3D Hand Pose Estimation in Point Cloud Using 3D Convolutional Neural Network on Egocentric Datasets

Van-Hung Le[1], Hai-Yen Tran[2]
[1] Tan Trao University, Vietnam
[2] Vietnam Academy of Dance, Vietnam
Correspondence: Van-Hung Le, email: van-hung.le@mica.edu.vn
Communication: received 26 November 2020, revised 27 December 2020, accepted 31 January 2021
Digital Object Identifier: 10.32913/mic-ict-research.v2020.n2.936

***Abstract*: 3D hand pose estimation from egocentric vision is an important study in the construction of assistance systems and modeling of robot hand in robotics. In this paper, we propose a complete method for estimating 3D hand pose from the complex scene data obtained from the egocentric sensor. In which we propose a simple yet highly efficient pre-processing step for hand segmentation. In the estimation process, we used the Hand Point Net (HPN), V2V-PoseNet (V2V), Point-to-Point Regression PointNet (PtoP) for fine-tuning to estimate the 3D hand pose from the collected data obtained from the egocentric sensor, such as CVRA, FPHA (First-Person Hand Action) datasets. HPN, V2V, PtoP are the deep networks/Convolutional Neural Networks (CNNs) for estimating 3D hand pose that uses the point cloud data of the hand. We evaluate the estimation results using the pre-processing step and do not use the pre-processing step to see the effectiveness of the proposed method. In particular, we evaluated the entire FPHA dataset with five different sampling configurations. The results show that 3D distance error is increased many times compared to estimates on the hand datasets are not obstructed (the hand data obtained from surveillance cameras, are viewed from top view, front view, sides view) such as MSRA, NYU, ICVL datasets. The results are quantified, analyzed, shown on the point cloud data of CVAR dataset and projected on the color image of FPHA dataset.**

Index Terms: *3D Hand Pose Estimation, Point Cloud Data, 3D Convolutional Neural Networks, Egocentric Vision Dataset.*

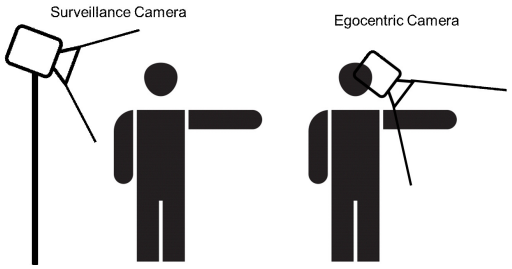

Figure 1: Illustrating the distinction between surveillance cameras and egocentric cameras.

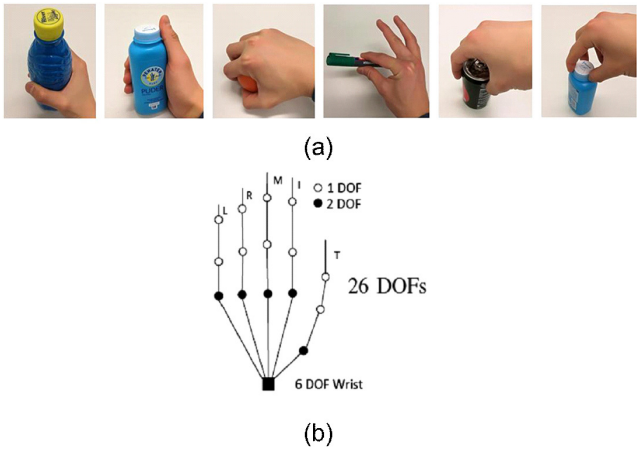

Figure 2: Illustration of some types of grasping the object and the number of Degrees Of Freedom (DOF) of the hand [1].

I. INTRODUCTION

Building robotic arms to perform complex actions like human hands is challenging research. Due to the joints on the human hand, there is a great Degree Of Freedom (DOF), as illustrated in Figure 2(b). In which the re-modeling of human hand actions is an approach [2]. From there build the action of fingers on robot arms like that, as simulating the human hand grasping objects in everyday life [3] (Figure 2(a)). To do this, hand poses in the 3D space / the real world need to be estimated.

The camera mounting is called ”Egocentric Camera” and the dataset obtained from this camera is called ”Egocentric Dataset”. Figure 1 shows the distinction between surveil-

Figure 3: Illustration of the RGB images captured from the mounted camera on the chest's person/ egocentric camera of FPHA dataset.

lance cameras and First-Person/Egocentric cameras. At the same time they are also defined in [4], [5]. The human hand in a real environment from an egocentric vision is missing due to the obscure data of the fingers, usually only obtained data of the palm as Figure 3. The human hand must first be detected, segmented, and fully estimated for joints in the 3D space. However, this issue is faced with many difficulties due to the data of the human hand is lost, obscured, and noises when collected from the egocentric vision.

Over the past four years, with the success of Convolutional Neural Networks (CNNs) in various computer vision, there have been about 60 valuable studies on estimating 3D hand pose using this method and the statistics of studies are shown in the first table of research by Le et al. [6]. The majority of studies on 3D hand pose estimation were evaluated on three outstanding datasets: MSRA [7], NYU [8], ICVL [9]. These datasets have the hand data that is sufficient (not obscured), thus the estimating results of the hand pose are low error, the results are shown in [6]. Recently, several CNNs have been proposed for 3D hand pose estimation for egocentric datasets on the RGB and based on the 3D joints annotation, their estimated results have low errors, as HOPE-Net [10]. The average error estimated HOPE-Net's hand pose on the FPHA [11] dataset was 8.61mm. This dataset have not been evaluated and compared with 3D CNNs appeared earlier as HPN [12], V2V [13], PtoP [14]. HPN, V2V, PtoP are the deep networks that the estimated results are high accuracy (the average of 3D joint error is 10.5 mm, 7.49mm, 7.71mm on the NYU dataset) on the datasets that captured the surveillance camera at the front view or top view. These CNNs estimate the joints of the hand based on point cloud data, it is the same as the real world. Currently, this data is collected from various types of depth sensors such as MS Kinect, Real scene, Creative Interactive Gesture.

The main contributions in our paper are as follows:

- An extension from the previous work [15] on 3D hand pose estimation are proposed. In this study, we deploy an end-to-end frame-work forfor 3D hand pose estimation from complex scene data that are obtained from the egocentric sensor. The proposed frame-work is a combination of the pre-processing step for the separating hand region in the complex scenes and a 3D CNN estimating a hand pose from the point cloud data.
- To adapt challenges when estimating hand poses from missing, obscured data, especially estimating the joints of the hands in the three dimension space,we performed the fine-tuning HPN, V2V, PtoP on the egocentric datasets such as CVAR [16], FPHA [11]. These CNNs achieve good estimation results because the training features are extracted from the point cloud data (3D data) of hand. In [15] we only evaluated performnaces of hand-pose estimation with the CVAR [16]. In this study, we expand the evaluations to the FPHA dataset. It is a large dataset with more complexity. Particularly, we evaluated multiple configuration schemes for training and testing samples.
- The evaluations are performed and shown on 3-D point cloud data. This type of data are more precisely and closed tothe real world expressions. In relevant works (e.g., in Doosti's study [10]), the estimation results are presented in color images. These presentations do not present intuitively poses and limit observing hand/joins in occlusion cases.

The paper is organized as follows: Section I introduces 3D hand pose estimation methods and the application. Section II discusses the related work by the methods, results, sensors, and the egocentric datasets. Section III presents 3D hand pose estimation by 3D CNNs. Section IV presents, discusses the experimental results of 3D hand pose estimation. Section V discusses the conclusion and future work of this paper.

II. Related works

The studies of estimating, restoring the full 3D hand skeleton, and pose involve many applications, the studies are heavily listed in the survey of Rui et al. [17]. This is a very comprehensive listing of 3D hand pose estimation and related studies: hand detection, hand tracking, hand parsing, fingertip detection, hand contour estimation, hand segmentation, gesture recognition. The challenges of estimating hand pose are also presented in the study of Rui et al. [17]: low resolution, self-similarity, occlusion, incomplete data, annotation difficulties, hand segmentation, real-time performance. 3D hand pose estimation is usually based on depth data obtained from depth sensors. The first table of research by Li et al. [17] lists 19 popular depth sensors that can collect depth data produced between 2010 and 2018 and

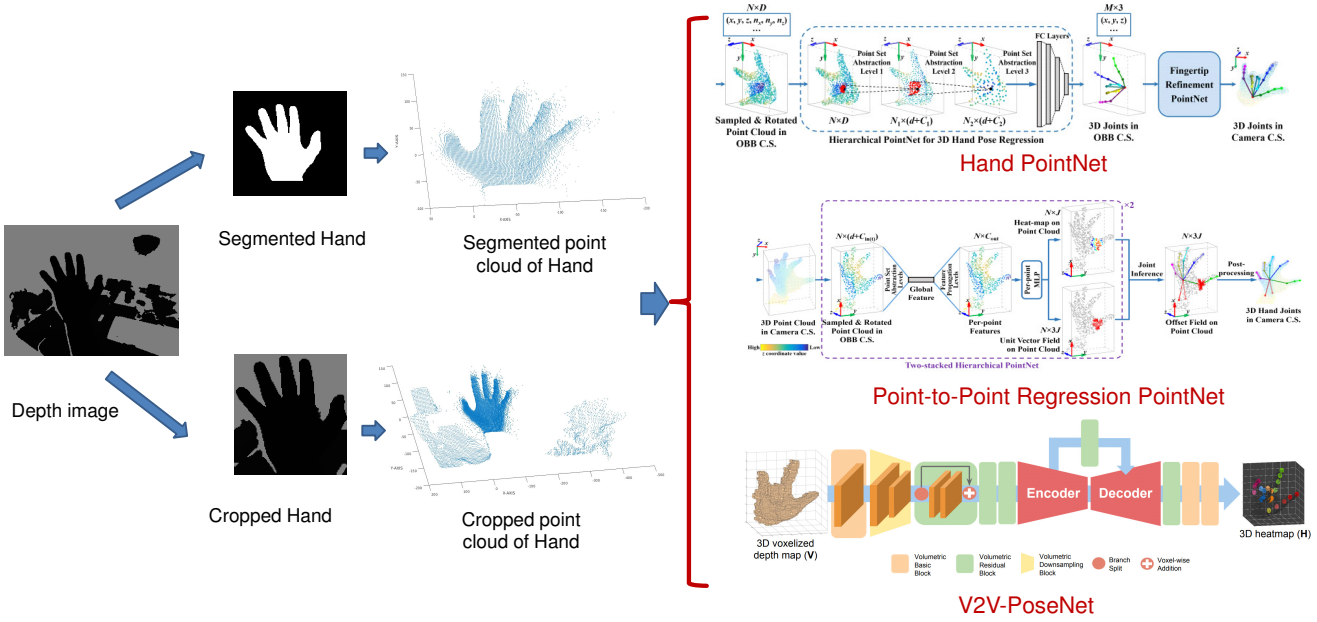

Figure 4: Overview of 3D hand pose estimation approach using 3D CNN-based from the point cloud data.

depth technology, measurement range, the maximum speed of depth data collection are also presented.

The depth sensors are divided into groups.In particle, the weight, the range limitation, and frame rate of the sensors have been improved. The weight, range limitation, and frame rate of MS Kinect v1 (2010) are about 907 grams, 0.5–4.5m, 30fps, respectively; The Intel real scene D435 (2018) are about 150 grams, 0.11–10 m, 90fps, respectively [17]. From this, the quality of the data collected is better.

For the hand pose estimation, Rui et al. [17] has presented three methods to solve 3D hand pose estimation: model-based method, appearance-based method, hybrid method. In which Erol et al. [18] also introduced two main methods (model-based method, appearance-based method) to solve this problem in the 2D space. The model-based method compares the hypothetical hand poses and the data obtained from the cameras. The comparison is evaluated based on objective function measures the discrepancy between the actual observations and the observations that are expected from the generative hand model. The appearance-based method based on learning the characteristics from the observations to a discrete set of the annotated hand poses. The model of training the annotated hand poses is the ability of the discriminative classifier or regression model to describe invariant characteristics of hand pose as a map of the joints of the fingers.

Estimating 3D hand pose has been interested in research with a large community in recent years, especially the method using CNNs. Doosti et al. [19] has been presented, there are two methods for estimating 3D hand pose: depth-based method and RGB-based method. In the depth-based method divided into two branches is to estimate the joints on the depth map of the image depth as researchers of Oberweger et al. [20], Zhang et al. [21], Wan et al. [22]; converting the depth image data to the point cloud data and estimate on it as researchers of Liuhao et al [23], Wan et al. [24], Ge et al. [25]. The RGB-based method trains the position of the annotated joints on the color image, then predicts the positions of the joints on the 2D heat-map. Finally, the regression of the 3D position of the joints is based on a synthesized of joints data or projected to 3D space through the depth image and point cloud data, respectively. Recently, the study of Doosti et al. [10] has very good results of 3D hand pose estimation on FPHA dataset (the average error is 6.81mm). This study used the ResNet10 to predict the initial 2D coordinates of the joints, after that using the graph convolution to estimate the better 2D pose. And then the predicted joints are passed to Graph U-Net [26] to find the 3D hand joints.

To evaluate the hand pose estimation methods, Rui et al. [17] proposed an evaluation experiment as shown in 4^{th} figure of research by Li et al. [17]. In particular, the benchmark datasets for evaluating 3D hand pose estimation were also presented: NYU [8], ICVL [27], and MSRA [28].

Another good survey of hand pose estimation is the study of Barsoum [29]. In this study, the author also conducted surveys on three methods to perform hand pose estimation from the depth image: appearance-based method is shown

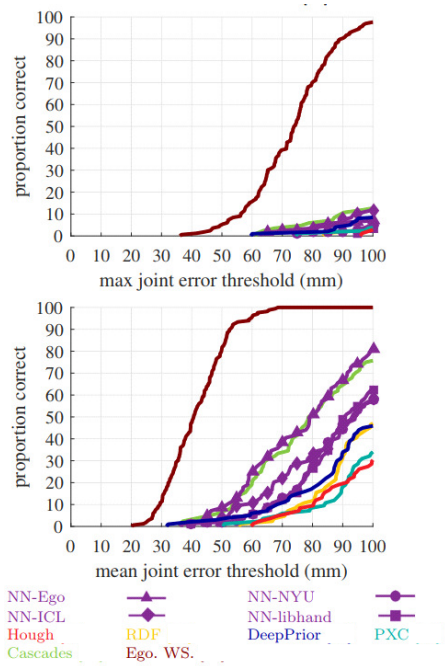

Figure 5: Distribution of 3D joints error estimation [36] on the UCI-EGO-Syn [37] dataset.

in 3^{th} figure, model-based method is shown in 4^{th} figure, the hybrid method is shown in 4^{th} figure of research by Barsoum et al. [29]. Hand segmentation steps are included in the three models because its outcome determines the accuracy of all these steps. This step can be solved by several methods as follows: Color or IR skin based; Temperature-based; Marker-based; Depth-based; Machine learning-based. In particular, Barsoum et al. [29] presented two studies that use deep learning to estimate hand pose in 2014 [8] and 2015 [30].

By egocentric dataset, already has a number of datasets published, these are listed as follows: UCI-EGO dataset [31], Graz16 dataset [16], Dexter+Object dataset [32], BigHand2.2M dataset [33], HO-3D Dataset [34]. The results of studies on estimating 3D hand pose before 2019 have great joints error, as illustrated in Figure 5. Recently, Doosti et al. [10] proposed and evaluated Hope-Net on FPHA [35] and HO-3D [34] datasets, the average 3D estimated joints error is very small (6.81mm).

III. 3D HAND POSE ESTIMATION FROM EGOCENTRIC VISION DATASETS

Different from previous studies by using CNNs for 3D hand pose estimation, they are evaluated on MSRA, NYU, ICVL datasets. These datasets are detected and segmented the human hands, as illustrated in Figur 6. In this paper, we have fine-tuning the HPN, V2V, PtoP to estimate 3D hand pose on the CVAR [16], FPHA [11] datasets. Before retrain the estimation model for 3D hand pose estimation on the CVAR dataset, as illustrated in Figure 4. We propose a simple pre-processing step to segment the hand data in the complex scenes. At the same time, we also re-trained and evaluated 3D hand pose estimation on the FPHA dataset that based on the 3D annotation of hand joints.

1. Hand segmentation from egocentric vision data

Previous studies on 3D CNNs estimated the hand pose on the hand data which have been segmented with data of the complex scenes. Ge et al. [14] used a single hourglass network [38] to detect 2D hand joint locations on the 2D heat-map of RGB image and use the corresponding depth information for hand segmentation. Ge et al. [12] used the random decision forest (RDF) [9] to segment the data of hand on the depth image. The presented methods of hand segmentation with the complex scene are complex, time consuming for feature extraction, hand model training.

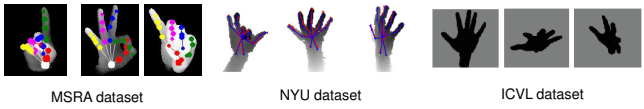

Figure 6: Illustration of the detected and segmented human hands on the MSRA, NYU, ICVL datasets.

Based on the contextual, as illustrated in Figure 3. The data of hand and the complex scene are collected from the egocentric vision sensor. Before performing 3D hand pose estimation from the point cloud data, we propose a pre-processing step to segment the hand from the complex scene data, as shown in Figure 4. The depth image contains the depth value of the hand data is the closest (the hand is closest to the sensor). From there we use a threshold d_{thres} which is the maximum depth value of the hand to segment the data of the hand and other data in the complex scene. The data segmentation algorithm on depth image obtained from egocentric vision is presented in Algorithm 1.

2. 3D hand pose estimation by Point Cloud-based

Different from previous CNN 3D hand pose estimation methods that take 2D images [22] or 3D volumes [25] as the input, the HPN, V2V, PtoP processes directly on the point cloud data generated from the depth images. The point cloud data of the segmented hand is represented by the Kd-tree [39] structure when used the HPN, V2V, PtoP on the MSRA, NYU, ICVL datasets.

a) HPN for 3D hand pose estimation

From this structure, it is possible to exploit the feature of the normal vector of points on the point cloud data. Due to a large number of points on the point cloud data and

Algorithm 1: The data segmentation algorithm on the depth image obtained from egocentric vision, based on the threshold.

Data: Depth image I_d
Result: Depth image I_h

$w = width(I_d)$;
$h = height(I_d)$;
$I_h = zeros(w, h)$;
for i=1 to w
  for j=1 to h
    $D_{value} = I_d(i, j)$;
    If($D_{value} > 0$ && $D_{value} < d_{thres}$)
      $I_h(i, j) = D_{value}$;
    else
      $I_h(i, j) = 0$;
    endif
  endfor
endfor

the 3D space, to reduce the volume of calculations and to make the network is robust, HPN has implemented the data standardization and rotation with the 3D point cloud in an Oriented Bounding Box (OBB) and data reduction in three levels, as illustrated in Figure 4(top branch): Level 1 data is standardized to 1024 points; Level 2 includes 512 points; Level 3 consists of 128 points. This is a simple pre-processing step, but it is highly effective for calculating time.

OBB is a tightly fitting bounding box of the input point cloud. The orientation of OBB is determined by performing the Principal Component Analysis (PCA) on the input point cloud. The x, y, z axes of the OBB coordinate system are determined by the eigenvectors of input points' covariance matrix, which correspond to eigenvalues from largest to smallest, respectively. The input point cloud p^{ori} is first transformed into OBB space, then these points are shifted to have zero mean and scaled to a unit size as follows Eq. (1).

$$p^{obb} = (R_{obb}^{ori})^T . p^{ori}, p^{nor} = (p^{obb} - \overline{p}^{obb})/L_{obb} \quad (1)$$

where R_{obb}^{ori} is the rotation matrix of the OBB in camera coordinate system; p^{ori} and p^{obb} are 3D coordinates of point p in camera coordinate system and OBB coordinate system, respectively; $\overline{p}^{obb}$ is the centroid of point cloud $\{p_i^{obb}\}_{i=1}^N$; L_{obb} is the maximum edge length of OBB; p^{nor} is the normalized 3D coordinate of point p in the normalized OBB coordinate system.

HPN is based on the PointNet [40], [41], it exploits the hierarchical PointNet for 3D hand pose estimation. The input of HPN is a set of normalized points PO, as presented in Eq. (2).

$$PO = \{po_i\}_{i=1}^N = \{p_i^{nor}, n_i^{nor}\}_{i=1}^N \quad (2)$$

where p_i is the 3D coordinate of the normalized point and n_i^{nor} is the corresponding 3D normal vector of p_i; N is the number of points of the standardized hand point cloud data then fed into a hierarchical PointNet [40].

The first two levels of group input points, as shown in Figure 4 (top branch), are $N_1 = 512$ and $N_2 = 128$ local regions, respectively and extract $C_1 = 128$ and $C_2 = 256$ dimensional features for each local region, respectively. Each local region contains k = 64 points. The last level extracts a 1024-dim global feature vector which is mapped to an F-dim output vector by three fully-connected layers. The PointNet is designed to output an F-dim $(F < 3 \times M)$ representation of the estimated 3D hand pose.

In the training process, HPN given the training samples with the normalized point cloud and the corresponding ground truth 3D joint locations $(po_T^{nor}, gr_T^{nor})_{t=1}^T$. It minimizes the following objective function, as shown in Eq. (3).

$$\theta^* = arg\min_{\theta} \sum_{t=1}^{T} \|\alpha_t - \gamma(po_T^{nor}, \theta)\|^2 + \lambda \|\theta\|^2 \quad (3)$$

where θ is the network parameters; γ represents the hand pose regression PointNet; λ is the regularization strength; α_t is an F-dim projection of gr_T^{nor}. By performing PCA on the ground truth 3D joint locations in the training dataset, the network can obtain $\alpha_t = E^T.(gr_T^{nor} - u)$, where E is the principal component, and u is the empirical mean.

In the testing process, the estimated 3D joint locations are reconstructed from the HPN outputs, as shown in Eq. (4).

$$\widehat{gr}^{nor} = E.\gamma(po^{nor}, \theta^*) + u \quad (4)$$

b) PtoP for 3D hand pose estimation

Just like the two methods presented, the input to estimate PtoP's 3D hand pose is also the point cloud data. To predict the hand poses that use the regression approach they are often require learn a highly non-linear mapping. However, if learning on all points in the point set is unnecessary and may make the per-point votes noisy, especially the training time is high. The main contributions of PtoP are to generate point-wise estimations of hand joint locations from the point cloud, which is able to better utilize the local evidence. The point-wise estimations can be defined as the offsets from points to hand joint locations. The process of estimating the offset of the hand joints is illustrated

in 2^{th} figure of research by Ge et al. [14]. PtoP uses the hierarchical PointNet [40] for learning heat-maps and unit vector fields on 3D point cloud. They are defined in Eq. (5) and Eq. (6), respectively.

$$H(p_i, \phi_j^*) = \begin{cases} 1 - ||\phi_j^* - p_i||/r & p_i \in \phi_j^* \\ and \quad ||\phi_j^* - p_i|| \leq r, \\ 0 \quad otherwise; \end{cases} \tag{5}$$

$$U(p_i, \phi_j^*) = \begin{cases} (\phi_j^* - p_i)/||\phi_j^* - p_i|| & p_i \in \phi_j^* \\ and \quad ||\phi_j^* - p_i|| \leq r, \\ 0 \quad otherwise; \end{cases} \tag{6}$$

where H is the heat-map of the neighboring points with the ground truth hand joint location; U is unit vector field/normal vector of the neighboring points with the ground truth hand joint location; $\phi_j^*(j = 1, ..., J)$ is is a set of K-Nearest Neighboring points (KNN) of the ground truth hand joint location; $p_i(i = 1, ..., N)$ is a set of points that want to estimate the offset; r is the maximum radius of ball for nearest neighbor search.

To get standard data as input to CNN, this method also performs the data normalization process to unify the view direction and size of the data. PtoP creates an Oriented Bounding Box (OBB) from the 3D point cloud and transform the 3D points into the OBB coordinate system, as shown in Figure 4. Standardized point cloud data is between -0.5 and 0.5 and includes 1024 points. The network architecture of PtoP uses a single hierarchical PointNet [40]. Base on the stacked hourglass networks method, PtoP uses two-stacked hierarchical PointNet modules end-to-end to boost the performance of the network with the same hyper-parameters. The architecture of the two stacks is shown in 4^{th} figure of research by Ge et al. [14]. To supervise the training process of the two-stacked hierarchical PointNet, PtoP uses the loss function which is defined in 4^{th} equation of research by Ge et al. [14].

c) V2V for 3D hand pose estimation

As shown in Figure 4 (bottom branch), the input of V2V method is 3D voxelized data. Thus, it reprojects each pixel of the depth map to the 3D space. After reprojecting all depth pixels, the 3D space is discretized based on the pre-defined voxel size. V2V-PoseNet is based on the hourglass model [38] and is designed to be divided into four volumetric blocks. The first volumetric basic block includes a volumetric convolution, volumetric batch normalization, and the activation function. This block is located in the first and last parts of the network. The second volumetric residual block extended from the 2D residual block in [42]. The third volumetric downsampling block that is identical to a volumetric max pooling layer. The last block is the volumetric upsampling block, which consists of a volumetric deconvolution layer, volumetric batch normalization layer, and the activation function.

Each phase of V2V-PoseNet method presented in Figure 4. Each phase consists of four blocks: Volumetric residual block, Volumetric downsampling block, Branch split, Voxel-wise addition. Therein, the volumetric downsampling block reduces the spatial size of the feature map while the volumetric residual bock increases the number of channels in the encoder phase. Otherwise, the volumetric upsampling block enlarges the spatial size of the feature map. When upsampling, the network decreases the number of channels to compress the extracted features. To predict each keypoint of the hand in 3D space through two stages: encoder, decoder. They are connected together by the voxel-wise. To supervise the per-voxel likelihood in the estimating process, V2V generates 3D heat-map, wherein the mean of Gaussian peak is positioned at the ground truth joint location in Eq. 7.

$$D_n^*(i, j, k) = exp\left(- \frac{(i - i_n)^2 + (j - j_n)^2 + (k - k_n)^2}{2\sigma^2} \right) \tag{7}$$

where D_n^* is the ground truth 3D heatmap of n^{th} keypoint, (i_n, j_n, k_n) is the ground truth voxel coordinate of n^{th}, and $\sigma = 1.7$ is the standard deviation of the Gaussian peak [13]. V2V also uses the mean square error as a loss function L in Eq. 8.

$$L = \sum_{n=1}^{N} \sum_{i,j,k} ||D_n^*(i, j, k) - D_n(i, j, k)||^2 \tag{8}$$

where D_n^* and D_n are the ground truth and estimated heatmaps for n^{th} keypoint, respectively, and N denotes the number of keypoints.

IV. Experimental Results

1. Datasets

In this paper, we conduct fine-tuning and evaluation on two egocentric vision datasets as following.

CVAR dataset [16] has more than 2000 depth frames of several egocentric sequences of six subjects. It captured from Creative Senz 3D, with intrinsics $fx = 224.5, fy = 230.5, ux = 160, uy = 120$ and contains several sequences of a single person. 3D annotations are made with 21 joint points, as illustrated in Figure 9. The size of the image is 320×240 pixels. The authors proposed a semi-automated the application that makes it easy to annotate sequences of articulated poses in the 3D space. This application asks

a human annotator to provide an estimate of the 2D re-projection of the visible joints in frames they are called reference frames. It proposes a method to automatically select these reference frames to minimize the annotation effort, based on the appearances of the frames over the whole sequence. It then uses this information to automatically infer the 3D locations of the joints for all the frames, by exploiting appearance, temporal and distances constraints. Figure 7 (top row) presents the depth images, the corresponding segmented human hands (bottom row), respectively.

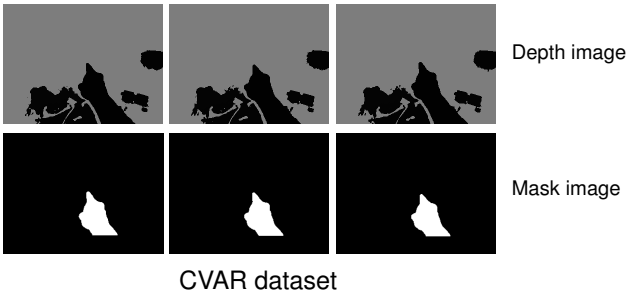

Figure 7: Illustration of CVAR [16] dataset.

FPHA dataset (First-Person Hand Action Dataset) [35] includes 100K frames, involving 26 different objects in several hand configurations. This dataset is collected from 6 people (6 subjects), 45 daily hand action categories, each action is collected from 3 to 9 times (instances), for a total of 1175 action sequences. It is captured from the Intel RealSense SR300, the size of color data is 1920x1080 pixels with *.jpeg* format and the size of depth data is 640x480 pixels with *.png* format, as illustrated in Figure 8. It also provided 3D hand joints annotation by the MOCAP system with 21 joints, as illustrated in Figure 9.

Figure 8: Illustration of FPHA [35] dataset. The MOCAP system used to make 3D hand joints annotation.

In this paper, to see the effect of the pre-processing step before performing 3D hand pose estimation. We evaluated the hand pose estimation on the segmented and cropped hand/no segmented hand data in the complex scenes of the CVAR dataset. On the FPHA dataset, the hand holds the object so the hand and object data are stuck together. Therefore, we perform fine-tuning and estimation on the cropped hand data based on the annotation data of the hand. This data is the same as the non-segmented hand data on the CVAR dataset.

2. CNNs parameters and evaluation measurement

We perform experiments on PC with Core i5 processor - RAM 8G, 4GB GPU. Pre-processing steps were performed on Matlab, fine-tuning, and development process in Python language on Ubuntu 18.04.

Hand PointNet parameters:

For training PointNets, we use Adam [43] optimizer with initial learning rate 0.001, batch size 8, and regularization strength 0.0005. The learning rate is divided by 10 after 50 epochs. The training process is stopped after 60 epochs. The number of PCA components is 42, the size of KNN search is 64. The square of radius for ball query in level 1, level 2 is 0.015, 0.04, respectively. This set of parameters is the same as in the experiment of [12]. Implementation details of HPN are shown in the link [1].

V2V-PoseNet parameters:

This deep network is developed in the PyTorch framework. The zero-mean Gaussian distribution with $\sigma = 0.001$ is initialized to all weights. The learning rate is set 0.00025 and batch size is set 4. The size of the input to the proposed system is $88 \times 88 \times 88$. This deep network also uses the optimizer method of Adam [43]. To standardize data for training, V2V rotates [-40, 40] degrees in XY space, scale [0.8, 1.2] in 3D space, and translate with the size of voxels [-8, 8] in 3D space. We trained the model for 15 epochs. Implementation details of V2V are shown in the link [2].

Point-to-Point Regression PointNet parameters:

The deep neural network models are implemented within the PyTorch framework. This deep network also uses the optimizer method of Adam [43], the learning rate 0.001, batch size 8, momentum 0.5 and weight decay 0.0005. The learning rate is divided by 10 after 30 epochs. The training process is stopped after 60 epochs. Implementation details of PtoP are shown in the link [3].

Evaluation measure:

Like the evaluations of the previous 3D hand pose estimation method, we used the average 3D distance error (as shown in Eq. 9) to evaluate the results of the 3D hand pose estimation on the datasets.

$$\widehat{Err_a} = \frac{1}{Num_s} \sum_{n=1}^{Num_s} \frac{1}{21} \sum_{k=1}^{21} DIS(p_g, p_e) \qquad (9)$$

[1] https://github.com/3huo/Hand-Pointnet.
[2] https://github.com/dragonbook/V2V-PoseNet-pytorch.
[3] https://github.com/ataata107/Point-to-Point-Regression-Hand-Pose.

where $DIS(p_g, p_e)$ is the distance between a ground truth joint p_g and an estimated joint p_e; Num_s is the number of testing frames. In this paper, we evaluated the 21 joints of hand pose, illustrated in Figure 9.

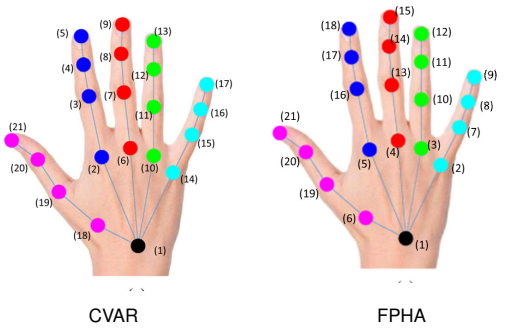

Figure 9: Illustrating the hand joints of CVAR and FPHA dataset.

In this paper, we use the rate at approximately 1:4 for testing and training samples with the CVAR dataset. This ratio is based on the setup given in [16]. This means that on the CVAR dataset using 1733 samples for training and 433 samples for testing. For each trial, the training and testing samples are selected randomly.

On the FPHA dataset, we used five configurations for training and testing. The first configuration used "Subject2", "Subject4", "Subject5", "Subject6" with 72260 samples for training and "Subject3" with 15770 samples for testing (the ratio is at approximately 1: 4 for testing and training model).

The second configuration ($Configu\_312$) used the 3^{rd} sequence in each subject "Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (23704 samples), the 1st sequence in each subject for validation (27097 samples) and remaining for training (54658 samples) (the ratio is at approximately 1:3 for testing and training model). The selection of samples was based on HOPE-Net [10] evaluations on the FPHA dataset.

The third configuration ($Configu\_123$) used the 1^{st} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (27097 samples), the 2^{nd} sequence in each subject for validation (25475 samples) and remaining for training (52887 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

The fourth configuration ($Configu\_213$) used the 2^{nd} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (25475 samples), the 1^{st} sequence in each subject for validation (27097 samples) and remaining for training (52887 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

The fifth configuration ($Configu\_321$) used the 3^{rd} sequence in each subject ("Subject1", "Subject2", "Subject3", "Subject4", "Subject5", "Subject6") for testing (23704 samples), the 2^{nd} sequence in each subject for validation (25475 samples) and remaining for training (56280 samples) (the ratio is at approximately 1: 2.5 for testing and training model).

In this paper, we have evaluated with many different sampling configurations training and testing 3D hand pose estimation models. From the second configuration to the fifth configuration (this experiment is based on the multi-fold not just n-fold cross-validation), we only evaluate the **cropped hand** method.

3. Results and Discussions

Like the evaluation of 3D hand pose estimation results shown in the 2^{nd} table of research by Yoo et al. [44], we also use the 3D distance error (mm) to evaluate the estimation results on CVAR and FPHA datasets with the five configurations. The average 3D distance error and the average processing time on the first configuration are shown in Table I, II, respectively.

The processing time of 3D hand pose estimation is shown in Table II. This time includes the hand segmentation time or the cropping hand time on the depth image. The distribution of the average 3D distance error when evaluating on CVAR dataset is presented in Figure 10.

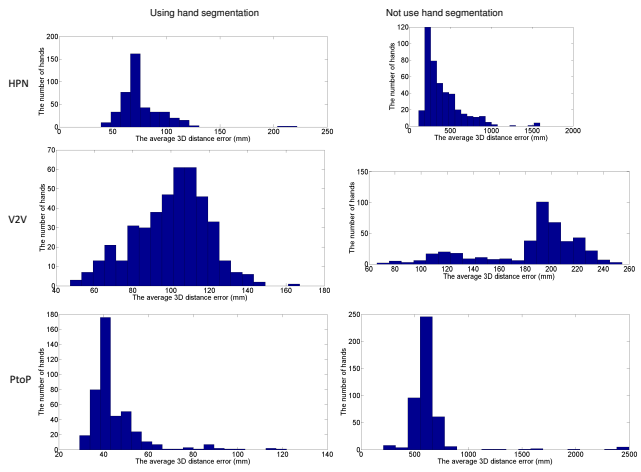

Figure 10: The distribution of average 3D distance error for 3D hand pose estimation on the CVAR [16] dataset when using the hand segmentation and not use the hand segmentation. The left is the error distribution of estimating 3D hand pose from the point cloud data using the hand segmentation. The right is the error distribution of estimating 3D hand pose from the point cloud data which not use the hand segmentation. The top line is based on HPN method; The mid line is based on V2V method; The bottom line is based on PtoP method.

TABLE I: The average 3D distance error (mm) of HPN [12], V2V [13], PtoP [14] on the CVAR, FPHA (the first configuration) datasets for 3D hand pose estimation when using the hand segmentation and the cropped hand.

Dataset	**Measurement/ Method**		**HPN [12]**	**V2V [13]**	**PtoP [14]**
CVAR	**Average of 3D joints error (mm)**	**Hand segmentation**	**77.53**	**100.58**	**45.27**
		Cropped hand	450.87	184.88	617.99
FPHA		**Cropped hand**	144.16	241.23	121.74

TABLE II: The average processing time (mm) of HPN [12], V2V [13], PtoP [14] to estimate a 3D hand pose on the CVAR dataset when using the hand segmentation and the cropped hand.

Measurement/ Method		**HPN [12]**	**V2V [13]**	**PtoP [14]**
Processing time (ms)/ hand	**Hand Segmentation**	**12.44**	**504.22**	**197.81**
	Cropped hand	17.26	638.25	198.52

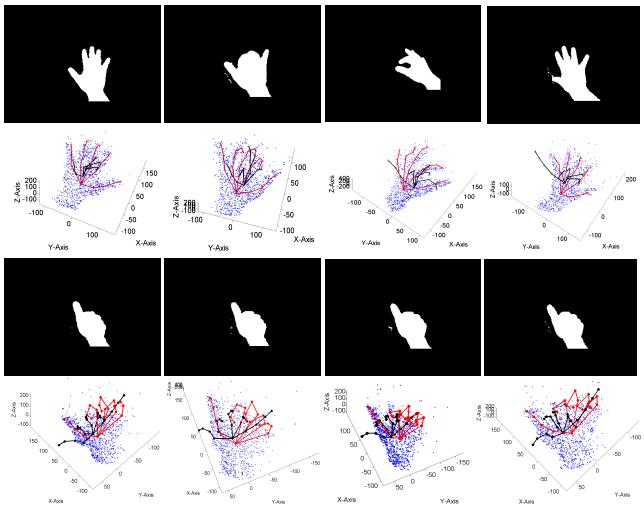

Figure 11: 3D hand pose estimation results on the CVAR [16] dataset; Top line is the depth images; The bottom line is 3D hand pose estimation results. The blue point cloud is the normalized point cloud of the hand. The red hand skeleton is 3D ground truth of 3D hand pose. The black hand skeleton is the estimated 3D hand pose.

Based on the results in Table I and Figure 10 when using the hand segmentation, the estimated error of joints in the 3D space on the CVAR dataset is more than 9 times higher than the errors on the MSRA dataset, more than 7 times the error on the NYU dataset, 11 times the error on the ICVL dataset, respectively. This problem starts from the complexity of the CVAR dataset, the data of the hand is obscured, is missing because the view direction of the sensor only sees the palm, the data of the fingers is obscured, as shown in Figure 11. Another problem is that the depth value of the hand data on the CVAR dataset is larger than the MSRA, NYU, ICVL. For example, the minimum hand depth value of 10 frames on CVAR is about 510mm while the minimum hand depth value of 10 frames

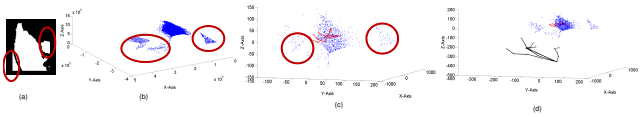

Figure 12: Illustration of estimation result on the hand data that cropped/did not segmented in the complex scene. (a) is the depth image data of the hand cut based on the ground truth data; (b) is point cloud data that has not been sample reduced; (c) is the point cloud data that is sampled and normalized, the ground truth skeleton/joints data; (d) is 3D hand pose estimation results based on standardized data based on HPN (black skeleton). The data areas marked with red circles are data of other objects in the complex scene.

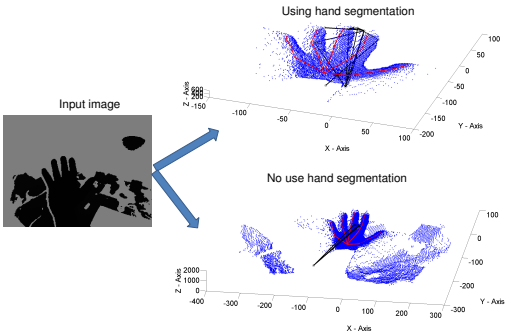

Figure 13: Illustration of 3D hand pose estimation result on the hand data when using hand segmentation and the cropped hand/not use hand segmentation in the complex scene that base on the V2V method. The blue points are the point cloud data of the hand without sample reduction and data normalization. The red skeleton is 3D ground truth of hand pose, the black skeleton is 3D hand pose estimation.

on MSRA is about 350mm. This means that the hand on the CVAR dataset is far more sensor than the MSRA dataset.

Figure 11 illustrates the result of estimating 3D hand pose on the obscured data by these fingers in the CVAR dataset. Although on the segmented hand there is little data noise, as illustrated in Figure 11. However, this data is few and not far from the hand data, so the estimation range of the hand pose is not much larger, so the estimation results when using the hand data segmentation is better more than in the case of not using the hand data segmentation.

When the hand data is not segmented in a complex scene the error is greatly increased (from 77.53 to 450.87 on the CVAR dataset), as shown in Table I and illustrated in Figure 12. This result is greatly increased because the estimated space for 3D hand joints on the non-segmefnted data is very large and there is a high dimension in the 3D space.

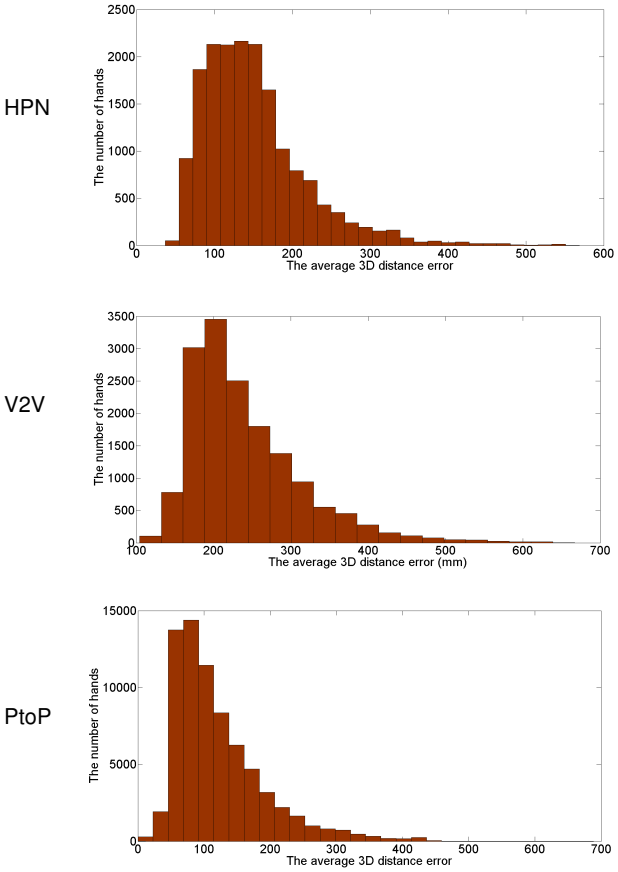

Figure 14: The distribution of average 3D distance error for 3D hand pose estimation on the FPHA [35] dataset when not use the hand segmentation. The top line is based on HPN method; The mid line is based on V2V method; The bottom line is based on PtoP method.

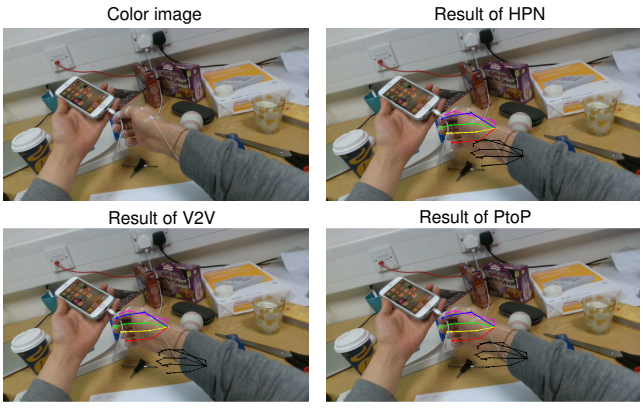

Figure 15: The estimated results on color image projected from 3D hand pose data by HPN, V2V, PtoP on the FPHA [35] dataset. 3D ground truth hand data with fingers magenta, blue, green, yellow, red. The estimated 3D pose of the hand is black.

The estimated 3D hand joints error and error distribution on the FPHA dataset with the first configuration are shown in Table I and Figure 14, respectively. The estimated 3D hand joints error is from 144.16 to 241.23mm.

The second configuration to fifth configuration results on the testing sets is shown in Table III. Although the estimated error was much lower than the results in the first configuration. However, their results are many times higher than the estimated results using HOPE-Net (6.81mm). The estimated results improved due to the posture learning of 6 people performing the activities. In the first configuration, only five people can learn the activities. Each person has a different hand size, while performing the same activity, each person's style is different. Especially, the training is done on 3D joints data, so it contains many challenges. The results of the FPHA dataset were evaluated on five configurations, in HOPE-Net only evaluated on a configuration similar to $Configu\_312$, but only evaluated on about 1000 samples. In this paper, we evaluate the entire dataset.

TABLE III: The average 3D distance error (mm) of HPN [12], V2V [13], PtoP [14] on the FPHA dataset with $Configu\_312$, $Configu\_123$, $Configu\_213$, $Configu\_321$ for 3D hand pose estimation when using the cropped hand.

Dataset	Measurement	HPN	V2V	PtoP
Configu_312	**Average of 3D joints error (mm)**	135.54	**115.34**	154.33
Configu_123		140.26	**123.85**	160.48
Configu_213		139.42	**112.78**	188.72
Configu_321		147.86	**129.46**	179.82

Figure 16 presented the average 3D distance error (mm) on each epoch of HPN that is based on the second configuration to the fifth configuration. Figure 15 shows the results of 3D hand pose estimation of HPN, V2V, PtoP projected from 3D data to color image.

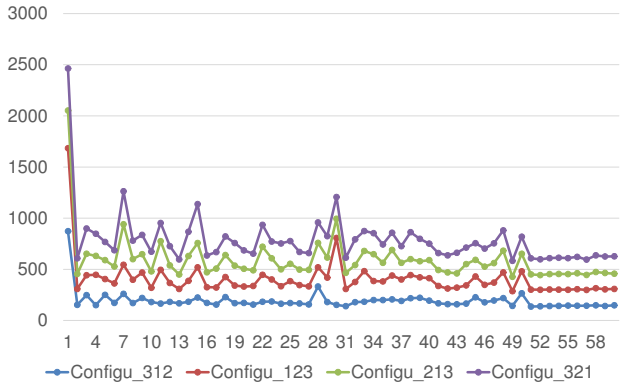

Figure 16: The average 3D distance error (mm) on each epoch of HPN that based on the second configuration to fifth configuration.

All three CNNs (HPN, V2V, PtoP) that we use for training and estimating 3D hand pose are in feature-based methods. Estimation results always depend entirely on the characteristics learned from the datasets. Therefore, the estimation results depend on the features learned from the two datasets (CVAR, FPHA). The fact that FPHA is a dataset which is contructed with many activities in daily life. These activities are perfomed in a natural way. The features arecalculated and extracted based on the point

cloud data (3D data) of the hand data as illustrated in Figure 4. Ultilizing the trained models, one can easily deploy transfer learning for a new hand pose dataset. For instance, another method such as HOPE-Net utilizing trained model from FPHA to detect the key points on an RGB image and then deploy the Adaptive Graph U-Net technique to estimate the detected key pointsinto 3D space. Because it always take time and label costs to prepare ground-truth of 3-D hand pose datasets. In the liteturatue works, there is not so many available datasets which support 3-D hand pose's ground-truth. We will continue consider other 3-D egocentric vision datasets whose ground-truth measurement are available.

V. Conclusion

Estimating the 3D hand pose from image data obtained from egocentric vision is an issue that contains many challenges. Most recent studies of 3D hand pose estimation using CNNs were evaluated on simple datasets such as MSRA, NYU, ICVL. These datasets have the data of hands that are complete. However, in practice developing the human assistance systems, such as assistance systems that grasping objects in complex environments, the sensors are often mounted on the person. So the hand data is often obscured by the objects or only the palm area is visible. To see the challenges in these data for fully estimating/restoring hands pose in the 3D space. We proposed a complete pipeline for estimating 3D hand pose from the complex scene data obtained of the egocentric sensor and performed an experiment using the common CNNs (HPN, V2V, PtoP) to estimate hand pose on the datasets obtained from an egocentric vision sensor (CVAR, FPHA datasets). The result found that the 3D hand pose estimation error is increased many times compared to the estimate on the full hand data datasets when using the hand segment and cropped hand/not using the hand segment. We evaluated the 3D hand pose estimation results across various configurations and all frames of the FPHA dataset. In particular, the results show that when using 3D CNNs to estimate the hand pose on the point cloud data, there is a large mean error (greater than 100mm). The results are displayed on the CVAR dataset, illustrated in the point cloud data and projected onto the color image on the FPHA dataset.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.01-2019.315.

References

[1] Z. Deng, G. Gao, S. Frintrop, F. Sun, C. Zhang, and J. Zhang, "Attention based visual analysis for fast grasp planning with a multi-fingered robotic hand," *Frontiers in Neurorobotics*, vol. 13, no. July, pp. 1–12, 2019.

[2] M. F. Reis, A. C. Leite, and F. Lizarralde, "Modeling and control of a multifingered robot hand for object grasping and manipulation tasks," in *Proceedings of the IEEE Conference on Decision and Control*, vol. 54rd IEEE. IEEE, 2015, pp. 159–164.

[3] J. A. Corrales, C. A. Jara, and F. Torres, "Modelling and simulation of a multi-fingered robotic hand for grasping tasks," in *11th International Conference on Control, Automation, Robotics and Vision, ICARCV 2010*, 2010, pp. 1577–1582.

[4] A. Fathi, A. Farhadi, and J. M. Rehg, "Understanding egocentric activities," in *Proceedings of the IEEE International Conference on Computer Vision*, 2011, pp. 407–414.

[5] S. Mann, K. M. Kitani, Y. J. Lee, M. S. Ryoo, and A. Fathi, "An introduction to the 3rd workshop on egocentric (first-person) vision," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops*, 2014, pp. 827–832.

[6] L. Van-Hung and N. Hung-Cuong, "A survey on 3d hand skeleton and pose estimation by convolutional neural network," *Advances in Science, Technology and Engineering Systems Journal*, no. 7, pp. 144–159, 2020.

[7] H. Su, S. Maji, E. Kalogerakis, and E. Learned-Miller, "Multi-view convolutional neural networks for 3d shape recognition," in *Proceedings of the IEEE international conference on computer vision*, 2015, pp. 945–953.

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

[9] D. Tang, H. Chang, A. Tejani, and T. Kim, "Latent regression forest: Structured estimation of 3D hand poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1374–1387, 2017.

[10] B. Doosti, S. Naha, M. Mirbagheri, and D. Crandall, "Hope-net: A graph-based model for hand-object pose estimation," in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[11] G. Garcia-Hernando, S. Yuan, S. Baek, and T.-K. 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*, 2018, pp. 409–419.

[12] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet : 3D Hand Pose Estimation using Point Sets," *Cvpr*, pp. 3–5, 2018. [Online]. Available: http://openaccess.thecvf.com/content-cvpr2018/papers/GeHandPointNet3DCVPR2018paper.pdf

[13] G. Moon, J. Y. Chang, and K. M. Lee, "V2V-PoseNet: Voxel-to-Voxel Prediction Network for Accurate 3D Hand and Human Pose Estimation from a Single Depth Map," in *IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR*, 20118, pp. 5079–5088.

[14] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *European Conference on Computer Vision*, vol. 11217 LNCS, 2018, pp. 489–505.

[15] L. Van-Hung, H. Van-Nam, V. Hai, L. Thi-Lan, T. Thanh-Hai, and V. Viet-Vu, "Using Hand PointNet-based 3D Hand Pose Estimation in Egocentric Datasets," in *2020 International Conference on Advanced Technologies for Communications (ATC)*, 2020, pp. 215–220.

[16] M. Oberweger, G. Riegler, P. Wohlhart, and V. Lepetit, "Efficiently creating 3D training data for fine hand pose estimation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2016-Decem, 2016, pp. 4957–4965.

[17] Y. Li, Z. Xue, Y. Wang, L. Ge, Z. Ren, and J. Rodriguez, "End-to-End 3D Hand Pose Estimation from Stereo Cameras," in *BMVC*, 2019, pp. 1–13.

[18] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "Vision-based hand pose estimation: A review," *Computer Vision and Image Understanding*, vol. 108, no. 1-2, pp. 52–73, 2007.

[19] B. Doosti, "Hand Pose Estimation: A Survey," *CoRR*, vol. abs/1903.01013, 2019. [Online]. Available: http://arxiv.org/abs/1903.01013

[20] M. O., P. W., and V. L., "Training a feedback loop for hand pose estimation," in *Proceedings of the IEEE International Conference on Computer Vision*, vol. 2015 Inter, 2015, pp. 3316–3324.

[21] Y. Zhang, C. Xu, and L. Cheng, "Learning to Search on Manifolds for 3D Pose Estimation of Articulated Objects," *CoRR*, vol. abs/1612.00596, 2016. [Online]. Available: http://arxiv.org/abs/1612.00596

[22] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Crossing Nets : Combining GANs and VAEs with a Shared Latent Space for Hand Pose Estimation," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[23] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D Hand Pose Estimation from Single Depth Images Using Multi-View CNNs," *IEEE Transactions on Image Processing*, vol. 27, no. 9, pp. 4422–4436, 2016.

[24] C. Wan, A. Yao, and L. Van Gool, "Hand pose estimation from local surface normals," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 9907 LNCS, 2016, pp. 554–569.

[25] L. G., H. Liang, J. Yuan, and D. Thalmann, "3D Convolutional Neural Networks for Efficient and Robust Hand Pose Estimation from Single Depth Images," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[26] H. Gao and S. Ji, "Graph u-nets," in *International Conference on Machine Learning*, 2019, pp. 2083–2092.

[27] D. Tang, H. J. Chang, A. Tejani, and T. K. Kim, "Latent regression forest: Structured estimation of 3D hand poses," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 7, pp. 1374–1387, 2017.

[28] X. Sun, Y. Wei, S. Liang, X. Tang, and J. Sun, "Cascaded hand pose regression," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 07-12-June, 2015, pp. 824–832.

[29] E. Barsoum, "Articulated Hand Pose Estimation Review," *CoRR*, vol. abs/1604.06195, pp. 1–50, 2016. [Online]. Available: http://arxiv.org/abs/1604.06195

[30] M. Oberweger, P. Wohlhart, and V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," in *Computer Vision Winter Workshop*, 2015. [Online]. Available: http://arxiv.org/abs/1502.06807

[31] G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan, "3d hand pose detection in egocentric rgb-d images," in *European Conference on Computer Vision*. Springer, 2014, pp. 356–371.

[32] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from rgb-d input," in *Proceedings of European Conference on Computer Vision (ECCV)*, October 2016. [Online]. Available: http://handtracker.mpi-inf.mpg.de/projects/RealtimeHO/

[33] S. Yuan, Q. Ye, B. Stenger, S. Jain, and T. K. Kim, "BigHand2.2M benchmark: Hand pose dataset and state of the art analysis," in *Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017*, vol. 2017-Janua, 2017, pp. 2605–2613.

[34] S. Hampali, M. Rad, M. Oberweger, and V. Lepetit, "Honnotate: A method for 3d annotation of hand and object poses," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 3196–3206.

[35] G. Garcia-Hernando, S. Yuan, S. Baek, and T. K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2018, pp. 409–419.

[36] J. S. Supančič, G. Rogez, Y. Yang, J. Shotton, and D. Ramanan, "Depth-based hand pose estimation: methods, data, and challenges," *International Journal of Computer Vision*, vol. 126, no. 11, pp. 1180–1198, 2018.

[37] G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan, "3d hand pose detection in egocentric rgb-d images," in *European Conference on Computer Vision*. Springer, 2014, pp. 356–371.

[38] N. A., Y. K., and D. J., "Stacked hourglass networks for human pose estimation," in *In European Conference on Computer Vision*, 2016.

[39] Y. Aggarwal, "K Dimensional Tree (K D Tree)," https://iq.opengenus.org/k-dimensional-tree/, [Accessed 25 July 2020].

[40] Q. C., Y. L., S. H., and G. L., "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5105–5114.

[41] C. Qi, L. Yi, H. Su, and L. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2017.

[42] H. Kaiming, Z. Xiangyu, R. Shaoqing, and S. Jian, "Deep Residual Learning for Image Recognition," in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[43] P. K. Diederik and B. Jimmy, "Adam: A Method for Stochastic Optimization," in *In ICLR*, 2015.

[44] C. Yoo, S. Kim, S. Ji, Y. Shin, and S. Ko, "Capturing Hand Articulations using Recurrent Neural Network for 3D Hand Pose Estimation," *CoRR*, vol. abs/1911.07424, 2019. [Online]. Available: http://arxiv.org/abs/1911.07424

Van-Hung Le received M.Sc. degree at Faculty Information Technology Hanoi National University of Education (2013). He received PhD degree at International Research Institute MICA HUSTCNRS/UMI - 2954 - INP Grenoble (2018). Currently, he is a lecture of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; machine leaning, deep learning.
Email: van-hung.le@mica.edu.vn

Hai-Yen Tran received Bachelor degree at Faculty Information Technology National Economics University (2009). She received M.Sc. degree at Faculty Information Technology Hanoi National University of Education (2013). Currently, she is a lecture of Vietnam Academy of Dance. Her research interests include computer science, deep learning.
Email: yenth@vnad.edu.vn