling the hand as a 3D model controlled by parameters. By controlling the motion of joints and the shape of the hand through parameters, the model can fit any possible hand shapes and postures. In *mmHand*, we use a model named hand Model with Articulated and Non-rigid defOrmations (MANO) [21] to reconstruct 3D hand meshes. MANO is developed based on a model for human body called Skinned Multi-Person Linear Model(SMPL) [22]. MANO considers the complexity and flexibility of hands, and uses a mathematical model to describe finger poses and motions. Specifically, a differentiable function $M(\boldsymbol{\beta}, \boldsymbol{\theta})$ uses a set of parameters $\boldsymbol{\beta} \in \mathbb{R}^{10}$ to control the shape, and a set of parameters $\boldsymbol{\theta} \in \mathbb{R}^{21 \times 3}$ to control the pose of the generated hand. $\boldsymbol{\beta}$ is the coefficients of a shape principal component analysis base learned from hand scans, and $\boldsymbol{\theta}$ is joint rotations in axis-angle representation, which can be expressed as

$$M(\boldsymbol{\beta}, \boldsymbol{\theta}) = W(T_p(\boldsymbol{\beta}, \boldsymbol{\theta}), J(\boldsymbol{\beta}), \boldsymbol{\theta}, \mathcal{W}), \tag{10}$$

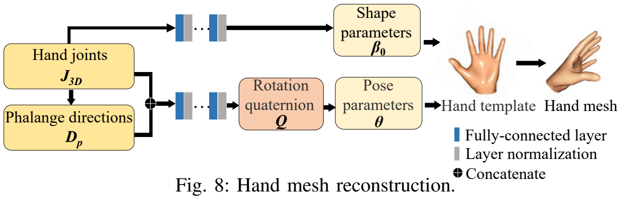

Fig. 8: Hand mesh reconstruction.

where $W(\cdot)$ is a linear blend skinning function [23], T_p is a deformed template hand mesh, $J(\boldsymbol{\beta})$ is the location of hand joints, and $\mathcal{W}$ is the skinning weights. The deformed template T_p is obtained by applying parameters to a standard template, which can be denoted as

$$T_p(\boldsymbol{\beta}, \boldsymbol{\theta}) = \vec{\boldsymbol{T}} + B_s(\boldsymbol{\beta}) + B_p(\boldsymbol{\theta}), \tag{11}$$

where $\vec{\boldsymbol{T}}$ is a standard template that represents a preset pose(T-pose) of the model, $B_s(\boldsymbol{\beta})$ and $B_p(\boldsymbol{\theta})$ are shape and pose blend shapes, respectively.

Mesh Reconstruction. To reconstruct 3D hand meshes based on MANO, we first determine the pose parameters $\boldsymbol{\theta}$ and the shape parameters $\boldsymbol{\beta}$, respectively. Then, the standard template $\vec{\boldsymbol{T}}$ is deformed using these parameters according to Eq.(11) to generate final 3D hand meshes. Fig. 8 shows the process of reconstructing 3D meshes using 21 hand joints J_{3D}.

The spatial distribution of hand skeleton joints represents the overall size and inner geometry of hand shapes. Hence, there is a mapping relation between the reconstructed skeletons and hand shapes. *mmHand* utilizes the reconstructed skeletons as the input, and adopts three fully-connected layers with layer normalization to output shape parameters $\boldsymbol{\beta}$ for virtualizing the hand shape. Then, we infer the joints' rotation parameters $\boldsymbol{\theta}$. Inferring the rotation of all hand joints $\boldsymbol{\theta}$ based on the 3D skeleton is an inverse kinematics problem [24]. To solve such a problem end-to-end, $mmHand$ uses a deep learning algorithm to learn the correspondence between the coordinates of hand joints and joint rotations. Specifically, *mmHand* employs fully-connected layers with layer normalization to infer the pose parameters $\boldsymbol{\theta}$. *mmHand* calculates the direction vector of the phalanges $D_p \in \mathbb{R}^{20\times3}$ from the 3D coordinates of 21 hand joints J_{3D}. Then, D_p and J_{3D} are flattened into a vector and concatenated as inputs to the neural network. Explicitly providing the direction of the phalanges is helpful for the network to predict joint rotations more accurately. To achieve higher computational efficiency, the network outputs the rotation quaternions $\mathbf{Q} \in \mathbb{R}^{21\times4}$ for all joints. Then, $\mathbf{Q} \in \mathbb{R}^{21\times4}$ is converted into the corresponding axis-angle representation $\boldsymbol{\theta}$.

The hand template $\vec{\boldsymbol{T}}$ takes the pose parameters $\boldsymbol{\theta}$ and the shape parameters $\boldsymbol{\beta}$ as input. With the deformed template, *mmHand* generates a 3D hand mesh from the 3D hand joints to express the pose of human hand realistically.

VI. Evaluation

In this section, we conduct experiments to evaluate *mmHand* in real environments.

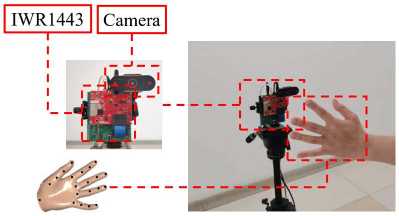

Fig. 9: Experimental setup.

A. Evaluation Setup

mmHand is implemented using a commercial off-the-shelf (COTS) mmWave radar (Texas Instruments (TI) IWR1443 [25]) connected with a data capture card (TI DCA1000EVM [26]). The mmWave radar utilizes 3 transmit antennas and 4 receive antennas to form a virtual antenna array based on TDM-MIMO technology. It transmits chirp signals with the frequency range from 77GHz to 81GHz. The cycle time of a chirp is set to $80us$. We sample 64 times on each chirp. In each frame, the 3 transmit antennas send chirps in turn and cycle 64 times. TI mmWave Studio is installed on a desktop computer equipped with an Intel Core i5-10440F processor to interact with the mmWave radar. The designed deep learning model is trained using an NVIDIA RTX 3090 Ti graphics card. The ground truth is captured by a depth camera. We use MediaPipe Hands [27] to generate 21 hand joints from images as the ground truth.

Fig. 9 shows the experimental setup where the mmWave radar and the camera are placed in the same position. They are simultaneously activated to collect mmWave signals and images respectively. We recruited a total of 10 volunteers to participate in the evaluation, including 5 male volunteers and 5 female volunteers aged between 20 and 50 years old. The volunteers came in a variety of heights ranging from 1.65m to 1.85m and different body types including lean, moderate, and slightly overweight. When collecting data, the volunteers stood in front of the radar and kept their hands within a range of $20cm$ to $40cm$ toward the radar. The volunteers performed continuous hand gestures, i.e., the interaction gestures and counting gestures, which are non-predefined and most common daily gestures. The mmWave radar and camera continuously collected the mmWave data and labels (i.e., coordinates of 21 hand joints). We collected a total of 150,000 valid frames of mmWave data and corresponding ground truth labels from each volunteer. The experiment was conducted in 3 different experimental environments, including classrooms, corridors, and playgrounds. In addition, to evaluate the robustness of the model and its performance in some special situations, we also collected a small amount of data on volunteers wearing different gloves and holding objects in their hands for testing.

We apply 5-fold cross-validations in the training and testing process. Specifically, the data of 10 volunteers is divided into 5 sub-datasets. Each sub-dataset contains 2 volunteers' data. In the k-th round of cross validations, the k-th sub-dataset is retained as the testing set, and the other 4 sub-datasets are used as the training set for model training. The cross-validations are designed to evaluate the performance taking into account

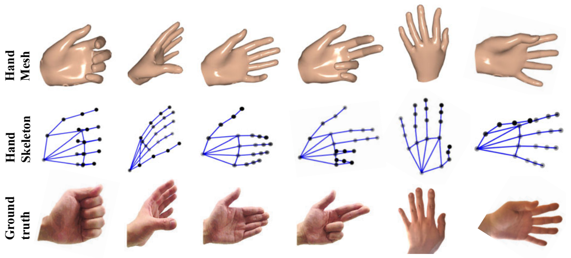

Fig. 10: Examples of hand meshes and hand skeletons for different gestures.

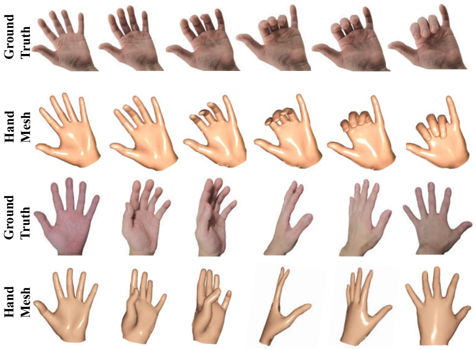

Fig. 11: Examples of hand mesh reconstruction for continuous gestures.

the difference between users and gestures. In model training, the initial learning rate is set to 0.001 and follows the cosine learning rate decay method. The batch size for training is 16, and the model is trained for a total of 500 epochs.

We use the following evaluation metrics:

- *Mean Per Joint Position Error (MPJPE)* is the mean per joint position error measured by Euclidean distance (mm) between the predicted hand joints and the ground truth, which can be denoted as

$$MPJPE = \frac{1}{N}\sum_{i=1}^{N} ||J_i^p - J_i^t||, \tag{12}$$

where N is the number of hand joints, J_i^p is predicted hand joints and J_i^t is the ground truth.

- *The Percentage of Correct Keypoints in 3D Space (3D-PCK)* is the percentage of correctly predicted hand joints under different thresholds, which can be denoted as

$$PCK_k = \frac{\sum_i \delta(\frac{d_i}{d} < T_k)}{\sum_i 1}, \tag{13}$$

where T_k is a manually set threshold, d_i is the Euclidean distance between the predicted value and the ground truth of the i-th joint, d is the scale normalization factor, and δ is an indicate function.

- *The Area Under the Curve (AUC)* is the area under the *3D-PCK* curve.

B. Overall Performance

We first evaluate the overall performance of *mmHand* in 3D hand skeleton generation and 3D hand mesh reconstruction. Fig. 10 shows examples of hand skeletons and hand meshes of different gestures respectively. It can be seen that the 21 hand joints accurately depict the corresponding poses of the hand. Moreover, the 3D hand meshes present realistic 3D animations that are consistent with the user's hand poses. *mmHand* can also capture and reconstruct continuous hand poses. Fig. 11 shows examples of hand mesh reconstruction for two continuous gestures. The hand meshes effectively exhibit the gradual change of hand poses for continuous hand gestures.

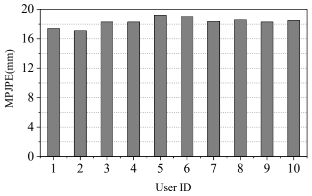

Fig. 12: Per-participant MPJPE.

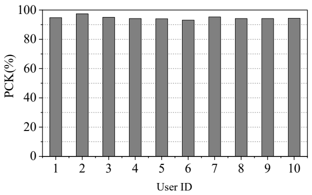

Fig. 13: Per-participant 3D-PCK.

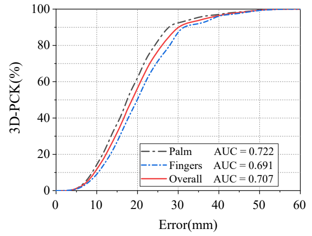

Fig. 14: 3D-PCK under different error thresholds.

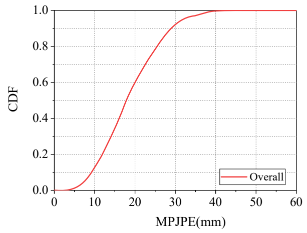

Fig. 15: CDF of MPJPE.

We further quantitatively evaluate *mmHand* by measuring MPJPE, 3D-PCK, and AUC of the 21 hand joints. Fig. 12 and Fig. 13 show the MPJPE and the 3D-PCK for each user respectively, where the threshold of 3D-PCK is 40*mm*. In general, *mmHand* achieves an average of 18.3*mm* MPJPE and 95.1% 3D-PCK with average standard deviations of 2.96*mm* and 1.17%, respectively. The results indicate that *mmHand* can accurately regress 21 hand joints with low mean errors. Furthermore, it can be seen from Fig. 12 and Fig. 13 that the differences in MPJPE and 3D-PCK between each user are insignificant. For instance, the differences of MPJPE and 3D-PCK between user 2 (with the lowest MPJPE and highest 3D-PCK) and user 6 (with the highest MPJPE and lowest 3D-PCK) are only 2.9*mm* and 3.3% respectively. This demonstrates the effectiveness and robustness in hand joint regression of *mmHand* for different individuals.

To accurately evaluate *mmHand*'s performance on regressing different hand joints, we divide 21 hand joints into the palm joints and the finger joints, and then evaluate the average 3D-PCK and AUC for all users. Fig. 14 shows the 3D-PCK of *mmHand*'s hand joint regression with the thresholds ranging from 0*mm* to 60*mm*. It can be observed that the 3D-PCK rapidly increases as the threshold increases. The overall 3D-PCK reaches 95.1% when the threshold is $40mm$. We also calculate the AUC of 3D-PCK curve, where a larger AUC indicates a better performance. The results show that *mmHand* has an overall AUC of 0.707, which achieves good performance on hand joint regression. Fig. 15 shows the cumulative distribution function (CDF) of MPJPE for all the hand joints. It can be seen that 90.2% of the MPJPE of the predicted hand joints are within $30mm$. Besides, we can also

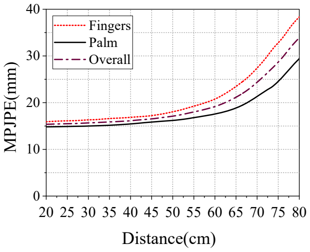

Fig. 16: MPJPE in different distances.

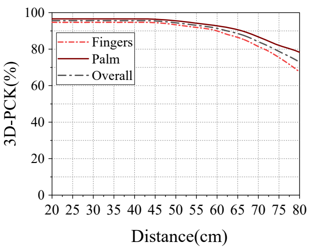

Fig. 17: 3D-PCK in different distances.

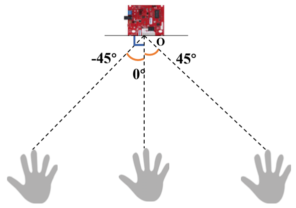

Fig. 18: Experimental setup of different angles.

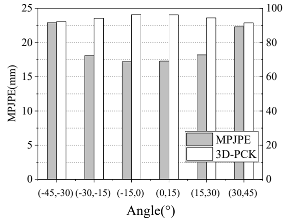

Fig. 19: MPJPE and 3D-PCK in different angles.

observe from the results that there is a bit of difference in the performance of hand joint regression for different parts of the hand. The reason is that the palm lacks flexible deformation in gestures, resulting in relatively stable hand joints. On the contrary, fingers are usually flexible and interact with each other to form various gestures, making it more difficult to regress finger joints.

C. Comparison with Existing Methods

We compare the performance of *mmHand* with several existing methods including vision-based and wireless signal-based solutions. The vision-based methods include Cascade [28], CrossingNet [29], DeepPrior++ [30] and HBE [31], which are implemented to calculate the MPJPE on two public 3D hand pose datasets (MSRA DataSet [28] and ICVL DataSet [32]). Since it is difficult to build a one-to-one mmWave dataset relative to MSRA and ICVL for a fair comparison, we utilize our self-collected mmWave dataset and present the comparison with existing vision methods. The two wireless signal-based methods are mm4Arm [16] and HandFi [33], in which mm4Arm utilizes mmWave signals and HandFi utilizes WiFi signals. Although the two wireless signal-based methods lack sources to completely reproduce for comparison, we still refer to the experimental setup of the two methods and collect mmWave data following their experimental setups for a relatively fair comparison. That is, in the lab environment, the users perform same hand gestures following other settings illustrated in the two works [16, 33], while our mmWave radar continuously collects mmWave signals. We use the mmWave data and the captured ground truth labels to output MPJPE and compare them with the typical results shown in mm4Arm and HandFi respectively.

Table I shows the MPJPE of the 6 existing methods and *mmHand*. The comparison with the 4 vision methods utilizes the MPJPE results in our above experiments, and the comparison with the 2 wireless signal-based methods utilizes the results obtained from the data collected following their experimental setups respectively. We can first see that although the MPJPE of *mmHand* is slightly inferior to other cutting-edge vision methods due to the resolution limitations of mmWave signals, the difference is insignificant. For example, the difference of MPJPE between the result of *mmHand* and the average value 10.94*mm* of these visual methods is within 10*mm*. In comparison with the wireless signal-based methods, it can be seen that mm4Arm achieves a superior performance of MPJPE utilizing their self-collected data compared to our method. However, it requires users' forearms to always face the radar, which may affect user experience in various interaction scenarios. Besides, our method achieves a similar performance to HandFi with each self-collected data. The results indicate that *mmHand* can also achieve effective hand joint regression comparable to vision solutions and other wireless signal solutions.

TABLE I: **MPJPE** of *mmHand* and existing methods.

Methods	Dataset	MPJPE	mmHand
Cascade[28]	MSRA	15.2	18.3
	ICVL	9.9	
CrossingNet [29]	MSRA	12.2	
	ICVL	10.2	
DeepPrior++[30]	MSRA	9.5	
HBE[31]	ICVL	8.62	
mm4Arm	Self-collected	4.07	20.4
HandFi	Self-collected	20.7	19.0

D. Impact of Distance

The distance between the radar and the sensed target, i.e., a user's hand, may affect the quality of signal reflection and cause variational signal patterns. Besides, distance also determines user experience of hand gesture-based human-computer interaction. We evaluate the impact of the distance between user's hand and mmWave radar. In our experimental setups, each user's hand is located within the distance from 20cm to 40cm for model construction of *mmHand*. To evaluate the impact of different distances, each user's hand is located with a distance to the radar between 20cm and 80cm. Fig. 16 and Fig. 17 show the MPJPE and 3D-PCK for hand joint regression under different distances respectively, where the threshold of 3D-PCK is set to 40mm. It can be seen from the two figures that the overall MPJPE and 3D-PCK are relatively stable when their distance is between 20cm and 60cm. When the distance exceeds 60cm, MPJPE gradually increases while PCK gradually decreases. From another perspective, the MPJPE of the palm is relatively smaller than that of the fingers, and the 3D-PCK of the palm is relatively larger than that of the fingers, which means regressing the palm joints is more accurate than the finger joints at different distances.

E. Impact of Angle

We evaluate the performance of *mmHand* in different angles of a user's hand toward the mmWave radar. In the experiment, a user's hand is located with the angles from -45° to 45° as shown in Fig. 18. We take 15° as a step and quantitatively evaluate MPJPE and 3D-PCK from different angles. Fig. 19

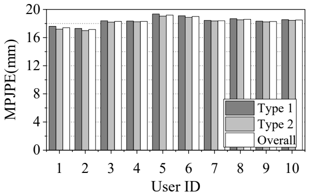

Fig. 20: Per-participant MPJPE for type 1 and type 2.

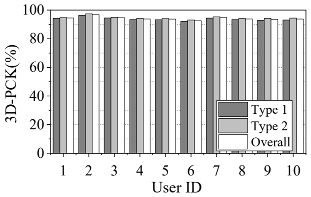

Fig. 21: Per-participant PCK for type 1 and type 2.

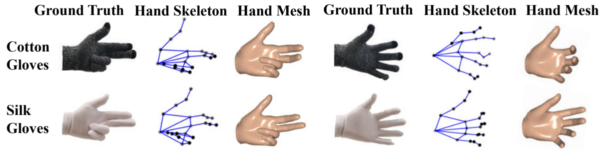

Fig. 22: Examples of hand pose estimation when users wear gloves.

shows the MPJPE and 3D-PCK from -45° to 45° with 15° steps in the six angle scopes respectively, where the distance between the radar and the hand is set to 40cm and the threshold of 3D-PCK is set to 40mm. We can see that the errors of hand joints increase as the absolute value of the angle increases. Especially, when the angle exceeds 30°, the errors of hand joint regression significantly rise. The reason is that the sensitivity of angle estimation decreases as the absolute value of the angle increases according to the principle of angle estimation. Although there are differences between different angles, the average MPJPE and PCK are only 17.95mm and 95.78$\%$ respectively when the angle is within -30° and +30°, which can effectively regress hand joints and generate hand poses. Hence, *mmHand* is also suitable for some practical interactive applications that require flexible angles of the hand's position.

F. Impact of Human Body

When a user is in front of a mmWave radar, the user's body could affect the propagation and reflection of mmWave signals, which may result in a significant impact on hand pose estimation. Hence, we further evaluate the impact of human body on the performance of *mmHand*. Two types of experiments where a user's body is in different positions are conducted for the evaluation. Specifically, in type 1, a user stands in front of the radar with the hand outstretched forward to perform various gestures. In type 2, a user stands on the side of the radar and the hand is reached out in front of the radar. Fig. 20 and Fig. 21 show the MPJPE and 3D-PCK of each user under the two types of experiments. When users stand in front of the radar, the overall MPJPE reaches 19.1mm and the overall 3D-PCK is 93.6$\%$. When users stand on the side of the radar, the overall MPJPE is 18.1mm and the overall 3D-PCK is 95.4$\%$. The differences in performance between the two types of experiments are insignificant. The reason is that the distance between the hand and radar is different from the distance between the human body and radar. *mmHand* eliminates most signals unrelated to the hand through filtering during signal pre-processing. Hence, the capability of hand pose estimation is less affected by the position of human body.

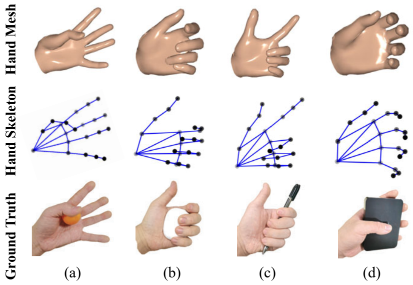

Fig. 23: Examples of hand pose estimation when users hold different objects.

G. Impact of Gloves

We conduct experiments to evaluate the impact of different gloves on hand pose estimation. Two kinds of gloves, i.e., silk gloves and cotton gloves, are worn by the users respectively in the experiment. The collected data is directly used as a testing dataset to verify the accuracy of *mmHand* on regressing 21 hand joints. Fig. 22 shows the result of hand joint regression and hand mesh reconstruction of users wearing the two gloves. It can be seen that *mmHand*'s prediction of the palm is relatively accurate, but there is a flaw in the prediction of fingers with some joints leaning together. The overall MPJPE on the two kinds of gloves is 28.6mm and the overall 3D-PCK is 86.3$\%$. Compared to the case of not wearing gloves, the accuracy of hand joint regression slightly decreases when users wear gloves because the materials of gloves could also be captured by mmWave signals and cause distortion of the sensed hand. This also leads to a slight deviation in the generation of hand meshes. However, despite the decline in accuracy, *mmHand* can still generate hand joints that reflect the basic pose of the hand.

H. Impact of Handheld Object

We evaluate *mmHand* when users hold objects. In the experiment, the users hold 4 objects respectively, i.e., a table tennis ball, a headphone case, a pen, and a power bank. Fig. 23 shows the examples of *mmHand* in estimating 3D hand poses when a user's hand holds an object. It can be seen from the result in Fig. 23(a) and 23(b) that when the objects held by the user are small and mainly located in the palm area, *mmHand* can accurately regress 21 hand joints and reconstruct 3D hand meshes. The reason is that the objects only cause slight interference with the reflected signals. Besides, since the objects are located in the center of the hand, they mainly affect the palm and the fingers are less influenced, which can still accurately estimate fingers. However, if the handheld objects affect the reflected signals in the finger area, or the area of the object covering the hand is very large, *mmHand* may suffer from performance decline. For example, Fig. 23(c) shows that *mmHand* mistakenly infers the pen as a finger, and Fig. 23(d) shows that the fingers generated by *mmHand* cannot correspond to the actual situation. This is caused by the interference of signal reflection of objects.

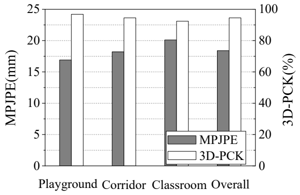

Fig. 24: MPJPE and 3D-PCK in different environments.

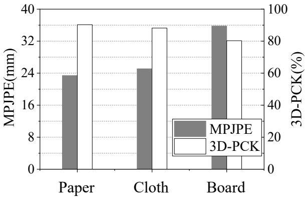

Fig. 25: MPJPE and 3D-PCK under different occlusions.

I. Impact of Environment

To explore the impact of background interferences, we present the respective performance of *mmHand* in the three environments, i.e., a playground, a corridor and a classroom. The playground is a large empty area. The corridor has an empty static background with a few people. The classroom has complex static background and also dynamic people moving around. Fig. 24 shows the performance of MPJPE and 3D-PCK in the three environments. It can be seen that the difference between different environments is insignificant. For example, the difference between MPJPE in the playground and the classroom is only 3.2*mm*. This is because *mmHand* can localize the range of hand by performing bandpass filtering on mmWave signals, which ignores the background interferences and focuses on hand sensing.

J. Impact of Obstacle

We evaluate *mmHand* under obstacle scenarios, i.e., an object located between the radar and hand to block the line-of-sight propagation, which can show whether our proposed solution can overcome the defect of vision-based methods. We use an A4 paper, a piece of cloth, and a thin board as obstacles respectively, and calculate MPJPE and 3D-PCK of the generated hand joints in the scenario. Fig. 25 shows the performance of MPJPE and 3D-PCK with different obstacles. The ground truth is obtained from the same gestures' repeated performance in line-of-sight scenarios captured by cameras. It can be seen that different obstacles have different impacts on hand pose estimation. The MPJPE under A4 paper and cloth and 23.4mm and 25.1mm respectively, which are slightly larger errors compared to non-obstacle scenarios. The performance of *mmHand* under the thin wood board suffers from a decline, i.e., 35.8mm MPJPE and 80.3% 3D-PCK on average. The results indicate that *mmHand* can generate hand poses under some materials such as paper and cloth even if the line-of-sight propagation is blocked. Hence, *mmHand* provides an illumination-robust and none line-of-sight solution for gesture-based interactions.

K. Time Consumption

We evaluate the time consumption of *mmHand*. Since *mmHand* first generates hand skeletons and then hand meshes, we analyze the time consumption of the two steps respectively. Fig. 26 shows the cumulative distribution function (CDF) of the time consumption for *mmHand* in generating hand skeletons, hand meshes, and the overall time respectively. The average time consumption of 3D hand skeleton and 3D mesh reconstruction is $459.6ms$ and $353.1ms$ respectively. Compared to only hand skeleton reconstruction, generating hand meshes does not introduce significant extra delay. Moreover, the average overall time consumption is $812.7ms$ and 90% of the overall time consumption is under $810ms$. The above results indicate that *mmHand* can reconstruct hand poses with meshes by a relatively small delay, providing satisfactory user experience in using the system.

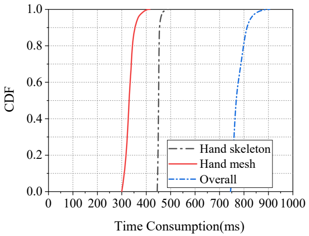

Fig. 26: CDF of time consumption.

VII. Related Work

In this section, we review works related to *mmHand*.

Wearable-based Hand Pose Estimation. Hand pose estimation has been a hot topic in the past decade. Early works exploit wearable devices especially data gloves to capture the shape and pose of hands and reconstruct 3D hands [1, 2]. Due to the highly sensitive sensors directly attached to the hand, wearable-based approaches usually achieve high accuracy in estimating hand joints. However, users need to wear on-body devices for hand pose estimation, which brings about intrusive user experience. Also, the cost of such devices is usually high, which limits their promotion and wide application.

Vision-based Hand Pose Estimation. Recently, with the boom in computer vision technology, vision-based methods dominate hand pose estimation market. Some works [4] rely on deep learning to generate 3D skeletons of a hand with images. They build deep learning models and train them on large-scale public datasets, which usually achieve high accuracy on hand joint regression. With the increasing demand for the quality of hand pose estimation, some works [3, 5, 6, 7] further reconstruct 3D hand meshes. They utilize datasets with 3D annotations to generate realistic hand meshes from RGB images. However, vision-based methods are highly dependent on lighting conditions and usually fail when users wear gloves. Besides, vision-based methods may pose privacy breaches for users, which is catching the increasing attention of people.

Wireless Sensing Technologies and Applications. Nowadays, wireless sensing technology has emerged in IoT scenarios. Wireless signal-based sensing attracts a lot of attention and yields many applications, such as user authentication [34, 35], respiratory and heartbeat monitoring [36, 37], sound sensing [38], autonomous driving [8], etc. Recent works [39, 40, 41, 42, 43] exploit mmWave signals for 3D human pose estimation and body mesh reconstruction. However, they do not pay attention to more subtle motions, such as gestures. Compared to human posture estimation, the motion of hand is more subtle and complex, which cannot be simply realized by direct parameter estimation. Early work [44] utilizes mmWave radar to track hand motions and recognize gestures. A recent study [16] exploits mmWave signals to sense human forearm and therefore infer finger motions, but it ignores the shape of hand palms and cannot render realistic hand meshes. Besides, the forearm is required to always face the radar to track finger motions, which limits the performance when users rotate their arms. Moreover, since mmWave signals do not directly capture depth information of hand, the depth information of hand is inferred and the method only generates pseudo 3D hand skeletons. Another work [45] implements occlusion-robust hand pose estimation using RF sensors. This method focuses more on estimating static hand postures in the presence of obstacles through dedicated devices, while our work achieves dynamic and realistic 3D hand reconstruction under various gestures. A recent work [33] utilizes WiFi signals to construct 3D hand skeletons. However, using WiFi devices to sense hand gestures requires users to reach out and put the hand between the transmitting and receiving ends, and the human body needs to move away from the signal sources. Differently, *mmHand* works even if the user faces the radar because the hand and body can be separated from different ranges.

Compared to existing related works, *mmHand* utilizes mmWave signals to construct realistic 3D hand poses, which is nonintrusive and robust to many scenarios.

VIII. Conclusion

In this paper, we propose a hand pose estimation system, *mmHand*, which uses a COTS mmWave radar to generate 3D hand skeletons, and reconstruct 3D hand mesh continuously. *mmHand* first leverages an attention-based hourglass network *mmSpaceNet* to extract multi-scale spatial features of the hand, and uses LSTM to extract temporal features. After that, *mmHand* regresses hand joints in 3D space to generate 3D hand skeletons, and finally reconstructs 3D hand meshes using the MANO model. Extensive experiments in real environments demonstrate the effectiveness of *mmHand* on hand pose estimation.

Acknowledgement

This work is supported by Artificial Intelligence Technology Support Special Project of Science and Technology Commission of Shanghai Municipality (No. 22DZ1100103), and National Natural Science Foundation of China (No. 62172277).

References

[1] R. Y. Wang and J. Popović, "Real-time hand-tracking with a color glove," *ACM transactions on graphics (TOG)*, vol. 28, no. 3, pp. 1–8, 2009.

[2] F. Hu, P. He, S. Xu, Y. Li, and C. Zhang, "Fingertrak: Continuous 3d hand pose tracking by deep learning hand silhouettes captured by miniature thermal cameras on wrist," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies (IMWUT)*, vol. 4, no. 2, pp. 1–24, 2020.

[3] D. Kulon, H. Wang, R. A. Güler, M. Bronstein, and S. Zafeiriou, "Single image 3d hand reconstruction with mesh convolutions," *arXiv preprint arXiv:1905.01326*, 2019.

[4] C. Xu and L. Cheng, "Efficient hand pose estimation from a single depth image," in *Proceedings of the IEEE international Conference on Computer Vision (ICCV)*, Sydney, New South Wales, Australia, 2013.

[5] X. Chen, Y. Liu, Y. Dong, X. Zhang, C. Ma, Y. Xiong, Y. Zhang, and X. Guo, "Mobrecon: Mobile-friendly hand mesh reconstruction from monocular image," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, New Orleans, Louisiana, USA, 2022.

[6] X. Chen, Y. Liu, C. Ma, J. Chang, H. Wang, T. Chen, X. Guo, P. Wan, and W. Zheng, "Camera-space hand mesh recovery via semantic aggregation and adaptive 2d-1d registration," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Virtual Conference, 2021.

[7] X. Tang, T. Wang, and C.-W. Fu, "Towards accurate alignment in real-time 3d hand-mesh reconstruction," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, Virtual Conference, 2021.

[8] Z. Wei, F. Zhang, S. Chang, Y. Liu, H. Wu, and Z. Feng, "Mmwave radar and vision fusion for object detection in autonomous driving: A review," *Sensors*, vol. 22, no. 7, p. 2542, 2022.

[9] S. H. Cen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Queensland, Australia, 2018.

[10] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "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(mmNets)*, Los Cabos, Mexico, 2019.

[11] A. Olivier, G. Bielsa, I. Tejado, M. Zorzi, J. Widmer, and P. Casari, "Lightweight indoor localization for 60-ghz millimeter wave systems," in *2016 13th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*. London, UK: IEEE, 2016.

[12] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Monitoring vital signs using millimeter wave," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing (MobiHoc)*, Paderborn, Germany, 2016.

[13] Y. Li, D. Zhang, J. Chen, J. Wan, D. Zhang, Y. Hu, Q. Sun, and Y. Chen, "Towards domain-independent and real-time gesture recognition using mmwave signal," *IEEE Transactions on Mobile Computing (TMC)*, 2022.

[14] J.-T. Yu, L. Yen, and P.-H. Tseng, "mmwave radar-based hand gesture recognition using range-angle image," in *2020 IEEE 91st Vehicular Technology Conference (VTC2020-Spring)*. virtual conference: IEEE, 2020.

[15] Y. Ren, J. Lu, A. Beletchi, Y. Huang, I. Karmanov, D. Fontijne, C. Patel, and H. Xu, "Hand gesture recognition using 802.11 ad mmwave sensor in the mobile device," in *2021 IEEE Wireless Communications and Networking Conference Workshops (WCNCW)*. Nanjing, China: IEEE, 2021.

[16] Y. Liu, S. Zhang, M. Gowda, and S. Nelakuditi, "Leveraging the properties of mmwave signals for 3d finger motion tracking for interactive iot applications," *Proceedings of the ACM on Measurement and Analysis of Computing Systems (POMACS)*, vol. 6, no. 3, pp. 1–28, 2022.

[17] J. Hu, L. Shen, and G. Sun, "Squeeze-and-excitation networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Salt Lake City, Utah, USA, 2018.

[18] Z. Gao, J. Xie, Q. Wang, and P. Li, "Global second-order pooling convolutional networks," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Long Beach, California, USA, 2019.

[19] M. Madadi, S. Escalera, X. Baró, and J. Gonzalez, "End-to-end global to local cnn learning for hand pose recovery in depth data," *arXiv preprint arXiv:1705.09606*, 2017.

[20] F. Yu, L. Zeng, D. Pan, X. Sui, and J. Tang, "Evaluating the accuracy of hand models obtained from two 3d scanning techniques," *Scientific Reports*, vol. 10, no. 1, p. 11875, 2020.

[21] J. Romero, D. Tzionas, and M. J. Black, "Embodied hands: Modeling and capturing hands and bodies together," *arXiv preprint arXiv:2201.02610*, 2022.

[22] M. Loper, N. Mahmood, J. Romero, G. Pons-Moll, and M. J. Black, "Smpl: A skinned multi-person linear model," *ACM transactions on graphics (TOG)*, vol. 34, no. 6, pp. 1–16, 2015.

[23] L. Kavan and J. Žára, "Spherical blend skinning: a real-time deformation of articulated models," in *Proceedings of the 2005 symposium on Interactive 3D graphics and games(I3D)*, Washington, District of Columbia, USA, 2005.

[24] A. J. Critchlow, "Introduction to robotics," 1985.

[25] T. Instruments, "Iwr1443 single-chip 76-to 81-ghz mmwave sensor," *IWR1443 datasheet, May*, 2017.

[26] Instruments, Texas, "Dca1000evm: Real-time data-capture adapter for radar sensing evaluation module," 2020.

[27] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C.-L. Chang, and M. Grundmann, "Mediapipe hands: On-device real-time hand tracking," *arXiv preprint arXiv:2006.10214*, 2020.

[28] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Boston, Massachusetts, USA, 2015.

[29] C. Wan, T. Probst, L. Van Gool, and A. Yao, "Crossing nets: Dual generative models with a shared latent space for hand pose estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Hawaii, USA, 2017.

[30] M. Oberweger and V. Lepetit, "Deepprior++: Improving fast and accurate 3d hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Venice, Italy, 2017.

[31] Y. Zhou, J. Lu, K. Du, X. Lin, Y. Sun, and X. Ma, "Hbe: Hand branch ensemble network for real-time 3d hand pose estimation," in *Proceedings of the European Conference on Computer Vision (ECCV)*, Munich, Germany, 2018.

[32] D. Tang, H. Jin Chang, A. Tejani, and T.-K. Kim, "Latent regression forest: Structured estimation of 3d articulated hand posture," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*,

Columbus, Ohio, USA, 2014.

[33] S. Ji, X. Zhang, Y. Zheng, and M. Li, “Construct 3d hand skeleton with commercial wifi,” in *Proceedings of the 21st ACM Conference on Embedded Networked Sensor Systems (SenSys)*, 2023, pp. 322–334.

[34] H. Kong, L. Lu, J. Yu, Y. Chen, and F. Tang, “Continuous authentication through finger gesture interaction for smart homes using wifi,” *IEEE Transactions on Mobile Computing (TMC)*, vol. 20, no. 11, pp. 3148–3162, 2020.

[35] H. Kong, L. Lu, J. Yu, Y. Chen, X. Xu, and F. Lyu, “Toward multi-user authentication using wifi signals,” *IEEE/ACM Transactions on Networking (TON)*, 2023.

[36] G. Li, Y. Ge, Y. Wang, Q. Chen, and G. Wang, “Detection of human breathing in non-line-of-sight region by using mmwave fmcw radar,” *IEEE Transactions on Instrumentation and Measurement*, vol. 71, pp. 1–11, 2022.

[37] X. Xu, J. Yu, C. Ma, Y. Ren, H. Liu, Y. Zhu, Y.-C. Chen, and F. Tang, “mmecg: Monitoring human cardiac cycle in driving environments leveraging millimeter wave,” in *Proceedings of the IEEE Conference on Computer Communications (INFOCOM)*. Virtual Conference: IEEE, 2022.

[38] M. Z. Ozturk, C. Wu, B. Wang, and K. Liu, “Radiomic: Sound sensing via mmwave signals,” *arXiv preprint arXiv:2108.03164*, 2021.

[39] C. Shi, L. Lu, J. Liu, Y. Wang, Y. Chen, and J. Yu, “mpose: Environment-and subject-agnostic 3d skeleton posture reconstruction leveraging a single mmwave device,” *Smart Health*, vol. 23, p. 100228, 2022.

[40] H. Kong, X. Xu, J. Yu, Q. Chen, C. Ma, Y. Chen, Y.-C. Chen, and L. Kong, “m3track: mmwave-based multi-user 3d posture tracking,” in *Proceedings of the 20th Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*. Portland, Oregon, USA: ACM, 2022.

[41] H. Xue, Y. Ju, C. Miao, Y. Wang, S. Wang, A. Zhang, and L. Su, “mmmesh: towards 3d real-time dynamic human mesh construction using millimeter-wave,” in *Proceedings of the 19th Annual International Conference on Mobile Systems, Applications, and Services (MobiSys)*. Virtual Conference: ACM, 2021.

[42] H. Xue, Q. Cao, Y. Ju, H. Hu, H. Wang, A. Zhang, and L. Su, “M4esh: mmwave-based 3d human mesh construction for multiple subjects,” in *Proceedings of the 20th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Boston, Massachusetts, USA, 2022.

[43] J. Xie, H. Kong, J. Yu, Y. Chen, L. Kong, Y. Zhu, and F. Tang, “mm3dface: Nonintrusive 3d facial reconstruction leveraging mmwave signals,” in *Proceedings of the 21st Annual International Conference on Mobile Systems, Applications and Services (MobiSys)*, 2023, pp. 462–474.

[44] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, “Soli: Ubiquitous gesture sensing with millimeter wave radar,” *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, pp. 1–19, 2016.

[45] J. Luo, J. Hu, Z. Chen, J. Liu, A. Khamis, S. Zhang, and T. Zheng, “Ochid-fi:occlusion-robust hand pose estimation in 3d via rf-vision,” in *Proceedings of the IEEE International Conference on Computer Vision (ICCV) Workshops*, Paris, France, 2023.