Multi-View Video-Based 3D Hand Pose Estimation

Leyla Khaleghi, Alireza Sepas-Moghaddam, Member, IEEE, Joshua Marshall, Senior Member, IEEE, Ali Etemad, Senior Member, IEEE

Abstract—Hand pose estimation (HPE) can be used for a variety of human-computer interaction applications such as gesture-based control for physical or virtual/augmented reality devices. Recent works have shown that videos or multi-view images carry rich information regarding the hand, allowing for the development of more robust HPE systems. In this paper, we present the Multi-View Video-Based 3D Hand (MuViHand) dataset, consisting of multi-view videos of the hand along with ground-truth 3D pose labels. Our dataset includes more than 402,000 synthetic hand images available in 4,560 videos. The videos have been simultaneously captured from six different angles with complex backgrounds and random levels of dynamic lighting. The data has been captured from 10 distinct animated subjects using 12 cameras in a semi-circle topology where six tracking cameras only focus on the hand and the other six fixed cameras capture the entire body. Next, we implement MuViHandNet, a neural pipeline consisting of image encoders for obtaining visual embeddings of the hand, recurrent learners to learn both temporal and angular sequential information, and graph networks with U-Net architectures to estimate the final 3D pose. We perform extensive experiments and show the challenging nature of this new dataset as well as the effectiveness of our proposed method. Ablation studies show the added value of each component in MuViHandNet, as well as the benefit of having temporal and sequential information in the dataset. We make our dataset publicly available to contribute to the field at: https://github.com/LeylaKhaleghi/MuViHand.

Index Terms—Hand pose estimation (HPE), Multi-view, Video, Dataset.

I. INTRODUCTION

Hand pose estimation (HPE) methods can play an important role in various human-computer interaction (HCI) applications, including virtual reality (VR) or augmented reality (AR) [1]–[5], gesture and sign language recognition [6]–[9], and smart vehicles [10], [11]. Despite the tremendous progress in HPE in recent years [12]–[14] due to advancements in deep learning systems, the accuracy and robustness of HPE methods still suffer from: (i) appearance variations such as the articulated shape of the hand or skin color; (ii) occlusion factors, such as wearing hand gloves or when a part of the subject’s own body occludes the hand [15], [16]; (iii) variations in the viewpoint of cameras [17]; and (iv) variations in the environment, such as complex backgrounds [18] and significantly high or low levels of lighting, which generally degrade the segmentation and estimation performance.

To date, most HPE solutions focus on estimating the pose from single RGB images [18], [20], [22], [25]–[29], [36], and consequently less emphasis has been placed on exploiting multi-view and/or video data [23], [24]. Only a few methods have recently have made use of multi-view angular information [24], whose fusion has shown to compensate for the shortage of data that a single view contains. Additionally, depth ambiguity can be significantly reduced by considering geometry relationships between viewpoints [17], [24], [37]. Hence, multi-view pose estimation methods result in a better performance when compared with single view methods. Furthermore, hand poses generally change quickly and, because the pose at any given time can be influenced by the previous poses, exploiting temporal information over video sequences could boost the performance of HPE, as shown in [23], [38].

Accordingly, the availability of large-scale video datasets captured from multiple view-points may play a key role in advancing the field of 3D HPE. Despite this, well-known hand-pose datasets such as RHD [20], GANerated [18], and FreiHAND [21] do not provide video data. Other datasets, like STB [19], Youtube Hand [22], and SeqHAND [23], lack simultaneous recording of the hands from multiple views. Consequently, development of HPE solutions that can learn jointly from multi-view information over time (videos) has been largely overlooked.

To this end, we first introduce a new HPE dataset named Multi-View Video-Based Hand (MuViHand), to enable research on multi-view video-based HPE systems. Our dataset includes more than 402,000 synthetic hand images, available in 4,560 video sequences, which have been synthetically generated. The data has been captured by using 12 cameras in a semi-circle topology, where six tracking cameras (Cameras 1-6) focus only on the hand while the other six fixed cameras (Cameras 7-12) capture the whole body.

Fig. 1. The multi-view camera topology that has been used for recording the MuViHand dataset is presented in this figure. The videos have been recorded using 12 cameras situated in two concentric circles, where six tracking cameras (Cameras 1-6) focus only on the hand while the other six fixed cameras (Cameras 7-12) capture the whole body.

Leyla Khaleghi, Alireza Sepas-Moghaddam, Joshua Marshall and Ali Etemad are with the Department of Electrical & Computer Engineering and the Ingenuity Labs Research Institute, Queen’s University, Kingston, ON, Canada. Contact information: 19lk12@queensu.ca.
Motivated by the availability of the MuViHand dataset, we propose a graph-based HPE method, called MuViHandNet, to jointly learn from both temporal and angular information. Successive to extracting spatial embeddings from each frame using an encoder, our model uses a pair of temporal and angular learners to learn effective spatio-temporal and spatio-angular representations. These representations are then concatenated and jointly learned for estimating 2D hand coordinates. Rather than estimating the 3D hand pose coordinates directly from the embeddings, MuViHandNet initially estimates the 2D hand coordinates prior to transforming them into 3D camera hand coordinates by implicitly estimating depth information. Given the graph-based structure of the hand skeleton, we use a graph convolutional network (GCN) to model the hand joint constraints and connections, and ultimately 3D HPE.

Our contributions in this paper are summarized as follows:

- We present the MuViHand dataset, which is the first and largest synthetic multi-view video hand pose dataset with two different types of cameras in which half capture the full body while the other half track the target hand. Our dataset includes more than 402,000 frames with complex backgrounds, occlusions, and dynamic sources of lighting. This dataset is publicly available to the research community.
- We propose MuViHandNet, the first method for multi-view video-based 3D HPE that achieves a robust performance on our dataset by considering both the temporal and angular relationships between hand-image embeddings. Our model consists of an image encoder, temporal learners, angular learners, and a graph U-Net.
- We demonstrate the benefits of this new multi-view video-based dataset and 3D HPE model, and present a comprehensive benchmarking study against other state-of-the-art HPE methods as well as ablated baselines. The experiments show that MuViHandNet achieves superior results, with considerable performance gains of up to 55% when compared to other state-of-the-art methods.

The remainder of this paper is organized as follows. Section 2 provides a review of recent advances in HPE datasets and methods. The newly captured hand dataset is described in Section 3 and the proposed method is presented in Section 4. Section 5 presents an extensive performance evaluation for the proposed and state-of-the-art methods using varied and challenging HPE tasks. Also, the limitation of our method with some remarks for future research directions are discussed. Finally, Section 6 concludes our paper.

### Table I

**Comparison between the MuViHand dataset and other public hand pose datasets.**

| Dataset          | Year | RGB/Depth | Real/Synth. | # Frames | Static/Seq. | Multi-View | Full/Zoomed | Fixed/Tracking |
|------------------|------|-----------|-------------|----------|-------------|------------|-------------|----------------|
| STB [19]         | 2016 | RGB+Depth | Real        | 18K      | Seq.        | No         | Full        | Fixed          |
| RHD [20]         | 2017 | RGB+Depth | Real        | 43.7K    | Static      | No         | Full        | Fixed          |
| GANerated [18]   | 2018 | RGB       | Synth       | 330K     | Static      | No         | Zoomed      | Fixed          |
| FreiHAND [21]    | 2019 | RGB       | Real        | 134K     | Static      | No         | Full        | Fixed          |
| YoutubeHand [23] | 2020 | RGB       | Real        | 48.65K   | Seq.        | No         | Full        | Fixed          |
| SeqHAND [23]     | 2020 | RGB       | Synth       | 410K     | Seq.        | No         | Zoomed      | Fixed          |
| MVHM [24]        | 2020 | RGB+Depth | Synth       | 320K     | Static      | Yes        | Zoomed      | Fixed          |
| MuViHand (ours)  | 2021 | RGB       | Synth       | 402K     | Seq.        | Yes        | Both        | Both           |

### II. RELATED WORK

#### A. Existing Hand Pose Datasets

Several publicly available hand pose datasets have been previously developed for different applications and scenarios. Table I provides an overview of the main characteristics of the well-known hand pose datasets, including the types of cameras used, the types of data (real vs. synthetic and static vs. sequential), the number of frames, the type of view-related acquisition (multi-view or not), whether the images only capture the hand or whether the entire body has been captured (which is more challenging), and whether the employed cameras are fixed or track the hand. In order to show the chronological evolution of these hand pose datasets in the table, they have been sorted by the order of release date. For comparison, we also included the characteristics of our dataset, MuViHand, which is proposed in this paper in Table I. We note that MuViHand is among those datasets with the highest number of frames and is both multi-view and sequential, which no other dataset provides.

Moreover, MuViHand contains the entire body, which is more representative of real-life scenarios while also making it more challenging for pose estimators because other body parts with similar skin characteristics might confuse candidate algorithms. Lastly, our dataset is the only dataset that contains both fixed and tracking cameras. In what follows we provide a brief description of the datasets mentioned in Table I.

**Stereo Hand Pose Tracking Benchmark (STB) [19]** is one of the most popular real-world single-view datasets for 3D HPE. In this dataset, only one subject perform random and number counting poses with six different backgrounds. These 12 sequences, each with 1500 frames, were annotated manually.

**Rendered Hand Pose Dataset (RHD) [20]** is a synthetic hand pose dataset, including 20 different characters who perform 39 actions. The dataset includes 43,700 images that were captured with multiple random backgrounds from different angles (not recorded simultaneously, hence not multi-view). Given that the dataset uses synthetic images, the keypoints were annotated automatically.

**GANerated [18]** was first generated synthetically, and subsequently used by a CycleGan network and translated to real images. GANerated includes more than 260,000 frames of hand poses with different skin tones.

**FreiHAND [21]** is a large-scale real-world hand pose dataset collected from 32 subjects. The dataset includes 134,000 hand images with various poses. The annotations were partially...
carried out manually and partially by an iterative semi-automated approach.

**YouTube Hand** [22] was generated from 109 YouTube videos. The dataset includes 48,650 hand images. The process of annotation was done by fitting a parametric hand model, named MANO [39], to a publicly available 2D prediction network called OpenPose [17].

**SeqHAND** [23] is a large synthetic sequential hand pose dataset with 410,000 images. The MANO [39] hand model was used with the ground-truth annotations of a sequential depth-based hand pose dataset called BigHand 2.2M, to render the hand pose sequences for the SeqHAND dataset.

**Multi-View Hand Mesh (MVHM)** [24] is a synthetic multi-view dataset with 320,000 images. Similar to the previous dataset, MVHM used ground truth annotations from a different dataset, in this case the NYU [40], along with a hand model called TurboSquid to render the hand images from eight views.

### B. Existing HPE Methods

While in this paper we focus on **multi-view** and **video-based** HPE, in this section we review related literature that has performed 3D HPE from single RGB images given the lack of multi-view and video-based datasets [18], [20], [22], [23]–[29], [29]–[35]. We then review the few works that have taken multi-view or sequential approaches [17], [23], [24].

1) **Single-view HPE:** In [20], the problem of 3D HPE is broken up into two steps. First, a CNN extracts image features and directly estimates the 2D heat maps, from which the normalized 3D hand coordinate is subsequently measured. Similar to [20], in [23], 3D HPE is performed in two steps. However, the 3D hand pose is estimated from a 2.5D heat map instead of a 2D heat map by estimating the depth data as well.

Some HPE methods [26], [27] boost the performance of RGB-based HPE with the help of privileged learning of depth information. In [26], a depth regularizer network is applied after the 3D HPE network during training to learn to generate the corresponding depth map from a 3D hand pose. However, during testing the RGB images go through only the 3D HPE network. Similarly, in [31] the network learns to generate the corresponding depth map from the 3D hand shape instead of the pose. In [27], an RGB-based HPE and a depth-based HPE network are independently trained. The depth-based network is then frozen and the RGB-based network’s training is resumed with paired RGB and depth images by sharing the information between the middle CNN layers of these two networks.

Several HPE methods [31], [33] rely on a predefined 3D hand model for estimating 3D hand poses. In [31], with the help of solving an optimization problem, a hand model with 27 parameters is fit to 2D joints locations estimated by OpenPose [17]. In [33] a hand image is passed through a ResNet-18 to generate the MANO hand model parameters for estimating the 3D hand pose. Similarly, in [32], a hand image and its 2D joint heat maps (obtained from OpenPose) are passed through a ResNet-50 for generating the input parameters to the MANO hand model, from which the 3D hand pose is measured.

2) **Multi-View HPE:** In the only multi-view HPE method found in the literature, the proposed solution [24] receives a number of hand images captured from different views as inputs, and passes them to individual single-view HPE networks to predict the 3D camera coordinates for each view independently. Then these 3D camera coordinates are concatenated and pass through a graph-based neural network to predict the 3D world coordinates for each pose.

3) **Temporal HPE:** In [38] 3D hand pose is estimated from a temporal sequence of 2D hand joints obtained from a hand pose estimator such as OpenPose. The method creates a GCN that considers the temporal relationships by incorporating additional edges between the same joints in consecutive frames. In [23], a recurrent layer is included after the encoder of a single-view hand pose estimator (proposed in [32]) to exploit the temporal relationships and generate the MANO hand model parameters for each frame.

Based on our above-described literature review, we observe that there are no HPE methods that simultaneously consider both spatio-temporal and spatio-angular relationships. Thus, this paper introduces the first method for multi-view video-based HPE, called MuViHandNet.

### III. Dataset

Our proposed dataset, MuViHand, is a synthetic multi-view video-based hand pose dataset, create using the freely available MIXAMO [2] a web-based service for 3D character animation synthesis. MIXAMO has also been used for creating the Rendered Hand Pose Dataset [20]. In our work, we select 10 characters with various appearances (shown in Figure 2) who perform 19 various full-body actions in sitting, standing, and walking posture, that have been captured from motion actors. We also randomly choose several street and in-vehicle images as backgrounds from the Pxfuel [3] website. Finally, Blender [4], an open-source 3D graphic computer software, is used to render

1Available online at https://www.turbosquid.com

2Available online at https://www.mixamo.com

3Available online at https://www.pxfuel.com

4Available online at https://www.blender.org
the videos from six different views. In total, 4,560 videos with 402,000 frames are rendered, a few samples of which are shown in Figure 3. This dataset is publicly available at https://github.com/LeylaKhaleghi/MuViHand.

A. Lighting Conditions

We use two dynamic lighting sources in creating the videos, namely (i) sun and (ii) point light. In Figure 4 several frames with the two sources of light are shown. In the videos with sun as the light source, the location of the sun is randomly picked between two concentric semi-spheres around the subject. Moreover, the sun source experiences very small motions towards the left or right during each video to create a slightly moving shadow. The point light sources are similarly located between two concentric spheres with much smaller radii compared to those of the sun light. During each video, the point light source moves closer to the user, creating a moving shadow similar to that experienced as a result of a moving vehicle at night.

B. Camera Topology

We use 12 cameras when generating the dataset, six of which are fixed in a semi-circle topology around the user, while the other six track the user, three of which focus on the right hand while the other three focus on the left hand. This topology is shown in Figure 1. Each set of cameras (fixed and tracking) have been positioned in six evenly spaced angles from $15^\circ$ to $165^\circ$ ($15^\circ$, $45^\circ$, $75^\circ$, ..., $165^\circ$) on a semicircle topology around the subject. Figure 5 presents a sample hand image and corresponding ground-truth poses as observed by the six fixed cameras.

C. Annotation

We provide 2D and 3D locations for 21 hand joints, including one joint for the wrist and four joints per each finger (Fingertip, DIP, PIP, MCP), similar to [20]. See Figure 5(a) for details about the joints used in this dataset. Moreover, Figure 5(b) shows sample ground-truth poses provided for different views. For each 3D hand pose, we provide the 3D world coordinates as well as the 3D camera coordinates. Finally, the intrinsic matrices for the cameras are also provided, where a single intrinsic matrix describes each static camera, while for the tracking cameras, a different matrix is provided for each frame.

D. Pose and Activity Distribution

In order to illustrate how the pose information is distributed in our dataset, we visualize the 3D root-relative pose coordinates ($1 \times 63$ vector) using t-Distributed Stochastic Neighbor Embedding (t-SNE) in Figure 6 similar to [23], [42], [43]. We observe that the pose space captured by the dataset is quite varied and non-skewed. Moreover, we use different colours to represent each of the 19 activities used in the dataset. From the figure, we observe that the distribution of activities is also spread out in the pose space.
IV. METHOD

Our work builds upon the idea that learning temporal [17], [24], [37] or angular [23], [38] information is beneficial for HPE. We propose MuViHandNet, ξ, a deep neural network for predicting the 3D hand camera coordinates P from the corresponding multi-view videos Φ, such that

\[ P = \xi(\Phi), \tag{1} \]

where \( \Phi = \{\phi_v^t\} \) is a frame captured at time \( t \) and from view \( v \). Each hand frame is described by an RGB image \( \phi_v^t \in \mathbb{R}^{H \times W} \), where \( H \) is the height of the image and \( W \) is the image’s width. Moreover, the 3D hand camera coordinates at time \( t \) and view \( v \), \( P = \{p_v^t\} \in \mathbb{R}^{J \times 3} \), are described by a hand skeleton with \( J \) joints.

A. Model Overview

Figure 7 depicts the overview of our proposed network architecture. Similar to many state-of-the-art HPE methods [1], [21], [24], [32], [33], [42], we use a ResNet as the encoder to extract features from each frame following by recurrent neural networks (RNNs) to learn the temporal relationships among the embeddings of the video frames. Furthermore, as shown in prior literature (in other domains) that RNNs can also effectively learn angular relationships [45]–[48], we incorporate an additional RNN in our model. The outputs of the two sets of RNNs (temporal and angular) are combined and used to estimate 2D joint positions, which are then fed to a Graph U-Net to provide the final 3D hand pose. Two-stage estimation of 3D hand pose (2D followed by 3D) has been widely used in prior works and shown advantages over direct 3D HPE [18], [20], [25], [49].

B. Image Encoder

We use a ResNet10 [50] pre-trained on ImageNet [51], as the image encoder. This encoder takes each hand frame \( \phi_v^t \) as its input and generates an embedding \( F_v^t \) at time \( t \) and from view \( v \), where \( t = 1, 2, \ldots, T \), \( T \) is the video length, \( v = 1, 2, \ldots, V \), and \( V \) is the number of the views.

C. Temporal and Angular Learning

Long short-term memory (LSTM) networks [52] are a popular type of RNNs used in a variety of different applications [53]–[55]. Due to their effective performance in many domains, we use LSTMs for both spatio-temporal and spatio-angular learning. Each LSTM unit consists of three different gates, namely input \( r \), forget \( f \), and output \( o \) gates as well as cell \( c \) and hidden \( h \) memories. Here we introduce the LSTM equations for the \( k^{th} \) instance of a sequence, where ‘instance’ is defined as a frame in a video or a particular viewpoint in a multi-view sequence.

Initially, the input gate for the \( k^{th} \) instance of a sequence is computed according to

\[ r_k = \sigma(w_{rx}x_k + w_{rh}h_{k-1} + b_r), \tag{2} \]

where \( x_k \) is the input vector to the LSTM unit, \( h_{k-1} \) is the previous hidden state, \( w_{rx} \) and \( w_{rh} \) are the input gate weights, and \( b_r \) is the input gate bias. \( \sigma \) denotes the sigmoid activation. In order to control how the cell forgets information from its state, the forget gate \( f_k \) is computed according to

\[ f_k = \sigma(w_{fx}x_k + w_{fh}h_{k-1} + b_f), \tag{3} \]

where \( w_{fx} \) and \( w_{fh} \) are the forget gate weights, and \( b_f \) is the forget gate bias. The output is then computed by using

\[ o_k = \sigma(w_{ox}x_k + w_{oh}h_{k-1} + b_o), \tag{4} \]

where \( w_{ox} \) and \( w_{oh} \) are the output gate weights, and \( b_o \) is the output gate bias.

The cell state that controls remembering values over the sequence (time or view) is updated according to

\[ \tilde{c}_k = \tanh(w_{cx}x_k + w_{ch}h_{k-1} + b_c), \tag{5} \]

and

\[ c_k = i_k \odot \tilde{c}_k + f_k \odot c_{k-1}, \tag{6} \]

where \( w_{cx} \) and \( w_{ch} \) are the cell weights, and \( b_c \) is the cell bias. Finally, the hidden state \( h_k \) is computed based on learning jointly the cell state \( c_k \), and the output gates \( o_k \) according to

\[ h_k = o_k \odot \tanh(c_k). \tag{7} \]

As mentioned earlier, we use two sets of LSTM networks to learn the temporal and angular relationships separately, which we name the temporal learner and angular learner, respectively. We denote these networks LSTM_t and LSTM_v. Accordingly,

\[ [X_v^1, X_v^2, \ldots, X_v^T] = LSTM_t([F_v^1, F_v^2, \ldots, F_v^T]), \tag{8} \]

and

\[ [Y_v^1, Y_v^2, \ldots, Y_v^V] = LSTM_v([F_v^1, F_v^2, \ldots, F_v^T]), \tag{9} \]
where $X^t_v$ and $Y^t_v$ are the outputs of the temporal and angular learner cells at time $t$ and view $v$.

Next, for the temporal and angular information to jointly contribute to the final HPE, the two sets of LSTM outputs $X^t_v$ and $Y^t_v$ are concatenated by using

$$Z^t_v = X^t_v \oplus Y^t_v,$$  \hspace{1cm} (10)

where $\oplus$ denotes the concatenation operation and $Z^t_v$ is the joint feature set. In total, a set of features with $V \times T$ vectors are generated at this stage. Each $Z^t_v$ of the $V \times T$ vectors are then fed to 2 fully connected layers to produce the 2D coordinates ($\nu^t_v$) for each frame.

### D. Graph U-Net

Because the goal of this work is to perform 3D HPE, the 2D coordinates estimated by the FC layers following the temporal and angular learners need to be converted to 3D. In this context, it has been extensively shown in the literature [24], [38], [41], [49] that the graph-based structure of the hand skeleton lends itself well to GCN-style networks [56]. Consequently, we employ a graph U-Net structure [57] which is illustrated in Figure 8. This module has an encoder-decoder structure [58] with a number of skip connections that concatenate the encoders and decoder features, along with a number of GCN layers (depicted in Figure 8 as Graph conv layers).

For a GCN layer, we define a graph $G = (N, A)$, where $N$ is the number of nodes and $A \in \mathbb{R}^{N \times N}$ as the adjacency matrix. The values of the adjacency matrix are defined based on the relationship between nodes; if two nodes are connected, the value is equal to 1, otherwise equal to 0. In HPE, one often applies the kinematic structure of the hand skeleton as the adjacency matrix [24], [38], [41]. In this paper, however, we learn the adjacency matrix to allow for more advanced connections to be dynamically discovered automatically. This approach was proposed by [49], and our results (presented in the next section) demonstrate that this approach in fact boosts performance when compared to random as well as pre-defined adjacency matrices.

The output of the layer with $F$ input features and trainable weight matrix $W \in \mathbb{R}^{N \times L}$, where $L$ is the output feature size, is computed according to

$$Y = \sigma(\tilde{A}XW),$$  \hspace{1cm} (11)

where $X \in \mathbb{R}^{N \times F}$ is the GCN layer input and $\tilde{A}$ is the normalized adjacency matrix of the graph [56]. $\tilde{A}$ is measured as

$$\tilde{A} = D^{-\frac{1}{2}} \hat{A} D^{-\frac{1}{2}},$$  \hspace{1cm} (12)

where

$$\hat{A} = A + I,$$  \hspace{1cm} (13)

$D$ is the diagonal node degree matrix, and $I$ is the identity matrix. Accordingly, the graph U-Net module $G^t_v$ transforms
each 2D coordinate $w^t_v$ to the 3D camera coordinate $p^t_v$ at time $t$ and view $v$ such that
\[ p^t_v = G^t_v(w^t_v), \quad p \in \mathbb{R}^{21 \times 3}, \quad w \in \mathbb{R}^{21 \times 2}. \] (14)

E. Training and Implementation Details

We employ a multi-stage training strategy for MuViHandNet. First (Stage 1) we aim to train the pipeline irrespective of the temporal and sequential learner components. To this end, we temporally replace the LSTM networks with a fully connected layer and train the entire pipeline, essentially re-training the image encoder (which is ResNet-10 pre-trained by ImageNet [51]) and training the graph U-Net. Next (Stage 2), we replace the temporary FC layer with the original LSTMs and retrain the entire network while the image encoder is kept frozen, in essence training the temporal and sequential learners and re-training the $T \times V$ instances of the graph U-Net.

In stage 1, we train the image encoder and the graph U-Net for 500 epochs with Adam optimizer. An initial learning rate of 0.001 is used and multiplied by 0.1 every 100 epochs. The utilized loss function is
\[ L = \alpha L_{2D} + L_{3D}. \] (15)

Here, the $L_{2D}$ is calculated according to
\[ L_{2D} = ||\hat{w} - w||_2, \] (16)

where $\hat{w}$ and $w$ are the predicted and ground truth 2D coordinates respectively. Also, the $L_{3D}$ is measured according to
\[ L_{3D} = ||\hat{p} - p||_2, \] (17)

where $\hat{p}$ and $p$ are the predicted and ground truth 3D coordinates, respectively.

In stage 2, when the temporal and angular learners are added to the pipeline, training is performed for 400 epochs. Here, the loss function is
\[ L = \frac{1}{V \times T} \sum_t \sum_v ||\hat{p}^t_v - p^t_v||_2, \] (18)

where $\hat{p}^t_v$ and $p^t_v$ are respectively the predicted and ground truth 3D coordinates at time $t$ and view $v$.

\begin{table}[h]
\centering
\caption{Training hyper-parameters used in stages 1 and 2.}
\begin{tabular}{|l|l|l|l|}
\hline
\textbf{Sub-Net} & \textbf{Parameter} & \textbf{Stage 1} & \textbf{Stage 2} \\
\hline
Encoder & Architecture & ResNet-10 & ResNet-10 \\
& Pretrained & ImageNet & ImageNet \\
& # of Inputs & 1 & 1 \\
& Embodiment Layer & Avg. Pooling & Avg. Pooling \\
& Feature Size & 512 & 512 \\
\hline
Multi-view LSTM & # of Inputs & - & 3 \\
& # of hidden layers & - & 2 \\
& Hidden Size & - & 128 \\
\hline
Temporal LSTM & # of Inputs & - & 5 \\
& # of hidden layers & - & 2 \\
& Hidden Size & - & 128 \\
\hline
FC + ReLU & Dimensionality & 256 & - \\
FC + ReLU & Dimensionality & 128 & 128 \\
FC + ReLU & Dimensionality & 21 & 21 \\
\hline
Drop out & Size & - & 0.25 \\
& Output size & 21 & 21 \\
\hline
Full Network & Batch Size & 64 & 8 \\
& Loss Function & 0.01L_{2D} + L_{3D} & $L_{3D}$ \\
& Optimizer & Adam & Adam \\
& Learning rate & 0.001 & 0.006 \\
& Weight Decay & 0.1 & 0.07 \\
& Step Decay & 100 & 100 \\
& # of Epochs & 500 & 400 \\
\hline
\end{tabular}
\end{table}

V. Experiments and Results

In this section, we describe our experiments and report results. We also report on the outcome of ablation experiments and investigate the effects of various components of our network on the overall HPE performance.

A. Test Protocol and Evaluation

To rigorously evaluate the result of our proposed method, two evaluation protocols have been tested on the MuViHand dataset, (i) cross-subject, in which seven subjects are used for training the network (subjects $\{3, \ldots, 9\}$) and three other subjects (subjects $\{1, 2, 10\}$) with different variations of skin tones, appearance, and gender are set aside for testing; (ii) cross-activity in which two random activities (activities
TABLE III

| Test          | Method       | Encoder      | Loss Function                                                                 | Root Pos | ↓ Avg. EPE | ↓ Avg. median EPE | ↑ AUC |
|---------------|--------------|--------------|-------------------------------------------------------------------------------|---------|------------|-------------------|-------|
| cross-subject | Boukhayma et al. [32] | ResNet-50    | $L_{mask}, L_{2D}, L_{3D}, L_\theta$                                       | Yes     | 135.899    | 138.276           | 0.012 |
|               | Boukhayma et al. [32] | ResNet-50    | $L_{3D}$                                                                    | Yes     | 48.840     | 40.837            | 0.280 |
|               | Hasson et al. [33]   | ResNet-18    | $L_{2D}, L_\theta$                                                          | Yes     | 62.674     | 65.524            | 0.14  |
|               | Hasson et al. [33]   | ResNet-18    | $L_{3D}$                                                                    | Yes     | 28.915     | 24.764            | 0.574 |
|               | Doosti et al. [39]   | ResNet-10    | $L_{init2D}, L_{2D}, L_{3D}$                                                | No      | 18.895     | 16.635            | 0.634 |
|               | MuViHandNet (GRU, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 13.450     | 11.646            | 0.739 |
|               | MuViHandNet (LSTM, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 10.493     | 8.715             | 0.798 |
|               | MuViHandNet (LSTM, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 10.092     | 8.380             | 0.807 |
|               | MuViHandNet (AutoEnc) | ResNet-10    | $L_{2D}, L_{3D}$                                                            | No      | 27.461     | 24.828            | 0.480 |
|               | MuViHandNet (GCN)    | ResNet-10    | $L_{2D}, L_{3D}$                                                            | No      | 13.175     | 11.263            | 0.745 |
|               | MuViHandNet (proposed) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 8.881      | 7.351             | 0.831 |
| cross-activity| Boukhayma et al. [32] | ResNet-50    | $L_{3D}$                                                                    | Yes     | 42.799     | 40.238            | 0.287 |
|               | Hasson et al. [33]   | ResNet-18    | $L_{3D}$                                                                    | Yes     | 66.851     | 68.457            | 0.152 |
|               | Doosti et al. [39]   | ResNet-10    | $L_{init2D}, L_{2D}, L_{3D}$                                                | No      | 46.745     | 45.086            | 0.217 |
|               | MuViHandNet (GRU, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 22.065     | 20.726            | 0.575 |
|               | MuViHandNet (LSTM, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 21.266     | 19.186            | 0.589 |
|               | MuViHandNet (LSTM, GRU) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 23.222     | 21.241            | 0.553 |
|               | MuViHandNet (AutoEnc) | ResNet-10    | $L_{2D}, L_{3D}$                                                            | No      | 31.212     | 30.812            | 0.423 |
|               | MuViHandNet (GCN)    | ResNet-10    | $L_{2D}, L_{3D}$                                                            | No      | 29.506     | 27.957            | 0.446 |
|               | MuViHandNet (proposed) | ResNet-10   | $L_{2D}, L_{3D}$                                                            | No      | 20.375     | 17.819            | 0.608 |

Fig. 9. Comparison of PCK curves for our method in comparison with state-of-the-art solutions [32], [33], [49]. Two different testing protocols have been used: cross-subject (top) and cross-activity (bottom).

which the right hand images are at the center of the frames and they have been resized to 224 × 224 pixels to fit the input size of the image encoder.

Similar to [20] for evaluating our method, three metrics are used. These include (i) the percentage of correct key points (PCK) with a threshold between 0-50 mm; (ii) the area under the curve (AUC) of the PCK; (iii) the mean and median endpoint error (EPE).

B. Benchmarking Methods

We compare our results against three state-of-the-art 3D HPE methods [32], [33], [49]. These works have been selected as benchmarks for the following reasons: (1) they obtained very strong results for 3D HPE; (2) their implementations are publicly available, which is essential given that the benchmarks need to be re-trained on our newly proposed dataset; (3) similar to our proposed method, they do not use any additional modalities such as depth towards 3D HPE. Nevertheless, because these methods have been originally optimized for other datasets and not MuViHand, we tune the parameters of the three state-of-the-art methods to obtain the best possible performance to allow for a fair comparison.

In addition to state-of-the-art benchmarks, we create and evaluate several variations of the proposed MuViHandNet. First, we substitute the LSTM temporal and angular learners with Gated recurrent unit (GRU) networks. We refer to this variation of the model as MuViHandNet(GRU). Next, we explore using a combination of LSTM and GRU for temporal and angular learners, and vice versa. These are referred to as MuViHandNet(LSTM, GRU) and MuViHandNet(GRU, LSTM). We then create another benchmark variant by swapping the GCN layers of the graph U-Net structure with fully connected layers, essentially creating a stacked autoencoder. We refer to this variant as MuViHandNet(AutoEnc). Lastly, we modify the graph U-Net to no longer have a U-Net architecture by using a three GCN layers instead. This variant is referred to as MuViHandNet(GCN). To differentiate the original model as proposed in Section IV, we
Fig. 10. Samples of estimated hand poses using our method in comparison with other works [32], [33], [49] are presented.

TABLE IV
IMPACT OF WINDOW SIZE WHICH IS USED IN THE LSTM$_t$ OF THE MuViHandNet ON THE EPE.

| Test       | Win. Size | 3   | 5   | 7   | 9   | 11  |
|------------|-----------|-----|-----|-----|-----|-----|
| cross-subject | 11.197    | 8.881 | 13.432 | 12.149 | 14.710 |
| cross-activity  | 21.605    | **20.375** | 24.621 | 22.991 | 27.290 |

TABLE V
THE IMPACT OF DIFFERENT TYPES OF ADJACENCY MATRICES IN OUR GRAPH U-NET.

| Test       | Adj. Mat. | Rand 1 | Rand 2 | Rand 3 | Hand Skel. | Learned |
|------------|-----------|--------|--------|--------|------------|---------|
| cross-subject | 20.951    | 16.951 | 43.225 | 22.826 | 25.829     | 23.565  |
| cross-activity  | 25.829    | 21.354 | 20.375 | 22.826 | 13.432     | 23.565  |

use the term MuViHandNet(proposed) in the tables in Section IV.

C. Performance and Discussion

The performance of MuViHandNet in comparison with the state-of-the-art methods [32], [33], [49], along with the variations discussed in Section V-B for the two test protocols described in Section V-A, are presented in Table III. MuViHandNet outperforms the state-of-the-art methods (with different loss functions) in both testing protocols by a considerable margin. It is also evident that the proposed method including the graph U-Net and LSTM learners outperforms the benchmarking variants, which we described in Section V-B. When comparing the performance of MuViHandNet in the cross-subject scheme with cross-activity, the cross-activity protocol is far more challenging. This is in line with prior works that have shown that HPE methods often fail on the unseen poses [59]. Moreover, most of the prior works [32], [33] require the root pose as an input to these models, while [49], along with our method, operate without such input.

Figure 9 presents the PCK curves for various thresholds (0-50) for the state-of-the-art methods for the two test protocols, as well as the proposed MuViHandNet. To subjectively evaluate the performance of our method, we highlight two challenging images along with the detected poses by our method along with the state-of-the-art benchmarks [32], [33], [49]. Note that existing methods often perform poorly when dealing with such challenging scenarios where lighting conditions are relatively poor or the pose contains hidden hand parts and fingers. This points to (a) the effectiveness of our proposed method, and (b) the challenging nature and thus contribution of our proposed

Fig. 11. The network architectures for ablated variations of MuViHandNet: (a) Baseline 1, (b) Baseline 2, and (c) Baseline 3.
To evaluate the impact of each component of MuViHandNet on our results, we create three ablated variations by systematically removing the main components of the model as follows. First, we remove the temporal learner module, LSTM_t. The ablated model is depicted in Figure [11]a). In this model, the encoder accepts frames from three different views and generates a 512 dimensional image embedding $F_v, v = 1, 2, 3$ for each view. The [$F_1, F_2, F_3] \in \mathbb{R}^{3 \times 512}$ is then fed to the angular learner LSTM_v to generate a feature vector $Y_v, v = 1, 2, 3$, where each $Y_v$ is fed to two FC layers for generating the 2D coordinates followed by the graph U-Net to transform the 2D joint locations to 3D camera coordinates for each hand frame. Next, we remove the angular learner LSTM_v from the structure of MuViHandNet, as shown in Figure [11]b). This method takes five consecutive frames and, similar to the previous ablation experiment, a 512 dimensional image embedding $F_t, t = 1, 2, \ldots, 5$ is generated for each frame to then feed the rest of the network. Lastly, both sequential learners LSTM_t and LSTM_v are removed, meaning that only a single hand frame is passed through the encoder followed by 3 FC layers and a graph U-Net module to estimate the 3D hand pose, as shown in Figure [11]c).

The results of this experiment are presented in Table VI for both testing protocols (cross-subject and cross-activity). The EPE is broken out across each joint and finger. These results demonstrate the advantages of each of the two angular and temporal learners in our pipeline. Here, when the angular learner is removed, the EPE increases by around 1.3 mm and 1.2 mm for cross-subject and cross-activity schemes, respectively. Next, when the angular learner is removed, the EPE increases by approximately 3 mm and 3.5 mm in the cross-subject and cross-activity schemes. This illustrates that the impact of learning angular information is higher than that of temporal information for 3D HPE. Finally, when both are removed, our method suffers from an increase in EPE of around 6 mm and 24 mm for the two evaluation schemes, respectively. This also demonstrates the added value of having both temporal and angular information in our dataset.

By breaking down the EPE across each joint, we remark that removal of both temporal and angular learners significantly increases the EPE for the wrist, pointing that the inclusion of the sequential learners in our model considerably impacts the ability to locate the position of the hand. Finally, when we break down the ablation experiments for different fingers, we observe that the removal of both sequential learners impacts

### Table VI

| Test          | Method | LSTM_t | LSTM_v | ↓ EPE | ↑ Wrist | ↓ MCP | ↓ PIP | ↓ DIP | ↓ TIP | ↓ Thumb | ↓ Index | ↓ Middle | ↓ Ring | ↓ Pinkie |
|---------------|--------|--------|--------|-------|---------|-------|-------|-------|-------|---------|---------|----------|--------|----------|
| Dataset       |        |        |        |       |         |       |       |       |       |         |         |          |        |          |
| Baseline 1    | ✔      | ✗      | ✔      | 10.034| 0.808   | 43.316| 8.389 | 8.413 | 9.644 | 12.929  | 10.829  | 10.698   | 9.525  | 8.530    | 9.910  |
| Baseline 2    | ✔      | ✗      | ✔      | 11.823| 0.766   | 53.096| 9.329 | 9.622 | 11.617| 15.924  | 13.274  | 12.226   | 10.968 | 9.966    | 11.682 |
| Baseline 3    | ✔      | ✗      | ✔      | 14.529| 0.723   | 61.555| 9.852 | 11.818| 15.030| 19.928  | 15.388  | 14.728   | 13.281 | 12.809   | 14.572 |
| Full Model    | ✔      | ✔      | ✔      | 8.881 | 0.831   | 36.882| 7.277 | 7.733 | 8.583 | 11.086  | 9.220   | 8.847    | 7.773  | 7.820    | 9.687  |
| Dataset       |        |        |        |       |         |       |       |       |       |         |         |          |        |          |
| Baseline 1    | ✗      | ✔      | ✔      | 21.463| 0.592   | 107.105|14.327|15.300|20.837|33.241   | 26.776  | 21.185   | 17.660 | 18.451   | 20.559 |
| Baseline 2    | ✔      | ✔      | ✔      | 23.631| 0.557   | 120.143|16.129|19.197|27.238|40.598   | 26.776  | 21.185   | 17.660 | 18.451   | 20.559 |
| Baseline 3    | ✗      | ✔      | ✔      | 44.021| 0.246   | 177.371|29.700|39.055|50.392|63.873   | 44.342  | 48.227   | 43.403 | 48.749   | 47.754 |
| Full Model    | ✔      | ✔      | ✔      | 20.375| 0.608   | 100.462|13.605|14.586|20.188|31.668   | 25.115  | 21.135   | 16.636 | 17.332   | 19.640 |

![Fig. 12. Comparison of PCK curves for 3 ablated variations of MuViHandNet along with the full model. cross-subject (top) and cross-activity (bottom).](image-url)
the pinkie finger more negatively than the others, which could be due to the higher likelihood for this finger to be obstructed by other fingers. The table presents the AUC, which also shows similar trends for all of the above-mentioned experiments.

Lastly, Figure 12 shows the PCK curves for our ablation study, where consistent behaviours for different threshold values are observed. Further, we visualize the estimated hand poses for several images with no self-occlusion using MuViHandNet and its ablated variations in Figure 13. It is observed that MuViHandNet performs better under poor illumination (row 1), challenging background with the same color tone as the hand (row 2), and different viewing angles (rows 3 through 5). Additionally, we present the performance of our method on samples with severe self-occlusions in Figure 14. Here, we observe that in these difficult scenarios, the performance drops considerably for the three baseline models in comparison to MuViHandNet. This is because our proposed pipeline can effectively learn information from additional views (angular leaner) or frames (temporal learner) to obtain a better sense of the occluded joints.

E. Limitations and Future Work

The MuViHand dataset includes full-body images of subjects, and thus often depicts the hands in front of other body parts, which generally contain the same skin tone. As expected, such scenarios posed challenges for MuViHandNet (see Figure 15) and while our model still performed better than other existing methods, the results could be further improved. Adding further modalities such as depth can allow for the model to focus on the hand in the foreground to overcome such issues, which we can explore in future work.

Moreover, our dataset was developed by synthetic images. For future work, to allow for more in-the-wild applicability, the use of generative adversarial networks might be explored to add realism to the synthetic images with the help of a real hand images. Lastly, weakly supervised techniques could be used to combine our dataset (which contains accurate ground-truths)
with real-world datasets (which often do not contain accurate ground-truths) for HPE applications.

VI. CONCLUSION

In this research we proposed a novel multi-view video-based hand pose datasets consisting of synthetic images and ground-truth 2D/3D pose values. Our dataset, MuViHand includes more than 402,000 images in 4,560 videos, and has been captured from six different views with 12 cameras in two concentric circles (one fixed and the to track the hand). The dataset, which we make public, is the first to include synthetic videos in a multi-view setting and provides a rich resource for performing 3D HPE in challenging scenarios. Next, we proposed a new model, MuViHandNet for detection HPE on our dataset. The model consists of different components for encoding images, learning temporal and angular relationships, and estimating the 3D poses. We performed rigorous experiments to evaluate the performance of our model, including comparisons with other methods and a number of ablated baselines. Our experiments demonstrated the effectiveness of our method as well as the challenging nature of our developed dataset.
[42] Guillermo Garcia-Hernando, Shantxin Yuan, Seungryul Baek, and Tae-Kyun Kim. First-person hand action benchmark with rgb-d videos and 3D hand pose annotations. In Proceedings of the IEEE conference on computer vision and pattern recognition, pages 409–419, 2018.

[43] Shantxin Yuan, Qi Ye, Bjorn Stenger, Siddhant Jain, and Tae-Kyun Kim. Bighand2. 2m benchmark: Hand pose dataset and state of the art analysis. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, pages 4866–4874, 2020.

[44] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2hands: Learning to infer 3D hands from conversational gesture body dynamics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 409–419, 2019.

[45] Alireza Sepas-Moghaddam, Ali Etemad, Fernando Pereira, and Paulo Lobato Correia. Facial emotion recognition using light field images with deep attention-based bidirectional lstm. In ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 3367–3371. IEEE, 2020.

[46] Alireza Sepas-Moghaddam, Ali Etemad, Paulo Lobato Correia, and Fernando Pereira. A deep framework for facial emotion recognition using light field images. In 2019 8th International Conference on Affective Computing and Intelligent Interaction (ACII), pages 1–7. IEEE, 2019.

[47] Alireza Sepas-Moghaddam, Ali Etemad, Fernando Pereira, and Paulo Lobato Correia. Long short-term memory with gate and state level fusion for light field-based face recognition. IEEE Transactions on Information Forensics and Security, 16:1365–1379, 2020.

[48] Yongyi Lu, Cewu Lu, and Chi-Keung Tang. Online video object detection using association lstm. In Proceedings of the IEEE International Conference on Computer Vision, pages 2344–2352, 2017.

[49] Rui Huang, Wanxue Zhang, Abhijit Kundu, Caroline Pantofaru, David A Ross, Thomas Funkhouser, and Alireza Fathi. An lstm approach to temporal 3D object detection in lidar point clouds. In Computer Vision–ECCV 2020: 16th European Conference, Glasgow, UK, August 23–28, 2020, Proceedings, Part XVIII 16, pages 266–282. Springer, 2020.

[50] Liang Zhang, Guangming Zhu, Lin Mei, Peiyi Shen, Syed Afzal Ali Shah, and Mohammed Bennamoun. Attention in convolutional lstm for gesture recognition. In Proceedings of the 32nd International Conference on Neural Information Processing Systems, pages 1957–1966, 2018.

[51] Evonne Ng, Shiry Ginosar, Trevor Darrell, and Hanbyul Joo. Body2hands: Learning to infer 3D hands from conversational gesture body dynamics. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition, pages 409–419, 2019.