*Advances in Science, Technology and Engineering Systems Journal*
*Vol. 6, No. 1, 01-09 (2021)*
www.astesj.com
*Special Issue on Multidisciplinary Innovation in Engineering Science & Technology*

**ASTES Journal**
**ISSN: 2415-6698**

# Standardized UCI-EGO dataset for evaluating 3D hand pose estimation on the point cloud

Sinh-Huy Nguyen[1], Van-Hung Le[*,2]

[1]*Institute of Information Technology, MIST, 100000, Vietnam*

[2]*Tan Trao University, Tuyen Quang, 22000, Vietnam*

ARTICLE INFO

*Article history:*
*Received: 14 September, 2020*
*Accepted: 23 December, 2020*
*Online: 10 January, 2021*

*Keywords:*
*UCI-EGO dataset*
*3D CNNs*
*3D hand pose estimation*
*Point cloud data*

ABSTRACT

*To evaluate and compare methods in computer vision, scientists must use a benchmark dataset and unified sets of measurements. The UCI-EGO dataset is a standard benchmark dataset for evaluating Hand Pose Estimation (HPE) on depth images. To build robotic arms that perform complex operations such as human hands, the poses of the human hand need to be accurately estimated and restored in 3D space. In this paper, we standardized the UCI-EGO dataset to evaluate 3D HPE from point cloud data of the complex scenes. We also propose a method for fine-tuning a set parameter to train the estimation model and evaluating 3D HPE from point cloud data based on 3D Convolutional Neural Networks (CNNs). The CNNs that we use to evaluated currently the most accurate in 3D HPE. The results of the 3D HPE from the point cloud data were evaluated in two branches: using the hand data segment and not using the hand data segment. The results show that the average of 3D joint errors of the 3D HPE is large on the UCI-EGO dataset (87.52mm) and that the error without using the hand data segment is many times higher than the estimated results when using the hand data segment (0.35ms). Besides, we also present the challenges of estimating 3D hand pose and the origin of the challenge when estimating real image dataset.*

## 1 Introduction

In computer vision when evaluating and comparing the methods, the scientists must use a benchmark dataset and unified sets of measurements. The benchmark dataset usually includes training sets and testing sets/ validation set [1], this ratio is defined in the cross-validation parameter [2]. And these sets just include the annotation data of each sample. The UCI-EGO dataset has been published in [3] [1] and evaluated in many studies of HPE [4]–[6]. However, these ratings are evaluated in 2D space on the depth image. The UCI-EGO dataset provided the annotation data, each key point is represented by the structure $(x, y, z)$, $(x, y)$ are the coordinates on the depth image, $z$ is the depth value of pixel which has the coordinates $(x, y)$. However, the actual depth values of the annotation data are different from the depth data on the depth image. They are shown as Figure 2.

Nowadays, building robotic arms with hands that can perform many complex actions like human hands is an issue that needs research [7]. Since the human hands have many degrees of freedom (DOF), the complex actions can be performed. In order to build a robotic hand that can perform complex operations (Figure 3), first of all, it is necessary to restore and estimate the hand poses in the 3D space. Therefore, we continue to perform research on estimating human hand pose in the 3D space. In particular, estimating the hand pose on the data obtained from the EGOcentric VIsion (EGO-VI) sensor, contains many challenges such as missing, data loss, or obscuring.

Therefore, this paper includes the main contributions as follows:

- Standardizing the annotation data of the UCI-EGO dataset based on the depth value on the depth image. That means replacing the depth value of each point in the annotation data with the depth value of the corresponding point at that coordinate on the depth image, as illustrated in Figure 4. From this, using 3D hand pose annotation data to train the hand pose estimation model in the 3D space.

- Fine-tuning a set of parameters to train the hand pose estimation model in the 3D space based on the (V2V - V2V-PoseNet [8]) and evaluating the 3D HPE based on the most accurate 3D CNN (V2V). The estimation results are presented and

[*] Corresponding Author: Van-Hung Le, Tan Trao University, & Lehung231187@gmail.com
[1] `https://github.com/hassony2/inria-research-wiki/wiki/uci-ego-dataset`.

https://dx.doi.org/10.25046/aj060101

evaluated on the point cloud data. This is the same data as the real world.

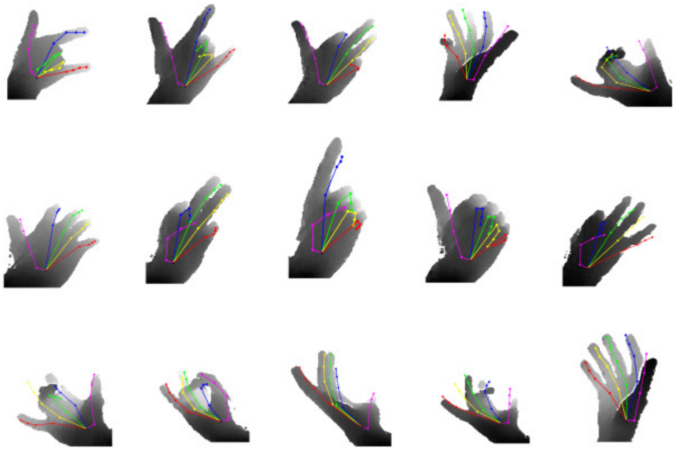


Figure 1: Illustration of 2D HPE results on the UCI-EGO dataset [6].

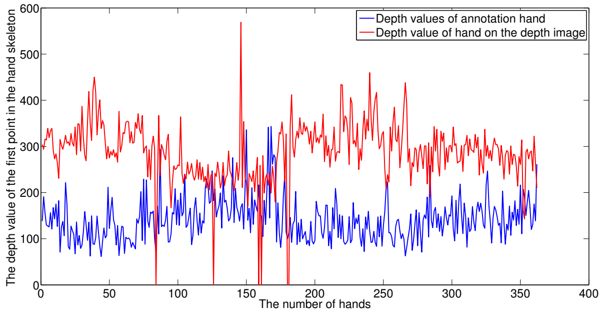


Figure 2: The depth value of the first point in the hand skeleton of annotation data and depth image data.

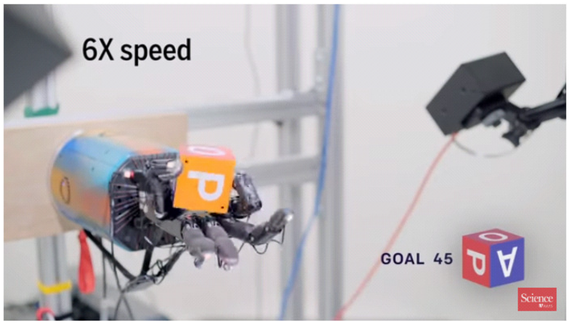


Figure 3: Robot arm illustration follows the operation of human hands [7].

- Presenting and comparing some 3D HPE results on the full hand dataset and the dataset obtained from the EGO-VI sensor. Presenting some challenges of estimating hand pose in the 3D space when estimating on the data obtained from EGO-VI sensors.

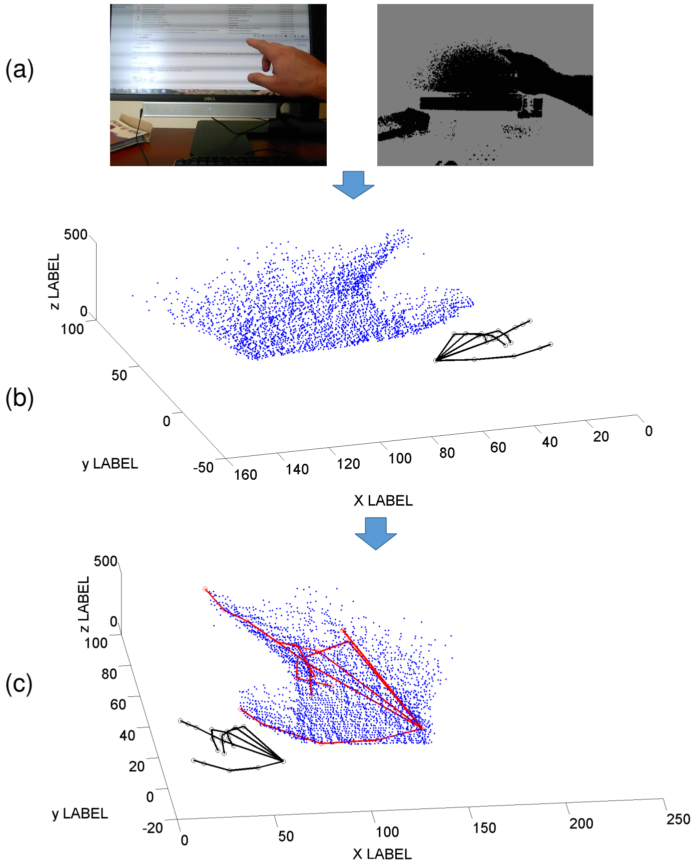


Figure 4: Illustrating of standardized the UCI-EGO annotation data process. (a) is the RGB-D images; (b) is the annotation data of UCI-EGO (the blue points are the point cloud of hand, the black skeleton is the annotation of UCI-EGO dataset); (c) is the standardized the annotation of UCI-EGO.

During the research on recognizing the daily activities of the human hand based on data collected from the EGO-VI sensor. We only research within a limited range as follows: We are interested in the dataset obtained from the EGO-VI sensor, namely the UCI-EGO dataset studied in this paper; We are also interested in 3D CNN that use point cloud as the input because the point cloud data is real data similar to the real environment.

The organization of the paper is shown as follows: Section 1 first introduce the benchmark dataset, the existence problem of the UCI-EGO [3] dataset, and the 3D HPE problem, we also introduce the application of 3D HPE to build robotic arms. Section 2 presents studies on the benchmark datasets to evaluate 3D HPE and some results. Section 3 presents the standardization of the UCI-EGO dataset and 3D HPE in the point cloud data. Section 4 presents the results and some discussions of 3D HPE. Finally, there are some conclusions and the next research direction of the paper (Section 5).

## 2 Related Works

Evaluating on the benchmark datasets is an important step to confirm the correctness of the detection, recognition, and estimation model of computer vision. Currently, there are many datasets for evaluating 3D HPE. The datasets are listed in Tab. 6 of [9]. In this paper, we only reintroduce some of the datasets used to evaluate 3D

HPE and some results based on typical CNNs.

In [10], published MSRA dataset, [2]. It includes 76k depth images of nine subjects of the right hands are captured using Intel's Creative Interactive Gesture Camera. Each subject include 17 gestures captured and include about 500 frames with 21 3D annotation hand joints for each frame: wrist, index mcp(metacarpal bone), index pip(proximal phalanges), index dip(distal phalanges), index tip, middle mcp(metacarpal bone), middle pip(proximal phalanges), middle dip(distal phalanges), middle tip, ring mcp (metacarpal bone), ring pip(proximal phalanges), ring dip(distal phalanges), ring tip, little mcp(metacarpal bone), little pip(proximal phalanges), a little dip(distal phalanges), little tip, thumb mcp(metacarpal bone), thumb pip(proximal phalanges), thumb dip(distal phalanges), and thumb tip. The size of the captured image is $320 \times 240$ pixels. The camera's intrinsic parameters are also provided, i.e. principal point of the image is (160, 120) and the focal length is 241.42. This dataset only has depth images, especially the hand data that is segmented with environmental data. This is a benchmark dataset for the evaluation of 3D HPE, the results of the CNNs are shown in table 2 of [11].

In [12] [3], the author includes 72,757 frames of the training set captured from a single person and 8,252 frames of the testing set captured two different persons from three MS Kinect v1, i.e. a frontal view and two side views. Each frame is a couple of RGB-D images. This dataset provided 25-joints in the annotation data with 42 DOF. The authors used the Randomized Decision Forest (RDF) to train a binary classification model by this dataset. And then this classification segments each pixel that belongs to a hand or background in the depth image. 3D HPE results of the CNNs are shown in table 2 of [11].

In [13] [4], the author includes 22K frames for training and 1.6K frames for testing, they captured from the Intel's Creative Interactive Gesture Camera with 10 subjects to take 26 different poses. It also provides 3D annotation data with 16 hand joints: palm, thumb root, thumb mid, thumb tip, index root, index mid, index tip, middle root, middle mid, middle tip, ring root, ring mid, ring tip, pinky root, pinky mid, and pinky tip.

The above are the datasets collected from a fixed number of perspectives of the image sensors. In many real applications, the image sensors are mounted on the body to collect data from the environment. These datasets are named the "Egocentric" dataset. In [14], the author published the **UCI-EGO** dataset, in [15] the author published the **Graz16** dataset, in [16] the author published the **Dexter+Object** dataset, in [17] the author published the **First-Person Hand Action** (FHAD) dataset, in [18] the author published the **UCI-EGO-Syn** dataset, in [15] the author published the **CVAR** dataset. Most EGO-VI datasets contain challenges for 3D HPE as follows: The frames do not contain hands, and the hands are obscured by objects in the scene; Data of the fingers is obscured; Data of visible hand only data of palm. The percent of challenges in CVRA [15] and UCI-EGO-Syn [18] datasets are shown in Figure 5.

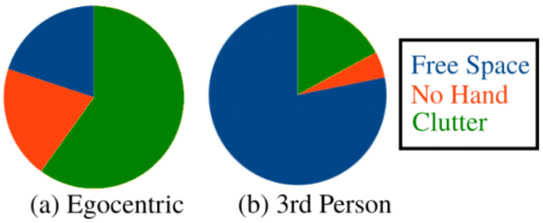


Figure 5: The percent of challenges in CVRA [15] and UCI-EGO-Syn [18] datasets.

In the past 5 years, many CNNs and related studies have been published for HPE. However, most of these studies were evaluated on the MSRA, NYU, ICVL datasets, the results are shown in table 2 of [11]. In this paper, we are interested in estimating 3D hand pose from the 3D annotation of hand skeleton data. The method we use is V2V, it has the input data of 3D annotation data and the point cloud data of hand. In [19], the author proposed 3D CNN for 3D HPE. This network projects 3D points of the hand following: x-direction, y-direction, z-direction. Synthesized in these three directions is encoded as 3D volumes storing the projective Directional Truncated Signed Distance Function (D-TSDF). Special, it only uses three 3D convolutional layers and three fully-connected layers to train the model. The estimated results have an average error of 9.58mm on the MSRA dataset.

In [20], the author proposed a deep regression network (SHPR-Net) for 3D HPE. This network consists of two components: A semantic segmentation network (SegNet) and the hand pose regression network (RegNet). The first component is used to segment the joints, the parts of the hand. That is, each part of the hand is segmented and labeled, RegNet is used to predict the 3D coordinates of the match corresponding to the segmented hand data areas. The estimated results have an average error of 10.78mm on the NYU dataset. In [21], the author proposed Hand PointNet to estimate 3D hand pose from the segmented hand on a depth image by using random decision forest [13], then convert to point cloud data using the Eq. 1. This method has improved the basic PointNet by using a hierarchical PointNet to generate the hierarchical feature extraction. Specifically, it uses three point set abstraction levels. Besides, In [22] and [23], the authors proposed the 3D DenseNet, Point-to-Point Net, respectively. The estimated error of result on the above methods is usually less than 10mm.

# 3 Standardized UCI-EGO Dataset and 3D HPE by V2V

## 3.1 Standardized UCI-EGO Dataset

As shown in Figure 4(b), the annotation data of the UCI-EGO dataset needs to be calibrated to meet the evaluation of 3D HPE in 3D space / on the point cloud data. Deviation in 3D annotation data of the hand pose is due to the 3D annotation data generated from the semi-automatic labeling tool. This issue is covered in Sec. 4.1 by UCI-EGO dataset introduction. This process is done as follows: The

[2] https://www.dropbox.com/s/c91xvevra867m6t/cvpr15_MSRAHandGestureDB.zip?dl=0.
[3] https://jonathantompson.github.io/NYU_Hand_Pose_Dataset.htm.
[4] https://labicvl.github.io/hand.html

coordinates of each keypoint $K(x, y, z)$ in the UCI-EGO annotation data, where $(x, y)$ is the coordinate of $K$ on the depth image, $z$ is the depth value of $K$ in the depth image [3]. In this paper, we replace the depth value of the $K$ point in UCI-EGO [3] with the depth value of the point with coordinates $(x, y)$ on the depth image. However, there are many cases of data loss or missing in the depth images, especially on the depth image sensors collected on previous depth sensors such as the Microsoft Kinect Version 1 [24]. We solve this problem by using the mean depth value of the k-neighbors of the $K$ on the depth image, where $k = 3, d_K = mean(d_{K1}, d_{K2}, d_{K3})$.

After that, the coordinates $(x, y)$ of the keypoint on the image combined with the depth value to generate one point $(x_a, y_a, z_a)$ in 3D space / point cloud data [25] according to Eq. 1.

$$\begin{aligned} x_a &= \frac{(x - cx_d) * z}{fx_d} \\ y_a &= \frac{(y - cy_d) * z}{fy_d} \\ z_a &= z \end{aligned} \tag{1}$$

where $fx_d, fy_d, cx_d$, and $cy_d$ the intrinsics of the depth camera.

The results of the standardized annotation data are illustrated on the point cloud data of the hand as shown in Figure 4(c).

### 3.2 3D HPE by V2V

Based on the results of the 3D HPE of the 3D CNN on ICVL, NYU, MSRA datasets. The V2V [8] network has the best estimation result (average of 3D joints error is 6.28mm, 8.42mm, 7.49mm, respectively). Therefore, we used the V2V network to evaluate HPE on the UCI-EGO dataset. The execution process is shown below.

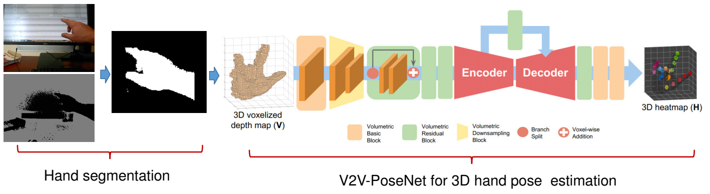

3D voxelized depth map (V)
Encoder
Decoder
3D heatmap (H)
Volumetric Basic Block
Volumetric Residual Block
Volumetric Downsampling Block
Branch Split
Voxel-wise Addition
Hand segmentation
V2V-PoseNet for 3D hand pose  estimation

Figure 6: The hand data segmentation and 3D HPE based on 3D CNN (V2V) [8].

As shown in Figure 6 (left), the hand is in the complex scene, where the input data of V2V is the point cloud data of the segmented hand. Therefore, we propose a preprocessing step to segment the hand data with the environment and other objects.

We based on the annotation data on the keypoints of the hand skeleton frame on the depth image to crop a region container on the depth image with a rectangle that is bounding box of the keypoints on the depth image. We then find the maximum depth value of the keypoints $M_d^h$ on the depth image. We rely on the context of capturing data from the EGO-VI sensor, the hand that is usually closest to the sensor. Therefore, the data near the hand that is large $M_d^h$ is not part of the hand data.

As shown in Figure 6 (right), the input of the V2V method is 3D voxelized data. Thus, it reprojects each pixel of the depth map to the 3D space. After that, this space is discretized based on the pre-defined voxel size. V2V-PoseNet is based on the hourglass model [26] 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. The location of the first volumetric basic block is in the first and last parts of the network. The second volumetric residual block extended from the 2D residual block in [27]. The third volumetric downsampling block is similar to the 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.

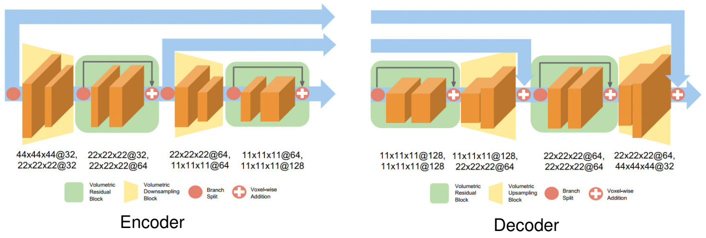


Figure 7: Encoder and decoder phase in the network architecture of V2V-PoseNet [8].

Each phase of the V2V-PoseNet method in Figure 6 is shown in Figure 7. Each phase consists of four blocks as shown in Figure 7. Therein, the volumetric downsampling block reduces the feature map space while the volumetric residual bock increases the number of channels in the encoder phase. Otherwise, the volumetric upsampling block enlarges the feature map space. When upsampling, to compress the extracted features the network reduce the number of channels. To predict each keypoint of the hand in 3D space through two stages: encoder, decoder. They are connected by the voxel-wise. To supervise the per-voxel likelihood in the estimating process, V2V generates a 3D heatmap, wherein the mean of the Gaussian peak is positioned at the ground-truth joint location in Eq. 2.

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

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

$$L = \sum_{n=1}^{N} \sum_{i,j,k} ||D_n^*(i, j, k) - D_n(i, j, k)||^2 \tag{3}$$

where $D_n^*$ and $D_n$ are the ground-truth and estimated results for $n^{th}$ keypoint, respectively, and the number of keypoints is denoted $N$.

## 4 Experimental Results

### 4.1 Dataset

In this paper, we are trained and tested on the UCI-EGO [14] dataset. It provides about 400 frames prepared the 3D annotation. 3D annotations of keypoints with 26 joint points are also provided. To annotate this dataset for evaluating 3D HPE and hand tracking the authors developed a semi-automatic labeling tool. It can annotate the accurate partially occluded hands and fingers in the 3D space by using the techniques: A few 2D joints are first manually labeled in the image and have used to select the closest synthetic samples in the training set; After that, a full hand pose is generated combining the manual labeling and the selected 3D sample; This pose is manually refined, resulting to the selection of a new sample, and the creation of a new pose; This process is repeated until acceptable labeling is achieved. This dataset captured from a chest-mounted Intel Senz3D RGB-D camera/EGO-VI and capture 4 sequences, 2 for each subject (1 male and 1 female), as illustrated in Figure 8. The authors labeled the keypoints of any visible hand in both RGB and Depth images every 10 frames, as illustrated in Figure 9.

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.

Before performing 3D HPE from the point cloud data, we propose a pre-processing step to segment the hand from the complex scene data, as shown in Figure 14. 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.

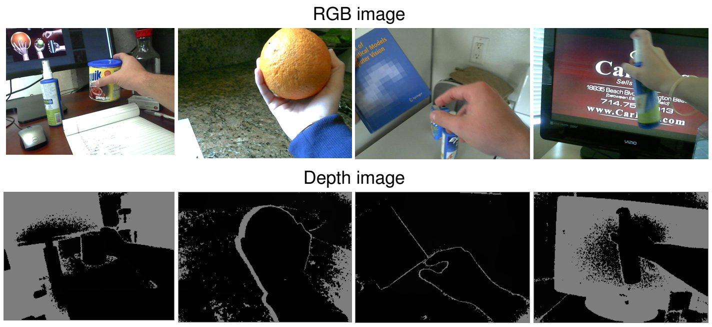


Figure 8: Illustration of hand grasping object in the UCI-EGO dataset [14].

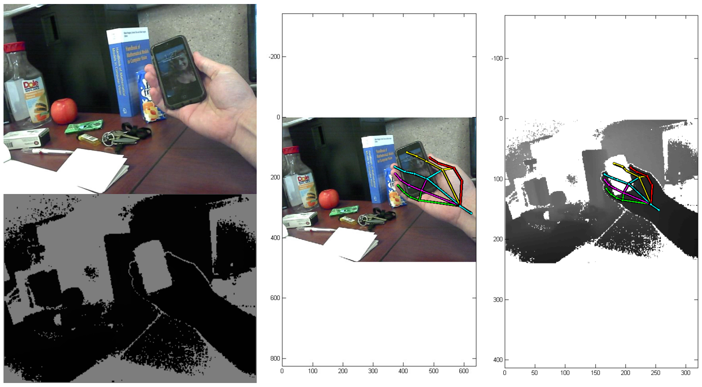


Figure 9: Illustration of 2D hand pose ground truth on the RGB and depth images.

## 4.2 V2V Parameters

Like in the original study of V2V-PoseNet [8], 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. This is the maximum value that V2V can train the model on our computer. The size of input is $88 \times 88 \times 88$. This deep network also uses the optimizer method of Adam [28]. 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 [5].

In the original V2V study [8], the model trained only for 10 epochs. Although, the depth data value on the MSRA dataset is from 0.3m-0.7m, while the depth data value on the UCI-EGO dataset is from 0.3-0.45m. In each epoch, we computed the total loss function of batch sizes by Eq. 4.

$$T_L = \sum_{j=1}^{N_p} L_j \tag{4}$$

where $N_p = N_s/batch\_size$ is the batch size number of the training data, $N_s$ is the sample number of the training data. $L_j$ is the total loss function $j^{th}$ of each batch size.

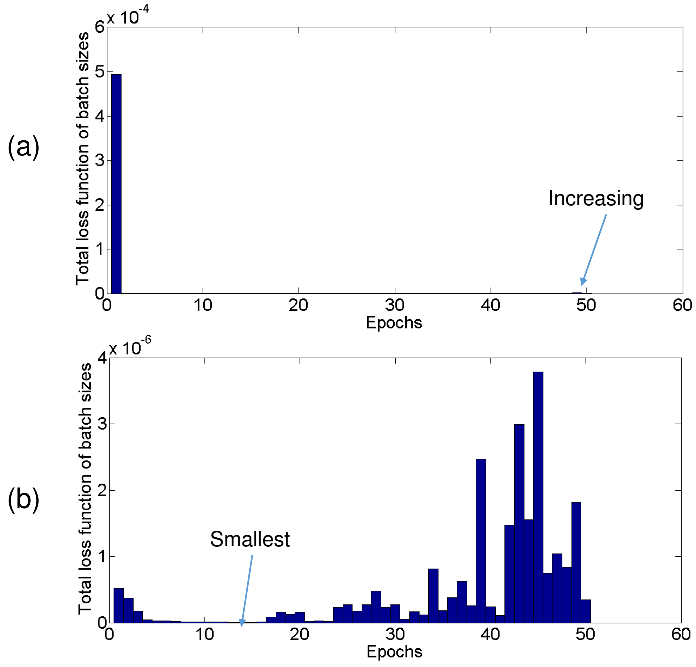


Figure 10: The total loss function of batch sizes at each epoch. (a) is the total loss function of batch sizes of training data when hand data is segmented on the depth image. (b) is the total loss function of batch sizes of validation data when hand data is segmented on the depth image.

We then compare the total loss function at each epoch. In a model with the smallest total loss function, that model is the best model for estimating the 3D hand pose. During training, we found that the total value of loss function up to the epoch $15^{th}$ does not

[5] https://github.com/dragonbook/V2V-PoseNet-pytorch.

decrease any more. We have trained the estimation model for 50 epochs, the total loss function of batch sizes at each epoch are shown in Figure 10. It can be seen that the value of the loss function decreases drastically and is about $10^{-11}$ at the epoch $10^{th}$. Specifically, on the validation data, there is the value of the smallest loss function at the epoch $10^{th}$ as Figure 10(b). Therefore in this paper, we train only 15 epochs.

## 4.3 Evaluation Measure

As the evaluations of the previous 3D HPE method, we used the average 3D distance error (as shown in Eq. 5) to evaluate the results of the 3D HPE on the dataset.

$$\widehat{Err_a} = \frac{1}{Num_s} \sum_{n=1}^{Num_s} \frac{1}{21} \sum_{k=1}^{21} DIS(p_g, p_e) \tag{5}$$

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 11.

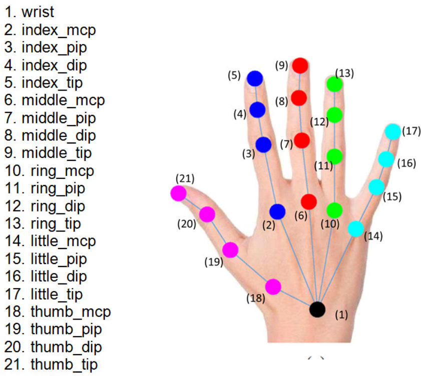
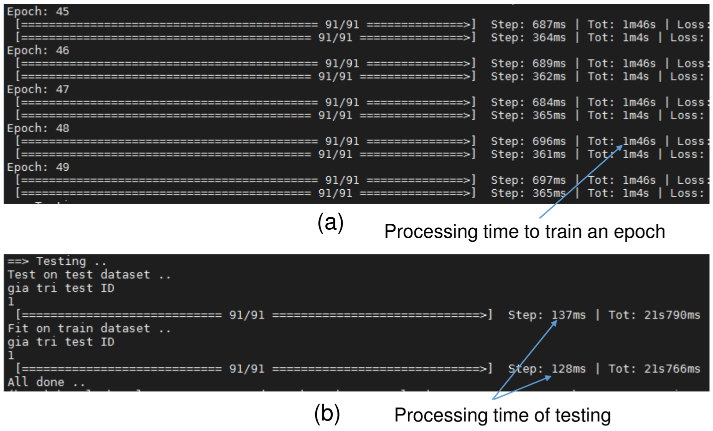


Figure 11: Illustrating the hand joints of the UCI-EGO [14] dataset.

In this paper, we use the rate at 1:5, which means 80% for training and 20% for testing. This ratio is based on the division of [15](using 5-fold cross-validation for testing and training). This means the UCI-EGO dataset uses 283 samples for training and 71 samples for testing. Although, the UCI-EGO dataset provides 26 annotation points. However, we only use 21 annotation data points to evaluate the 3D HPE. The order of points is shown in Figure 11.

## 4.4 Results and Discussions

As the evaluation of 3D HPE results shown in Tab. 2 of [11], also use the 3D distance error (mm) to evaluate the estimation results on the UCI-EGO dataset. The average 3D distance error is shown in Table 1. The processing time of training process and 3D HPE process is shown in Figure 12. As figure 12, the processing time to train for 50 epochs is $1.472h$ and $0.442h$, it is calculated by $(1m46s = 106s) * 50epochs = 1.472h$ and $(1m46s = 106s) * 15epochs = 0.442h$, respectively. The processing time for testing is shown in Table 2. It is calculated by $128ms/362samples$ and $137ms/362samples$, respectively.



```
Epoch: 45
 [===================================== 91/91 ===============>]  Step: 687ms | Tot: 1m46s | Loss:
 [==================================== 91/91 ================>]  Step: 364ms | Tot: 1m4s | Loss:
Epoch: 46
 [===================================== 91/91 ===============>]  Step: 689ms | Tot: 1m46s | Loss:
 [==================================== 91/91 ================>]  Step: 362ms | Tot: 1m4s | Loss:
Epoch: 47
 [===================================== 91/91 ===============>]  Step: 684ms | Tot: 1m46s | Loss:
 [==================================== 91/91 ================>]  Step: 365ms | Tot: 1m4s | Loss:
Epoch: 48
 [===================================== 91/91 ===============>]  Step: 696ms | Tot: 1m46s | Loss:
 [==================================== 91/91 ================>]  Step: 361ms | Tot: 1m4s | Loss:
Epoch: 49
 [===================================== 91/91 ===============>]  Step: 697ms | Tot: 1m46s | Loss:
 [==================================== 91/91 ================>]  Step: 365ms | Tot: 1m4s | Loss:
```

```
==> Testing ..
Test on test dataset ..
gia tri test ID
1
 [========================== 91/91 ==============================>]  Step: 137ms | Tot: 21s790ms
Fit on train dataset ..
gia tri test ID
1
 [========================== 91/91 ==============================>]  Step: 128ms | Tot: 21s766ms
All done ..
```

Figure 12: Illustrating the processing time to train the estimation model and testing 3D HPE. (a) is the processing time to train an epoch. (b) is the processing time for testing.

Table 1: The average 3D distance error of the V2V on the UCI-EGO dataset for 3D HPE when trained through 15 epochs and 50 epochs.

| **Training** | **Measuremet/ Method** | | **V2V** |
|---|---|---|---|
| **15 epochs** | **Average of 3D joints error $Err_a$(mm)** | **Hand segmentation** | **87.52** |
| | | **No hand segmentation** | 95.49 |
| **50 epochs** | **Average of 3D joints error $Err_a$(mm)** | **Hand segmentation** | **87.07** |
| | | **No hand segmentation** | 88.73 |

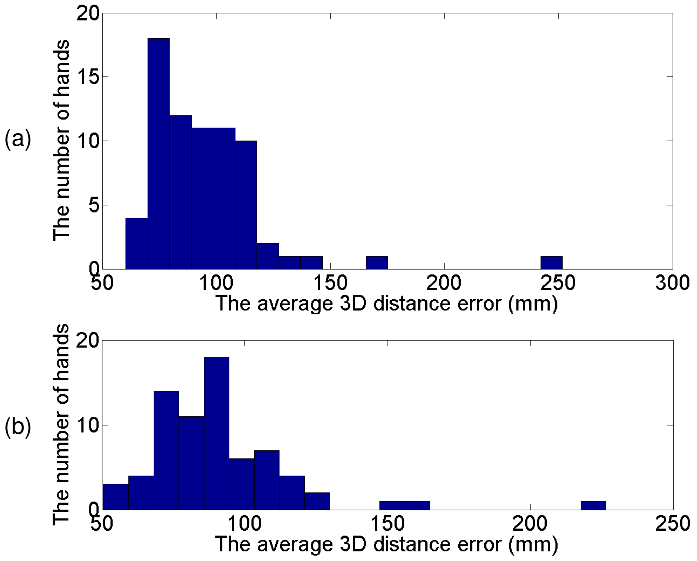


Figure 13: The distribution of 3D joints error for 3D HPE based on the UCI-EGO dataset by V2V-PoseNet [8] when trained through 15 epochs. (a) The distribution of 3D joints error when using hand segmentation on the depth image; (b) The distribution of 3D joints error when do not use hand segmentation on the depth image.

Table 2: The average of processing time of the V2V on the UCI-EGO dataset to estimate a 3D hand pose.

| **Measurement/ Method** | | **V2V** |
|---|---|---|
| **Processing time (ms)/ hand** | **Hand segmentation** | **0.35** |
| | **No hand segmentation** | 0.38 |

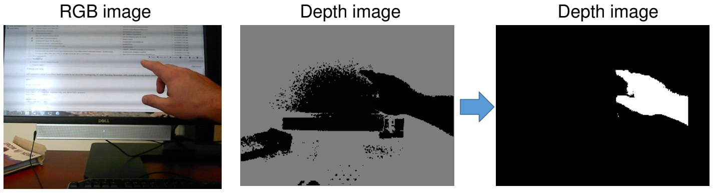


Figure 14: Illustrating the data well segmented of hand in the complex scene.

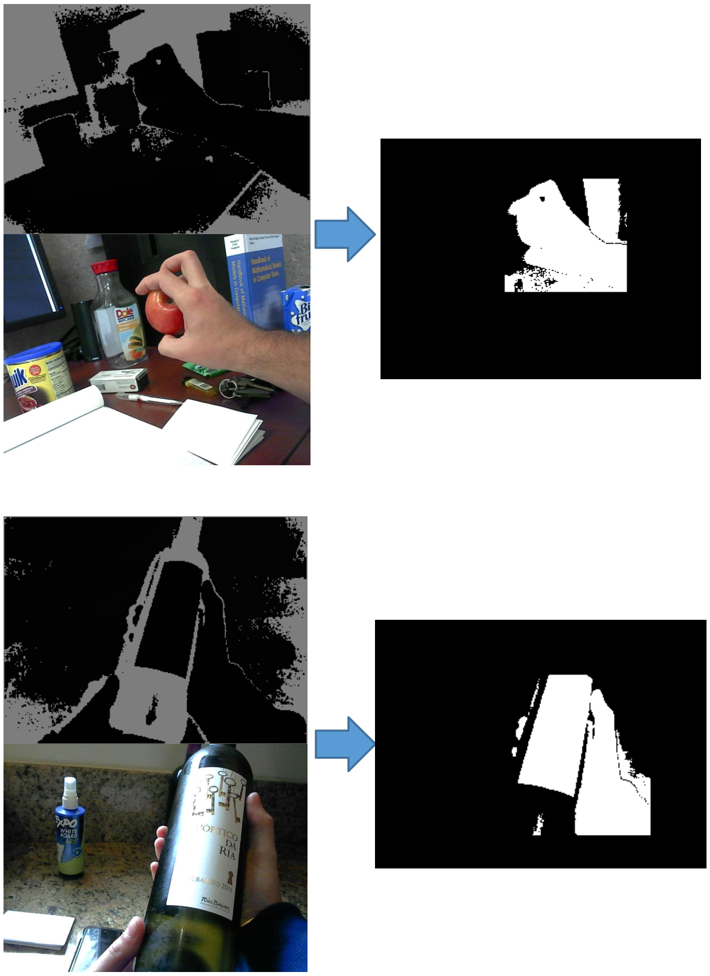


Figure 15: Illustrating of the hand data and the objects in the UCI-EGO dataset.

From table 2 of research in [11] and Table 1, the error results of estimating the 3D hand pose on the EGO-VI dataset are much larger than those estimated on the obtained datasets from a fixed number of perspectives. The time of estimated hand joints when using V2V is enormous, as shown in Table 2. This high estimate time due to carrying CNN using the input data is the point cloud data that is not reduced by number of points.

As Table 1 and Figure 13, the 3D HPE results when using the hand data segment are better when not using the hand data segment in the complex scenes. The error distribution when using the hand data segment concentrated in bins closer to 0.

Although when segmenting the hand data, the estimation results were better than when the hand was not segmented. However, the estimated results have not improved much. Since in the UCI-EGO dataset only about 9% of the hands are well segmented with other subject's data, as illustrated in Figure 14. The remaining about 90% of the hands are grasping objects like phone, book, spray bottle, bottle, ball, etc. Therefore, the hand data gets stuck with the data of the objects being handled, as illustrated in Figure 15.

Figure 16 shows the results of estimating 3D hand pose on the point cloud data in two methods: The segmented hand data (a) and no segmented hand data (b).

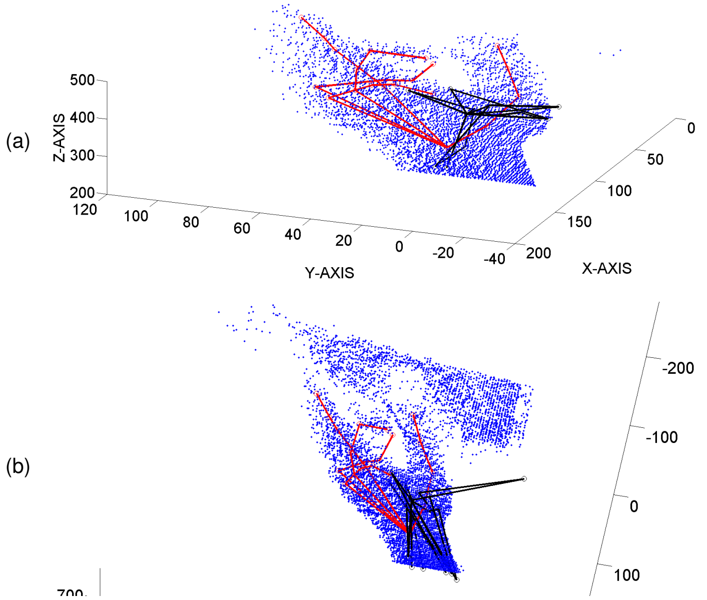


Figure 16: The results of the estimated 3D hand pose on the point cloud. (a) is the result of estimating the 3D hand pose in the 3D space on the segmented hand data; (b) is the result of estimating the 3D hand pose in the 3D space on hand data is not segmented. The blue points are the point cloud of hand and others object. The red skeleton is the ground truth of 3D hand pose, the black skeleton is the estimated 3D hand pose.

Based on the reading paper of [29], we find that the error of estimating the 3D hand pose on the UCI-EGO-Syn [18] dataset is high with: Hough [30], RDF [31], Deep Prior [32], PXC [33], Cascader [14], EGO.WS. [3]. The error distribution is from 35 to 100mm, as illustrated in Figure 18.

The UCI-EGO dataset has the hand data that performs grasping objects and is collected from an EGO-VI mounted on the person, the hand data is obscured, as shown in Figure 15. In particular, the data of the fingers is obscured. In this paper, we used V2V for estimating 21 joints of hand (3D hand pose), the input data of V2V is the coordinate of 21 joints in the 3D space of ground truth data, the output is also 21 joints, as illustrated in Figure 6. Therefore, the fields of the obscured fingers, the hand joints can still be estimated.

However, the point cloud data of the hand is much missing, the estimation results have large errors, as illustrated in Figure 17.

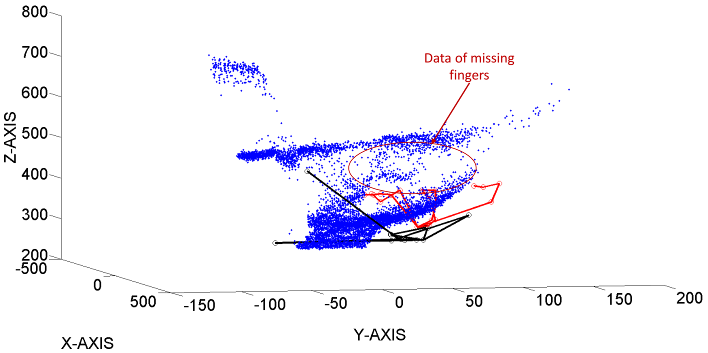


Figure 17: Illustrating of estimation results on the data with obscured fingers of Figure 9. The red skeleton is the ground truth of the 3D hand pose, the black skeleton is the estimated 3D hand pose.

During the research on 3D HPE, we found the challenges as follows:

- ***The high degree of freedom***: In realistic/3D space human hand models have between 15 and 24 degrees of freedom [34]. From 21 joints there are about 63 coordinates of in the 3D space. To train such a large dimension vector requires a very strong model and very large learning data to train model.
- ***Data obscured***: As shown above, the hand data is obscured making the hand's point cloud data missing. This makes the 3D HPE result from a high error value. This problem can be seen when comparing the results estimated on datasets with complete hand data (MSRA, ICVL, NYU) (table 2 of [11]) with the estimation results on the UCI-EGO-Syn [18] dataset (Figure 18). The quality of the depth images is also an issue affecting the 3D data / point cloud data. The depth images can be collected from a stereo camera or ToF (Time of Flight), this data still contains error noise.
- ***Hand size in the space***: When moving in the 3D space to perform operations, the size of the hand is constantly changing. To train hand estimation models with different sizes, a large number of samples and strong models are required.
- ***3D hand pose annotation***: The quality of the 3D HPE model depends on the training data. To prepare the training data requires an expensive system like in the FPHA (First-Person Hand Action) dataset [17], or use the estimated data through an estimation model like in the Ho-3D dataset [35].

## 5 Conclusion

To perform complex operations such as human hands then robot hand operations need to be built based on human hand operation. To do this, the poses of the human hand need to be accurately estimated and restored in 3D space. In this paper, we perform the standardization of the UCI-EGO dataset to evaluate 3D HPE. Simultaneously, we retrained, evaluated 3D HPE, and presented the results on the point cloud data based on the V2V-PoseNet. The estimation results on the UCI-EGO dataset have been a large error and many challenging. The estimation results above also show that estimating 3D hand pose on the EGO-VI dataset is challenging and needs to be studied to improve the accuracy of 3D HPE.

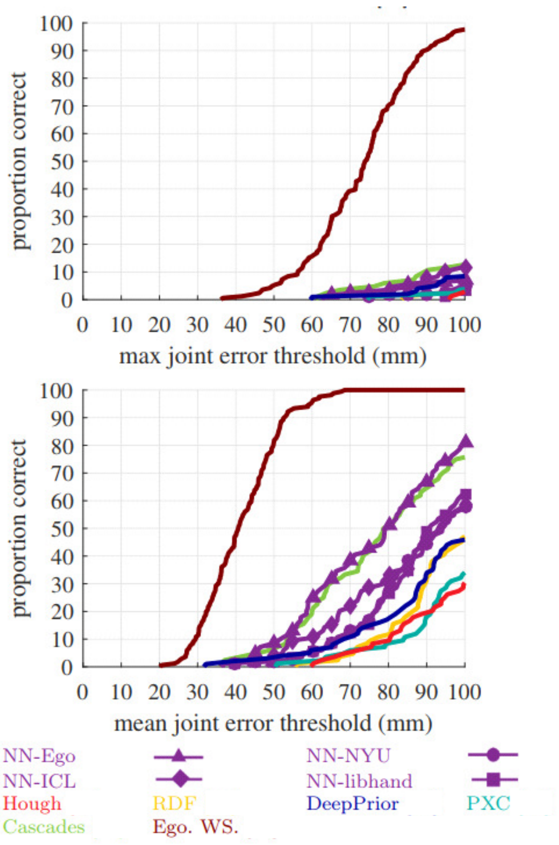


Figure 18: Distribution of 3D joints error estimation [29] on the UCI-EGO-Syn [18] dataset.

## Acknowledgement

This research is funded by Tan Trao University in Tuyen Quang, Vietnam.

## References

[1] S. Anasua, Y. Yang, V. Mauno, "Variation benchmark datasets: update, criteria, quality and applications," Database, Volume 2020, 2020, baz117, https://doi.org/10.1093/database/baz117, 2020.

[2] J. Brownlee, "A Gentle Introduction to k-fold Cross-Validation," `https://machinelearningmastery.com/k-fold-cross-validation/`, [Accessed 1 September 2020].

[3] G. Rogez, J. S. Supanvcivc, D. Ramanan, "First-person pose recognition using egocentric workspaces," in Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, volume 07-12-June, 4325–4333, 2015, doi:10.1109/CVPR.2015.7299061.

[4] R. Gregory, S. S. James, R. Deva, "Egocentric Pose Recognition in Four Lines of Code," https://arxiv.org/pdf/1412.0060.pdf, 2014.

[5] S. S. James, R. Grégory, Y. Yi, S. Jamie, R. Deva, "Depth-based hand pose estimation: data, methods, and challenges," International Journal of Computer Vision, **126**, 1180–1198, 2015.

[6] S. Yuan, Q. Ye, B. Stenger, S. Jain, 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, volume 2017-Janua, 2605–2613, 2017, doi:10.1109/CVPR.2017.279.

[7] M. Hutson, "Watch a robot hand learn to manipulate objects just like a human hand," https://www.sciencemag.org/news/2018/07/watch-robot-hand-learn-manipulate-objects-just-human-hand, [Accessed 1 September 2020].

[8] G. Moon, J. Y. Chang, 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, 5079–5088, 20118.

[9] R. Li, Z. Liu, J. Tan, "A survey on 3D hand pose estimation: Cameras, methods, and datasets," Pattern Recognition, **93**, 251–272, 2019, doi:10.1016/j.patcog.2019.04.026.

[10] H. Su, S. Maji, E. Kalogerakis, E. Learned-Miller, "Multi-view Convolutional Neural Networks for 3D Shape Recognition," in Proc. ICCV, 264–272, 2015, doi:10.1109/CVPR.2018.00035.

[11] C.-h. Yoo, S.-w. Kim, S.-w. Ji, Y.-g. Shin, S.-j. Ko, "Capturing Hand Articulations using Recurrent Neural Network for 3D Hand Pose Estimation," https://arxiv.org/abs/1911.07424, 2019.

[12] J. Tompson, M. Stein, Y. Lecun, K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," ACM Transactions on Graphics, **33**(5), 2014.

[13] D. Tang, H. J. Chang, A. Tejani, T. K. Kim, "Latent regression forest: Structured estimation of 3D hand poses," IEEE Transactions on Pattern Analysis and Machine Intelligence, **39**(7), 1374–1387, 2017, doi:10.1109/TPAMI.2016.2599170.

[14] G. Rogez, M. Khademi, J. S. Supanvcivc, J. M. Montiel, D. Ramanan, "3D hand pose detection in egocentric RGB-D images," in Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), volume 8925, 356–371, 2015, doi:10.1007/978-3-319-16178-5_25.

[15] M. Oberweger, G. Riegler, P. Wohlhart, 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, volume 2016-Decem, 4957–4965, 2016, doi:10.1109/CVPR.2016.536.

[16] S. Sridhar, F. Mueller, M. Zollhoefer, D. Casas, A. Oulasvirta, 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), 2016.

[17] G. Garcia-Hernando, S. Yuan, S. Baek, T.-K. Kim, "First-Person Hand Action Benchmark with RGB-D Videos and 3D Hand Pose Annotations," .

[18] S. S. James, R. Gregory, Y. Yi, S. Jamie, R. Deva, "Depth-based hand pose estimation: methods, data, and challenges," International Journal of Computer Vision, Springer Verlag, 2018, **126**(11), 1180-1198, 2018. doi: 10.1007/s11263-018-1081-7, 2018.

[19] L. Ge, H. Liang, J. Yuan, 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, doi:10.4324/9781315556611.

[20] X. Chen, G. Wang, S. Member, C. Zhang, K. I. M. Member, X. Ji, "SHPR-Net : Deep Semantic Hand Pose Regression From Point Clouds," IEEE Access, **PP**(c), 1, 2018, doi:10.1109/ACCESS.2018.2863540.

[21] L. Ge, Y. Cai, J. Weng, J. Yuan, "Hand PointNet : 3D Hand Pose Estimation using Point Sets," Cvpr, 3–5, 2018.

[22] L. Ge, H. Liang, J. Yuan, S. Member, D. Thalmann, "Real-time 3D Hand Pose Estimation with 3D Convolutional Neural Networks," IEEE Transactions on Pattern Analysis and Machine Intelligence, **8828**(c), 2018, doi:10.1109/TPAMI.2018.2827052.

[23] L. Ge, Z. Ren, J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in European Conference on Computer Vision, volume 11217 LNCS, 489–505, 2018, doi:10.1007/978-3-030-01261-8_29.

[24] D. Abdul, H. Ammar, "Recovering Missing Depth Information from Microsoft Kinect," pdfs.semanticscholar.org, 2011.

[25] N. Burrus, "Kinect Calibration," http://nicolas.burrus.name/index.php/Research/KinectCalibration, [Accessed 25 July 2020].

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

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

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

[29] J. S. Supancic, G. Rogez, Y. Yang, J. Shotton, D. Ramanan, "Depth-Based Hand Pose Estimation: Methods, Data, and Challenges," International Journal of Computer Vision, **126**(11), 1180–1198, 2018, doi:10.1007/s11263-018-1081-7.

[30] X. C., C. L., "Efficient Hand Pose Estimation from a Single Depth Image." in International Conference on Computer Vision (ICCV), 2013.

[31] K. C., Kırac, K. F., Y. E., A. L., "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in International Conference on Computer Vision (ICCV), 2012.

[32] M. Oberweger, P. Wohlhart, V. Lepetit, "Hands Deep in Deep Learning for Hand Pose Estimation," in Computer Vision Winter Workshop, 2015.

[33] Intel, "Perceptual computing SDK," 2013.

[34] S. Cobos, M. Ferre, R. Aracil, "Simplified Human Hand Models Based On Principal Component Analysis," in IFIP Conference on Human-Computer Interaction, 610–615, 2010.

[35] S. Hampali, M. Rad, M. Oberweger, V. Lepetit, "HOnnotate: A method for 3D Annotation of Hand and Objects Poses," https://arxiv.org/abs/1907.01481, 2019.