Empirical study of 3D-HPE on HOI4D egocentric vision dataset based on deep learning

Van-Hung Le [a,1,*]

[a] Information Technology Department, Tan Trao University, k6, Trung Mon, Yen Son, Tuyen Quang, 22000, Vietnam
[1] van-hung.le@mica.edu.vn
* corresponding author

ARTICLE INFO

Article history
Received October 7, 2023
Revised March 16, 2024
Accepted March 23, 2024
Available online May 31, 2024

Keywords
3D Hand pose estimation
HOI4D dataset
Egocentric vision
Convolutional neural networks(CNNs)
Deep learning

ABSTRACT

3D hand pose estimation (3D-HPE) is one of the tasks performed on data obtained from egocentric vision camera (EVC) such as hand detection, segmentation, and gesture recognition applied in fields such as HCI, HRI, VR, AR, Healthcare, supporting for the visually impaired people, etc. In these applications, hand point cloud data obtained from EV is not very challenging due to being obscured by gaze direction and other objects. Our paper performs a comparative study on 3D right-hand pose estimation (3D-R-HPE) from the HOI4D dataset with four cameras used to collect and animate the dataset. This is a very challenging dataset and was published at CVPR 2022. We use CNNs (P2PR PointNet, Hand PointNet, V2V-PoseNet, and HandFoldingNet - HFNet) to fine-tune the 3D-HPE model based on the point cloud data (PCD) of hand. The resulting error of 3D-HPE is presented as follows: P2PR PointNet (average error (Erra) is 32.71mm), Hand PointNet (average error (Erra) is 35.12mm), V2V-PoseNet (average error (Erra) is 26.32mm), and HFNet (average error (Erra) is 20.49mm). HFNet is the latest CNN (in 2021) with the best results. This estimation error is small and can be applied and modeled to automatically detect, estimate, and recognize hand pose from the data obtained by EV. The average processing time is 5.4fps when done on the GPU of the HFNet, which is the fastest. Detailed quantitative and qualitative results were presented that are beneficial to various applications such as human-computer interaction, virtual and augmented reality, and healthcare, particularly in challenging scenarios involving occlusions and complex datasets.

1. Introduction

3D-HPE is applied in many applications: robotics, human-machine interaction (HCI) [1], human-robot interaction (HRI) [2], virtual reality (VR) [3], augmented reality (AR), Healthcare [4], supporting the visually impaired people in moving and grasping objects [5], [6], etc. This problem can be solved with input data types such as color images, depth images(DI), PCD, or a combination of two or more information from the RGB-D sensor.

Previously, most studies performed 3D-HPE on the obtained data from second and third-person cameras (surveillance cameras). To build systems to help the blind or robotic, often use traditional machine learning models, and deep learning (DL) to train the collected data from bearing sensors (EVC). Previously, we have done some research on 3D-HPE on the collected data from EVC as FPHAB (First-Person HA Benchmark) [7]. However, the FPHAB dataset does not describe all actual cases of the hand in the real environment, there are many changes. In survey research [8], there are three types of input data for the 3D-HPE model: color image, DI, and PCD. The CNN models estimate 3D-HP based on PCD to produce real-world-like and intuitive results that resemble coordinate systems with data obtained from EVC. In particular, Bandini *et al.* [6] survey research presented and analyzed most of the applications of EVC and research according to applications.

Today, the advent of DL has had many impressive results with computer vision problems, in which CNN is a typical approach for solving estimation problems. There have been many studies using CNNs (Convolutional Neural Networks) to estimate 3D-HP [9]–[11]. These studies often use existing hand models, DI as input, or color images from multiple views of the hand, so the results depend heavily on the existing hand model or complex data.

Recently, the HOI4D dataset was published to the research community for HPE and hand action (HA) recognition. The HOI4D dataset is an EVC dataset published at the prestigious conference CVPR 2022 [12]. Currently, there are very limited studies on HPE based on this data.

Our paper performs a study comparing 3D-HPE with PCD input for the most advanced CNNs. The CNNs used in comparative research are Point-to-Point Regression PointNet (P2PR PointNet) [13], Hand PointNet [13], V2V-PoseNet [14], HFNet [15] on the right-HAs of the HOI4D dataset. These are advanced DL that have very convincing results in 3D-HPE on first-person and second- person camera datasets. Before evaluating the estimation process, we performed the normalization of the 3D-R-HP annotation and divided it into training data and testing data. The estimated results are presented, analyzed in detail, and shown by each camera and each type of right-HA.

Our main contributions include:

- We normalize the 3D-R-HP annotation according to the published 3D-HP annotation and divide the data into training and testing sets of the HOI4D dataset.
- We standardize and build the 3D PCD area of the hand to limit the data as input for 3D-HPE based on the original hand pose on the image (2D-HP annotation).
- We fine-tune a 3D-R-HPE model with the input data being 3D PCD of the hand and 3D-HP annotation of the HOI4D based on P2PR PointNet, Hand PointNet, V2V-PoseNet, and HFNet.
- We tested and compared the 3D-R-HPE results of the trained model of P2PR PointNet, Hand PointNet, V2V-PoseNet, and HFNet on the HOI4D dataset. The results are analyzed and further application of 3D-HPE is discussed.

Our paper is organized as follows. In Sec. 2, the related studies to 3D-HPE are presented. In Sec. 3, we present CNNs for 3D-HPE from 3D point cloud data. The dataset and experimental results on the HOI4D dataset will be presented in Sec. 4. Sec. 5 presents conclusions and research in short-term and long-term plans.

2. Related Works

The problem of estimating HP on data obtained from EVC is new today. Therefore, we present related research problems based on two directions: the first is about 3D-HPE, and the second is about data obtained from EVC.

The problem of 3D-HPE has been of interest in research for the past 10 years. There have been many valuable studies and surveys published on this issue [8], [16], [17]. In the study of Huang *et al.* [16], we also conducted a survey on the challenges of HPE based on two types of images observed from the camera: color images and DI. At the same time, the authors also surveyed methods to estimate 3D-HP from the above two types of data. With the approach based on DI, the approaches are divided into directions: generative methods, regression methods, detection methods, structural constraint methods, multi-stage prediction methods, ensemble prediction methods, synthetic data methods, and other CNNs. With generative approaches, the 3D model of the hand is represented by geometric models and ensures constraint satisfaction: skeleton model, sphere model, triangulated mesh model, cylindrical model, etc. With regression-based methods for 3D-HPE, these methods often use DL to build estimation models in two directions: 3D CNN framework [14] and point-set networks [18]. With detection-based methods, 2D heatmaps and 3D heatmaps are built based on predicting the density of DI, PCD, or voxel data. From this build a dense probability map for each joint [13], [14], [19]. Other methods have been introduced in [16]. With color image input data, the process of 3D-HPE can be performed according to the following methods: generative methods, 2D- to-3D lifting methods, cross-modality methods, disentanglement methods, model-based 3D hand reconstruction methods, model-free 3D hand reconstruction methods, weakly-supervised and semi-supervised learning CNNs methods, sequential modeling, and tracking methods. Ohkawa *et al.* [17], the authors have conducted an exhaustive survey of methods to annotate data for the evaluation of 3D-HPE models; they are classified into 4 types as follows: manual method, synthetic-model-based methods, hand-sensor-based methods, and computational methods. At the same time, this study also introduces several standard datasets over the past 3 years to evaluate 3D hand pose models: GRAB, YouTube3DHands, HO-3D, DexYCB, H2O, InterHand2.6M, AssemblyHands [17]. In this study, some databases collected from EVC were also introduced. However, the HOI4D dataset has not been introduced. Choi *et al.* [20] implement a 3D-HPE method based on limited finger data on the touch screen, then rely on pre-defined hand models to estimate 3D-HP. Drosakis *et al.* [21] proposed a method that uses 2D hand key points of the hand detected on color images, then uses a MANO 3D hand model with a predefined hand shape to optimize and estimate the 3D-HP. Ivashechkin *et al.* [22] proposed using ResNet to detect 2D-HP based on color images or video sequences, and then the MANO model is used to regress and optimize 3D-HP. Cheng *et al.* [11] proposed a combination of virtual hand viewpoint selections on DI using a confidence network and projection into the 3D geometry and 3D space as a PCD. Finally, use FC and Softmax to combine and evaluate the confidence of the viewing directions. Wen *et al.* [23] and Kazakos *et al.* [24] proposed another non-DL approach for 3D-HPE. The authors used a transformer approach based on temporal information of color image frame sequences for estimation.

By 3D-HPE on the EVC dataset, Le *et al.* [25] proposed an end-to-end process for estimating and recognizing hand movements on the FPHAB dataset, where the problem of 3D-HPE is solved using HopeNet [26] combined with GraphCNN, the best result has the smallest distance error of 36.6 mm. Le [27] performed research on automatic 3D-HPE using HFNet with a limited input data area detected by the bounding box (BB) using YOLOv7, The best-estimated error result is 19.98mm when performed on the original data hand area (2D BB annotation). Prakash *et al.* [28] proposed the WildHands dataset for evaluating 3D-HPE. Data collected from EVC includes color images and annotation data collected and marked with the FrankMocap system. Data were collected in a kitchen environment. Plizzari *et al.* [29] presented the applications of data collected from EVC and the conditions for collecting EVC data. Pramanick *et al.* [30] introduced the EgoVLPv2 database collected from EVC to evaluate sign language. Zhang *et al.* [31] proposed the EgoHOS dataset, collected from EVC, including 11,243 color images for object segmentation when hands perform interactions. The original data of this dataset is pixel-level marked. Gong *et al.* [32] proposed the MMG-Ego4D database that includes four information points: PCD, video, audio, and inertial motion sensor modalities. At the same time, creates robustly generalize across modalities based on supervised learning. Khaleghi *et al.* [33] proposed the MuViHand database to evaluate 3D-HPE methods. The dataset was collected and generated using ECV and compiled from 12 ECVs. It includes 402,000 synthetic hand RGB images across 4,560 videos. The ground truth (GT) data provided is 3D-HP labels and annotation with 21 joints by using MIXAMO. Plizzari *et al.* [34] introduced the N-EPIC-Kitchens database to evaluate hand activity recognition models from EVC collected data, its data includes RGB and optical flow. Ohkawa *et al.* [35] proposed a large-scale standard of egocentric activities called AssemblyHands. AssemblyHands includes 3.0M annotated images of 490K egocentric images, especially 3D-HP annotation is built very accurately with an error of only 4.21mm and 85% lower than Assembly101's error. Wang *et al.* [36] proposed a learning method to recognize hand activities obtained from ECV when performing medical and surgical operations. Color image data is built according to the viewing direction and aggregated. The authors use spatiotemporal features to train the activity recognition model. Planamente *et al.* [37] have researched hand activity recognition on data obtained from ECV with EPIC-Kitchens- 55 and EPIC-Kitchens-100 datasets. These are two color image databases that collect data in a kitchen environment with 32 interactive object class labels of hands. Wu *et al.* [38] have researched predicting future HAs based on the ImagineRNN method by learning feature regression, this method has been evaluated on the EPIC Kitchens Action dataset.

3. 3D-HPE based on CNNS

As shown in [8], the input data for 3D-HPE based on CNNs can be color images, DI, and PCD. To select a CNN capable of accurately estimating 3D-HP in the process of building a system to help blind people find and grasp objects in daily life. This helps the system accurately determine the hand position in 3D space and determine the hand size, grasping type, and grasping direction, to perform safe and accurate grasping. Therefore, we performed this empirical to compare and select the best CNN for 3D-HPE. In this paper, we use some typical CNNs for 3D-HPE based on PCD as input, which means CNNs perform direct 3D-HPE from PCD. The flow chart of the comparative study is presented in Fig. 1. Similar to Le [27]'s study, we use limited hand data area by detecting the hands-on color images and taking the corresponding DI, then converting this data area to 3D PCD and as input to CNNs. The output of the estimation model is a 3D-HP, including 21 key points. We perform fine-tuning of the 3D-HPE model on the HOI4D dataset because the HOI4D is a large dataset containing realistic hand-grasping activities. Therefore, the pre-trained model will learn more about the pose characteristics of hand-holding objects in 3D space. Next, we will present the architecture of 3D-HPE by the CNNs.

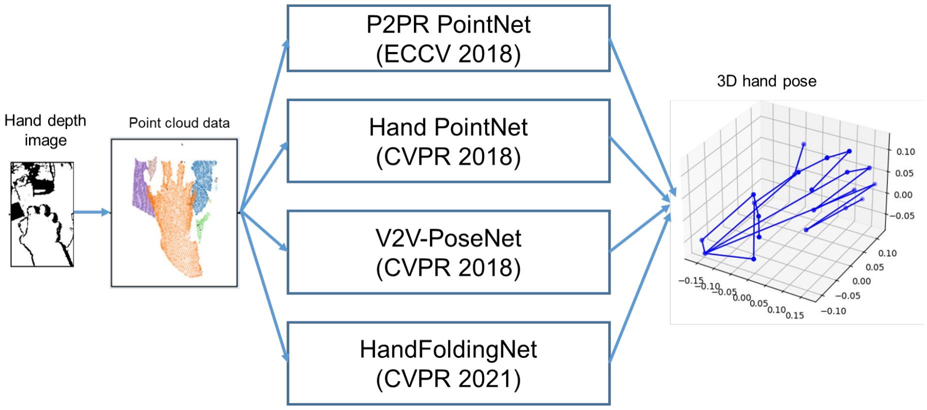

Fig. 1. The flow chart of the comparative study

3.1. Point-to-Point Regression PointNet (P2PR PointNet)

P2PR PointNet [13] is a CNN proposed on ECCV 2018 that uses the hand PCD as the input to regress the 3D-HP/3D key points. The architecture of P2PR PointNet is presented in Fig. 2. The input data is the DI of the hand and is converted to PCD.

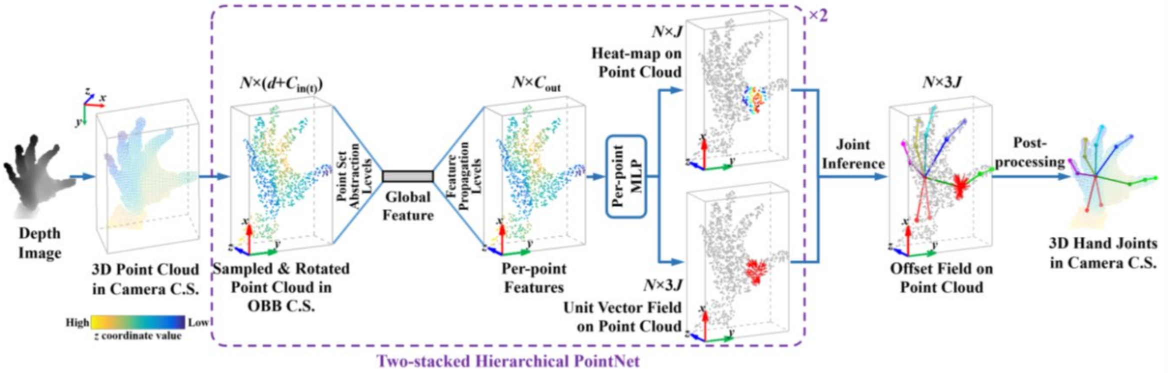
Depth Image
3D Point Cloud in Camera C.S.
High
Low
z coordinate value
N×(d+C_in(t))
Sampled & Rotated Point Cloud in OBB C.S.
Point Set Abstraction Levels
Global Feature
Feature Propagation Levels
N×C_out
Per-point Features
Per-point MLP
N×J
Heat-map on Point Cloud
N×3J
Unit Vector Field on Point Cloud
×2
Joint Inference
N×3J
Offset Field on Point Cloud
Post-processing
3D Hand Joints in Camera C.S.
Two-stacked Hierarchical PointNet
x
y
z

Fig. 2. Point-to-Point Regression PointNet implementation for 3D-HPE [13].

The PCD is first normalized to the camera coordinate system (CS) then it is downsampled to N points. As shown in Fig. 2, the PCD data of the hand is standardized along the y-axis; the direction of the y-axis is the direction from the finger to the wrist in the vertical direction. For the regression network to work better on hands with different orientations, the authors created an OBB (oriented bounding box) and standardized the hands to their one orientation. The direction of the hand is standardized to the x-axis. The 3D coordinates (x, y, z) of 3D hand points are standardized from 0.5 to 0.5 based on the centroid of the 3D PCD on each subtraction level and divided by Lobb. Then, two-stacked hierarchical PointNet (Hi-Po-Net) [19] is applied to regress a 3D heat map on the 3D PCD based on global feature (GF) points extracted from abstraction levels. 3D heatmaps regressed on the input PCD are the unit vector fields, and 3D heatmap corresponds to a 3D hand joint/3D keypoint of 3D-HP. The direction and distance between 3D heatmaps are similar to input points to J hand joints in the real world. To perform 3D heatmaps regression on the per-point feature, P2PR PointNet used the MLP (multilayer perceptron network), as Fig. 3. The 3D HP is then inferred from the 3D heatmaps based on the unit vector fields by using the last Hi-Po-Net module. The final step is post-processing to solve two problems of P2PR PointNet: (1) unreliable estimation in large space and (2) no constraints between hand joints. To solve the first problem, P2PR PointNet uses a threshold when the estimate of a larger 3D heatmap candidate is replaced by the result of directly regressing the (x, y, z) of the HP. To solve the second problem, P2PR PointNet used constraints and estimated the hand pose in more space using PCA. The output is 3D HP in the CS camera.

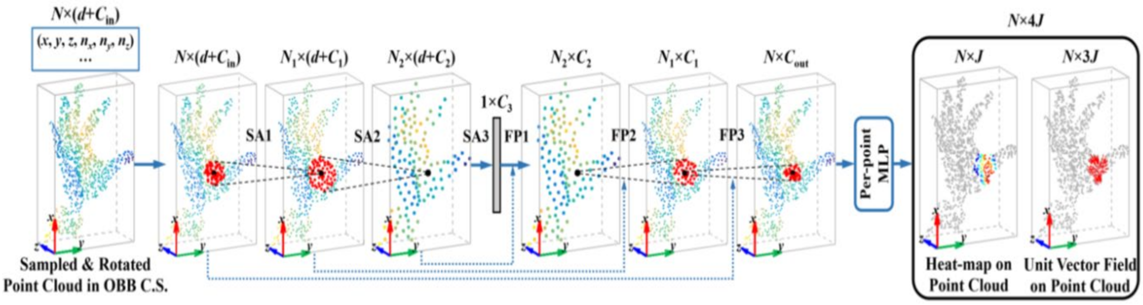

Fig. 3. Illustrating of the first network module to generate heat maps and unit vector field [13].

3.2. Hand PointNet

Hand PointNet [13] is a CNN used to regress the 3D joints of the 3D HP based on PointNet [18], with the input being a DI of the hand, then converted to 3D PCD. The architecture of the Hand PointNet is presented in Fig. 4.

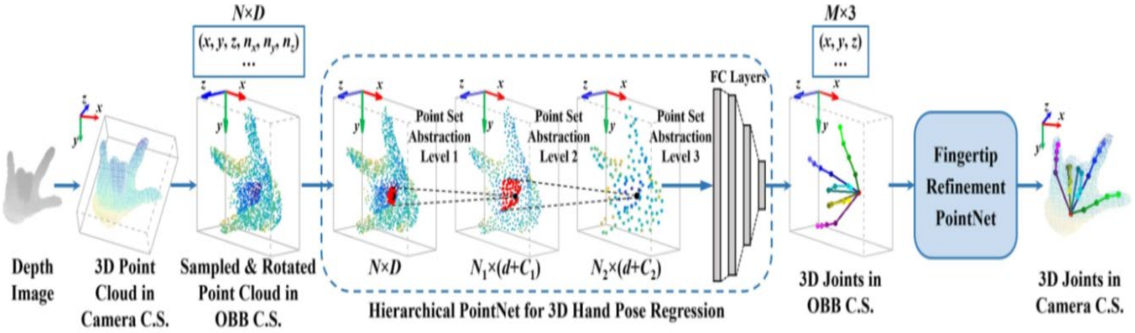
$N\times D$
(x, y, z, n_x, n_y, n_z)
...
$M\times 3$
(x, y, z)
...
FC Layers
Point Set Abstraction Level 1
Point Set Abstraction Level 2
Point Set Abstraction Level 3
Fingertip Refinement PointNet
Depth Image
3D Point Cloud in Camera C.S.
Sampled & Rotated Point Cloud in OBB C.S.
$N\times D$
$N_1\times(d+C_1)$
$N_2\times(d+C_2)$
Hierarchical PointNet for 3D Hand Pose Regression
3D Joints in OBB C.S.
3D Joints in Camera C.S.

Fig. 4. Model of the Hand PointNet [13] network for 3D-HPE with the data input of the DI data region converted to the PCD

Hand PointN*et al*so uses the DI of the hand and then converts it to 3D PCD. 3D PCD in the camera CS of the hand normalized in one direction of the 3D OBB by the PCA. This means that PCD of hands with different orientations are normalized to the same direction. Then the data is extracted and sampled into three abstraction sets with the number of 512, 128, and 64 points, respectively.

Then, Hi-Po-Net, as Fig. 5, is used to estimate the (x, y, z) of the 3D key points (joints) on the three abstraction sets, and the results of the regression are combined through fully-connected layers to estimate the locations (x, y, z) of the hand joints. The dimension of F-dim is 1024. Finally, fingertip refinement is based on the structure of fingers being straight, and the authors added a 3D random offset within a radius sphere of r = 15mm to find the k points are the neighboring of the estimated point location of the fingertip on the GT PCD and calculate the angle between joints. This work makes the fingertip refinement network (PointNet) at the training stage to increase fingertip estimation accuracy. At the testing stage, Hand PointNet uses the estimated joint locations to calculate the angle between the joints and find neighboring points.

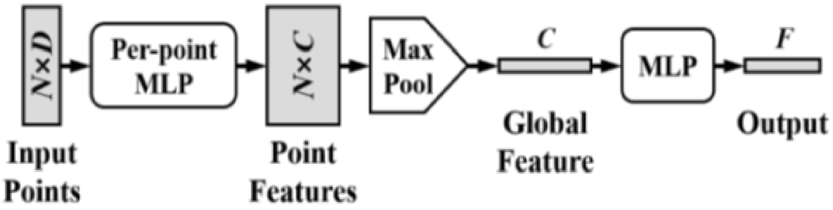

Fig. 5. PointNet architecture [13]. The input is point, then point features are extracted using per-point MLP, then GF is extracted using max-pooling, and finally, an F-dim is created by mapping the GF using MLP

3.3. V2V-PoseNet

V2V-PoseNet [14] converted a DI to a 3D volumetric (VO) form by reorienting points in 3D space and desecrating the continuum. The V2V-PoseNet estimates each candidate 3D keypoint based on each per-voxel likelihood derived from 3D voxelized data by encoder and decoder process. This process is shown in Fig. 6.

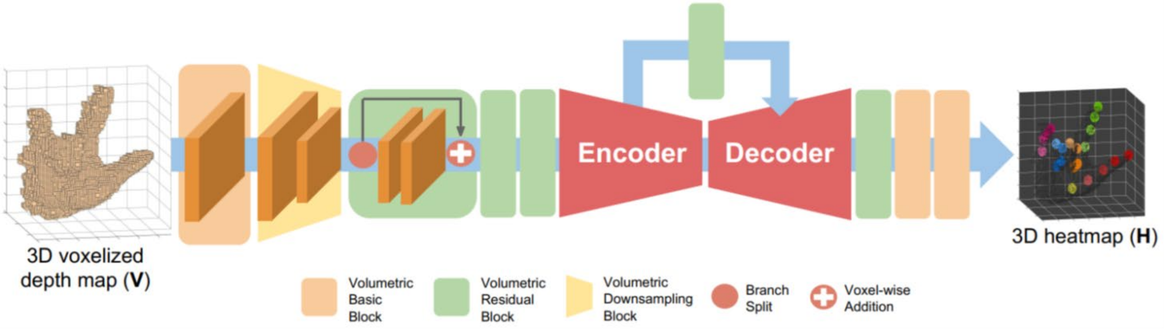

Fig. 6. Encoder and decoder architecture for estimating a candidate 3D keypoint of V2V-PoseNet [14].

Which V2V-PoseNet is divided into two stages, as Fig. 7: (1) converting from pixels to coordinates using 2D CNN and (2) converting from Voxel-to-Voxel using 3D CNN.

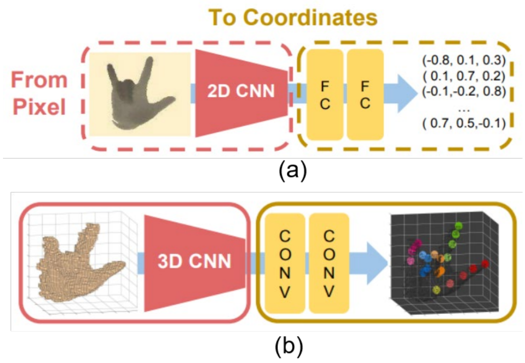

Fig. 7. Two stages of the 3D HP estimation process of V2V-PoseNet [14].

In the architecture of V2V-PoseNet, there exist four building blocks: (1) the VO basic block is first and includes a VO convolution, ReLU, and VO batch normalization. This block is placed at the first and last of V2V-PoseNet. (2) The VO residual block is second and is an extension of the 2D residual block. (3) The VO down-sampling block is third and includes a VO max pooling layer. (4) The VO up-sampling block is final and includes a VO de-convolution layer, a VO batch normalization layer, and the activation function. To simplify the training process, V2V-PoseNet can add the ReLU.

3.4. HandFoldingNet – HFNet

Fig. 8 presents the architecture of HFNet [15], it consists of four blocks: the first is the PointNet++ encoder, the second is the global folding (Glo-Fo) decoder, and the third and fourth are two local folding blocks. The backbone of HFNet is the PointNet++ encoder. The input data of HFNet is a PCD and the PCD is converted from DI into 3D space by using the (fx, fy, cx, cy) of the camera. The input data pre-processing process of HFNet is similar to Hand PointNet [13]. In addition, the input data of HFN*et also* includes the 2D hand joint skeleton, 3D surface normal vector (f_i^{nor}) is the corresponding normalized PCD (p_i^{nor}). With input point data, the $Hi - Po - Net + +$ dencoder with the architecture shown in Fig. 5 is used to extract LFs with different levels (three levels of the set of abstracts) and a single GF. The size of the set abstract level 1 is $N \ * \ (3 \ + \ C)$, the set abstract level 2 is N 1 $*$ $(3 \ + \ C1)$, and the set abstract level 3 $is\ N\ 2 \ * \ (3 \ + \ C2)$. This summarizes a GF of the PCD input. The size $Nl - 1 \ * \ 3 \ + \ Cl - 1$ of a matrix is a presentation of each level (l), the Hi-Po-Net encoder uses the input points with the size N points for extracting LFs at various levels, and each abstraction level (AbL) has the number of sub-sampled centroids is C. The neighbors of S points with each centroid p_i^l and their corresponding features have gathered the ball query within a specified radius r for a local region $\{p_{s,i}^{l-1}, f_{s,i}^{l-1}\}_{s=1}^{S}$.

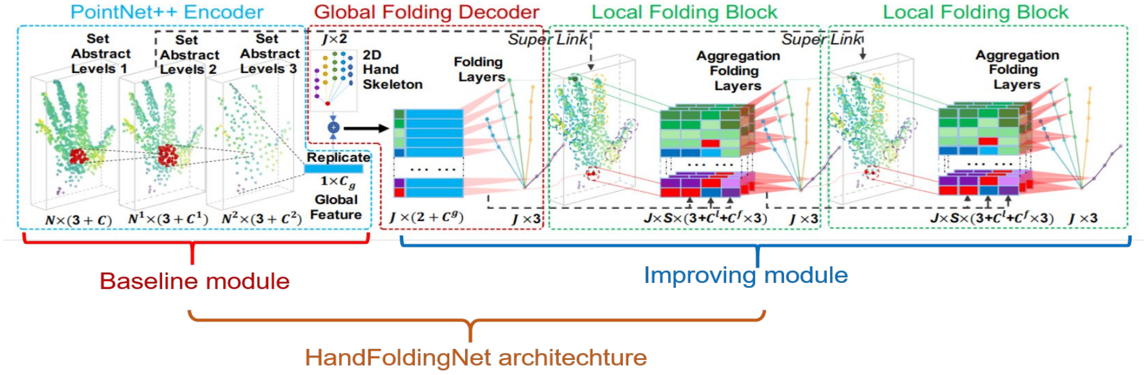

Fig. 8. HFNet architecture [15]

The matrix of HFNet is a matrix of size (J × 3), where J is the number of joints to be estimated with three 3D coordinate values (x, y, z). GF was introduced into the Glo-Fo decoder to redefine 2D HP into the initial key point coordinates. Each local folding block will perform an MLP with the size (1x1) and a Max Pool, then finally an MLP (1x1) to regress to a matrix of size J × 3. Each joint-wise LF-based Folding Block performs the detail in Fig. 9 it includes three inputs: previous estimated joint coordinates, m intermediate layers are embedded in the previous folding block, and the LF map is extracted at the previous AbL. The spatially dependent components are obtained by rearranging the folds. The output in each of these blocks is a map of aggregated features, and the rearranged embeddings are the input to the residual computation. This makes the joint estimation results more accurate. Two local folding blocks next by next performed accurate estimates of the coordinates (x, y, z) of the joint by grouping the LFs near the initial (x, y, z) of the key point.

Three inputs of each LFB are calculated: $(k \ - \ 1)$th folding block is used to estimate the joint coordinates; the intermediate layers of the folding embeddings at $(k \ - \ 1)$th folding block; $(k \ - \ 1)$th set AbL is used to extract the LF map.

In HFNet architecture, to cluster the PCD, the DBScan is used and get the clustered data area closest to the camera, and then Voxelization representation is used for sample reduction and faster computation. The exact number of points is selected by random selection in case the PCD is too few points, and in other cases, sample the farthest point. The final PCD includes 1024 points in 6D (3D is the (x, y, z) and the normal vector at each point). PCD's directional BB is also used to normalize the data, thereby making the model more invariant to rotation and resizing. Opposite to Hand PointNet [13] and some MLPs that perform the above feature extraction on 3D PCD, the backbone of HFNet is PointNet++, and three components are added: a Glo-Fo decoder and two local folding blocks. At three AbLs, the feature is extracted, and the Glo-Fo decoder receives the GF to guide the folding layer of pre-defined 2D hand key points into the initial joint coordinates for estimating the coordinates of each joint in 3D space.

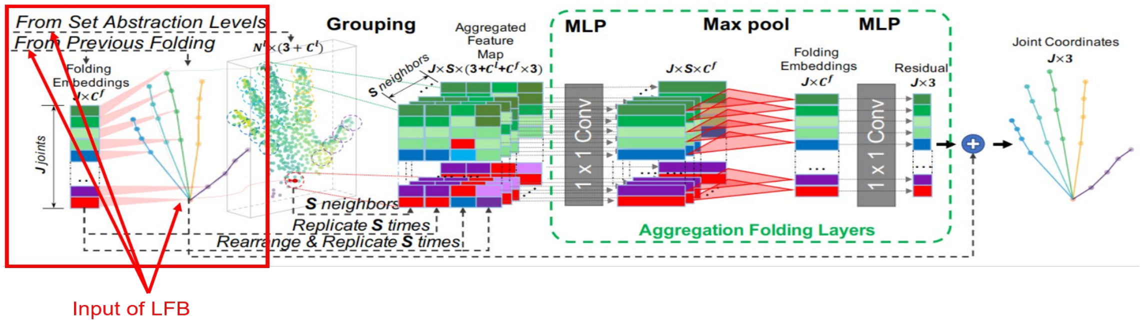
From Set Abstraction Levels
From Previous Folding
Folding Embeddings $J \times C^f$
J joints
$N^l \times (3 + C^l)$
Grouping
S neighbors
Aggregated Feature Map
$J \times S \times (3+C^l+C^f \times 3)$
S neighbors
Replicate S times
Rearrange & Replicate S times
MLP
Max pool
MLP
1 x 1 Conv
$J \times S \times C^f$
Folding Embeddings $J \times C^f$
1 x 1 Conv
Residual $J \times 3$
Aggregation Folding Layers
Joint Coordinates $J \times 3$
Input of LFB

Fig. 9. Representation of the processing of joint-wise LF based folding block [15].

4. Experimental Results

4.1. Data collection

To evaluate 3D-HPE models based on the CNNs with input data PCD and DI, we use the HOI4D [12] dataset. This is the dataset collected from EVC. The HOI4D dataset [12] is a large- scale 4D radial dataset. HOI4D contains 2.4 million RGB-D frames in radial videos with over 4,000 sequences collected by 9 people who performed operations with 800 different objects from 16 types (actions) collected at over 610 different indoor rooms. The data was collected using 4 cameras, named (”ZY20210800001”, ”ZY20210800002”, ”ZY20210800003”, ”ZY20210800004”).

Each camera has a different set of intrinsic parameters. In this paper, we divide the HOI4D data by each data collection camera. 16 HAs of the HOI4D dataset are presented in Fig. 10. In this paper, we divide the data of the HOI4D dataset by each data collection camera, and each hand activity, and perform the removal of frames without hands in the image. The GT/annotation of the HOI4D dataset is manually marked on 2D images for 20% of the frames per video. In which 11 keypoints of the hand are marked manually: the wrist, the five fingertips, and the second phalanx on the finger from the tip of the finger. The remaining 10 points are estimated based on the predefined hand model. In particular, we only use right-hand data for the fine-tuning model and evaluation (the total frame is 603,332 frames). We use a ratio (7:3), with 70% (422,332 frames) for the training model and 30% (181,000 frames) for the testing model.

In this paper, we cropped the hand data area on the image with the BB of GT. The 2D BB of GT is determined based on the 2D GT HP. To generate the input data for the CNNs to estimate 3D HP. We performed to change the hand data on the image to the 3D PCD. Where each data point has (Px , P y, P z) in the PCD generated by Eq. (1).

$$P_x = \frac{(x_d - c_x) \times D_{ya}}{fx}$$

$$P_y = \frac{(y_d - c_y) \times D_{ya}}{fy}$$

$$P_z = D_{ya} \tag{1}$$

where $fx, fy, cx,$ and cy are taken from inside the camera. D_{ya} is the depth value at the pixel (x_d, y_d).

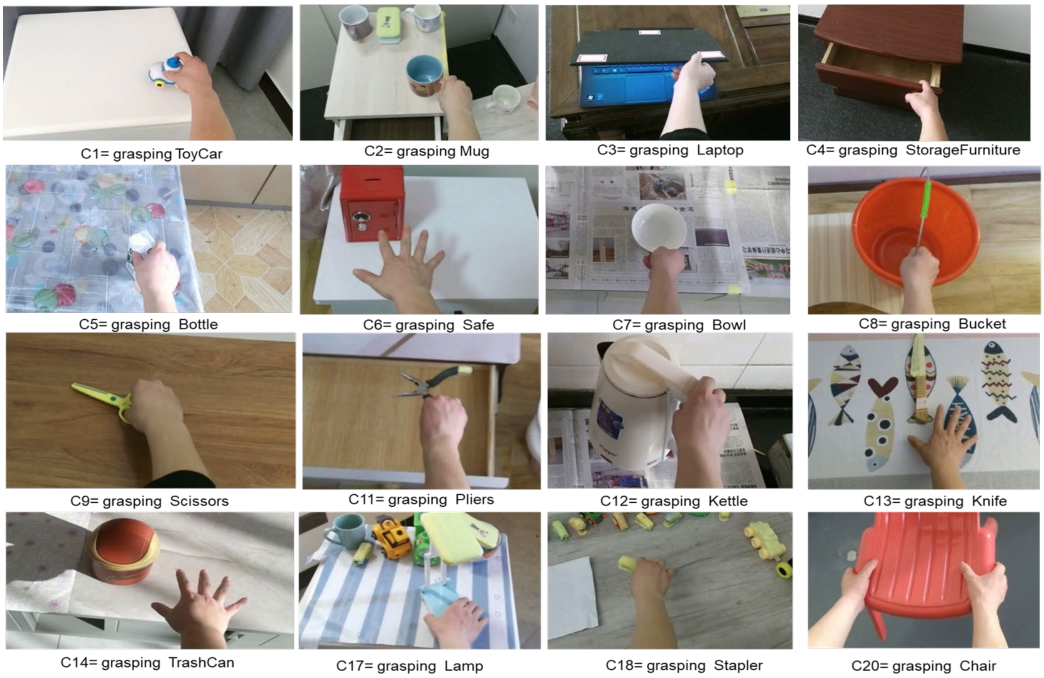

Fig. 10. Illustrating HAs in the HOI4D dataset

Fig. 11 shows the process of generating the PCD using the formula (1), and then segmented based on the Euclidean distance (ED) [39] to remove non-hand data areas and the remaining PCD data region of the hand. It is then standardized based on DB.

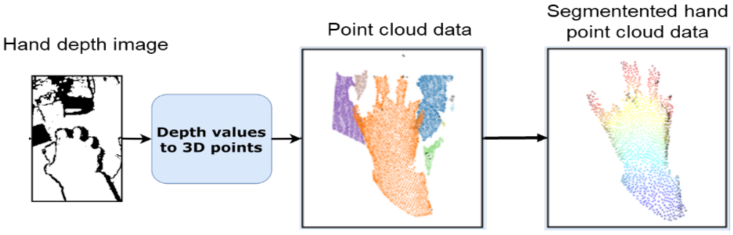

Fig. 11. Illustration of creating a hand PCD

4.2. Metrics

The results of 3D HPE are evaluated similarly to previous studies, the average 3D distance error Err_a is used, as calculated in Eq. 2. The results are averaged based on the ED between the 3D HP annotation/GT and the estimated 3D HP on the HOI4D datasets.

$$Err_a = \frac{1}{Num_s} \sum_{n=1}^{Num_s} \frac{1}{k} \sum_{k=1}^{K} EDIS\left(p_g, p_e\right) \tag{2}$$

where $EDIS\ (pg, pe)$ is the ED between a GT keypoint p_g and an estimated keypoint pe in millimeters(mm); Num_s is the testing frames; K = 21 is the keypoints/joints of 3D HPE.

The hands of the HOI4D dataset include 21 joints. Therefore, we evaluated the 21 joints of the HP. The pre-trained model for 3D HPE is fine-tuned with 50 epochs with the number batch size being 32. The size of the samples at the 1st AbL is 1024, the 2nd AbL is 512, and the 3rd AbL is 128. The components of PCA have a size of 42, the size of the K for the KNN (K-Nearest Neighbors) search is

64, and the square of radius for the ball query in level 1 is 0.015, and level 2 is 0.04. The Adam optimizer is used in the deep network (HFNet). These parameters are selected as the best parameters. The source code of pre-processing program, to process the DI, PCD, is developed on MatLab. The CNNs model training and model testing source code is developed in the open source programming language Python with the support of several programming libraries such as Tqdm, PyTorch, etc. Programs executed on the Ubuntu Server have the following configuration: NVIDIA GeForce RTX 3060 12GB GPU. Our paper uses the published source codes of P2PR PointNet ([1]), Hand PointNet ([2]), V2V-PoseNet ([3]), and HFNet ([4]).

5. Results and Discussion

The results of 3D-HPE of the right hand based on CNNs on the HOI4D dataset are shown in Table. 1. The average error Erra of 3D R-HPE on the ZY20210800001 data is (P2PR PointNet is 18.2mm), (Hand PointNet is 13.1 mm), (V2V-PoseNet is 15.2mm), and (HFNet is 14mm), respectively. The average error Erra of 3D R-HPE on the ZY20210800002 data is (P2PR PointNet is 35.9mm), (Hand PointNet is 61.07 mm), (V2V-PoseNet is 28.9mm), and (HFNet is 22.72mm), respectively. The average error Erra of 3D R-HPE on the ZY20210800003 data is (P2PR PointNet is 50.44mm), (Hand PointNet is 49.68 mm), (V2V-PoseNet is 45.7mm), and (HFNet is 32.85mm), respectively. The average error Erra of 3D R-HPE on the ZY20210800004 data is (P2PR PointNet is 26.33mm), (Hand PointNet is 16.65 mm), (V2V-PoseNet is 15.5mm), and (HFNet is 12.41 mm), respectively.

The results in Table. 1 show that HFNet has the best estimate with an average error (Erra = 20.49 mm) on all cameras and right-hand activity. This distance error is acceptable for simulating complex HAs for grasping objects to build robotic arms capable of handling objects. Or you can guide the visually impaired to grasp objects in the environment.

Table 1. The average error (Err_a) result of 3D-HPE on the HOI4D dataset

Methods	Average error distance (Erra) (mm)			
	ZY202108 00001	*ZY202108 00002*	*ZY202108 00003*	*ZY202108 00004*
P2PR PointNet [13]	18.2	35.9	50.44	26.33
Hand PointNet [13]	**13.1**	61.07	49.68	16.65
V2V-PoseNet [14]	15.2	28.9	45.7	15.5
HFNet [15]	14	**22.72**	**32.85**	**12.41**

Fig. 12 is the 3D-HPE error distribution on the testing set of the HOI4D dataset. This distribution can be seen, the estimation error of the data obtained from the 4th camera (ZY20210800004) is the highest (yellow line), and the error of the data received from the first camera (ZY20210800001) is the smallest (deep sky blue line).

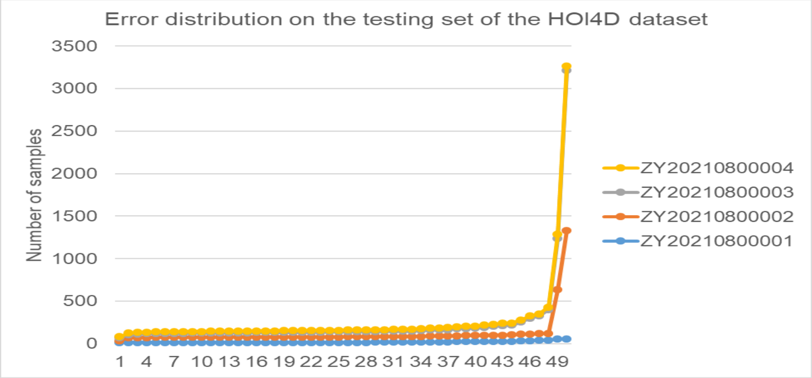

Fig. 12. Illustration of the 3D-HPE error distribution on the HOI4D dataset

Fig. 13 shows the results of 3D-HPE with 16 actions of holding objects using HFNet. The blue skeleton is a 3D HP annotation/GT. The red skeleton is a 3D-HPE.

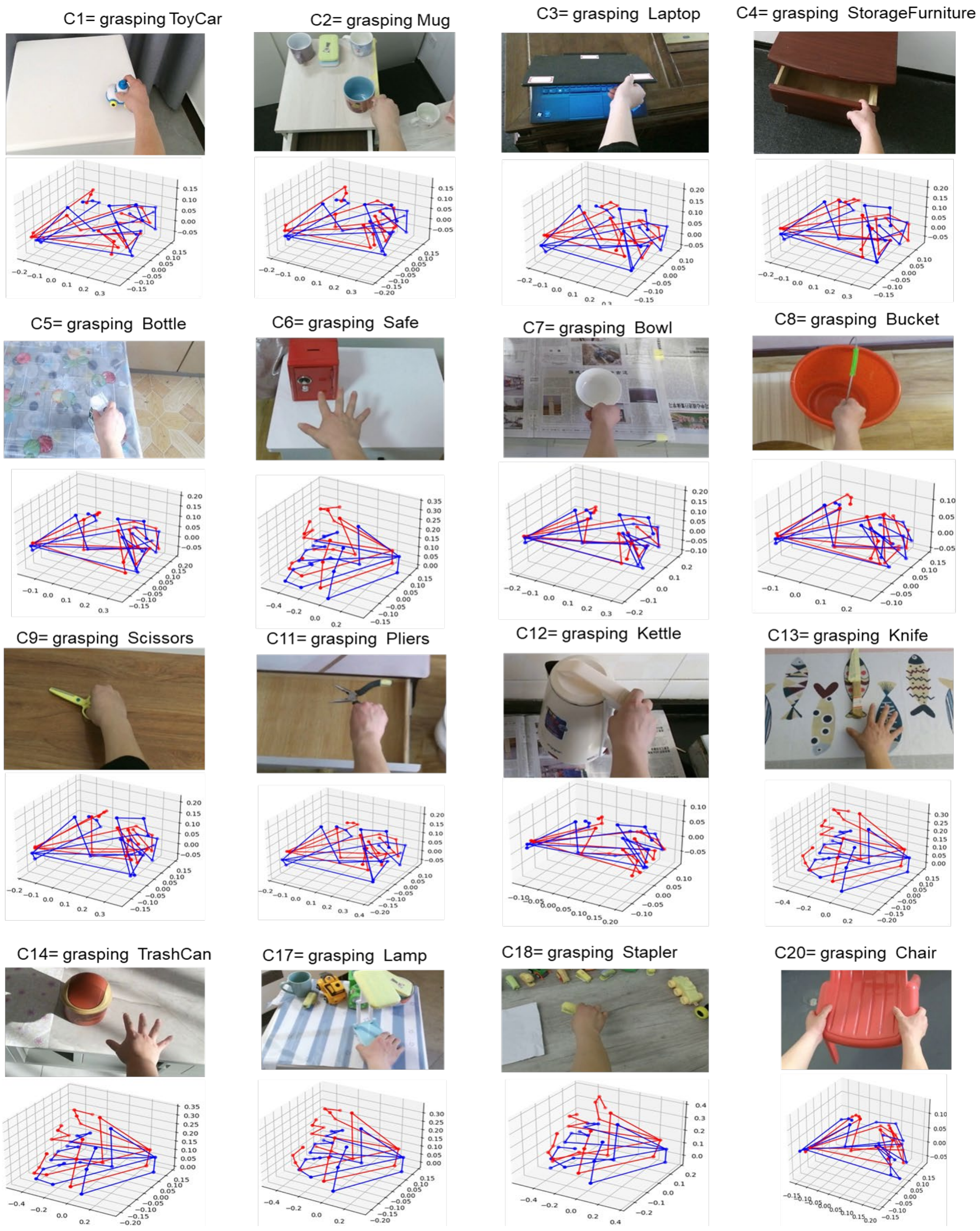

Fig. 13. Illustrating the results of 3D-HPE of 16 actions of holding objects

Table 2 shows the average speed of 3D R-HPE on the HOI4D dataset with P2PR PointNet, Hand PointNet, V2V-PoseNet, and HFNet.

Table 2. The average processing time to estimate 3D R-HP

Methods	P2PR PointNet	Hand PointNet	V2V-PoseNet	HFNet
Processing time (fps)	3.5	4.2	3.8	**5.4**

The processing time of 3D-HPE of the HFNet is the fastest (5.4fps). This is the time to only estimate the 3D skeleton/pose/joints of the hand. On the HOI4D dataset, some activities have only one hand in the scene, which is the right hand; in some activities, there are two hands (both left and right hands). We are only interested in estimating the 3D-HP of the right hand. Since the input of CNNs for 3D-HPE is the detected hand data, segmented with other data (this means only hand data). The data pre-processing of P2PR PointNet, Hand PointNet, and HFNet is similar and has been performed on MatLab. The estimation process of CNNs is developed in Python language and run on GPU, which shows the computation time shown in Table. 2 is on par.

6. Conclusion

Estimating a 3D-HP from an EVC presents challenges, as the hand can be obscured. The results of 3D-HPE have great applications in robotics and building assistive and guiding systems to hold objects for visually impaired people. To find and grasp the objects, based on the estimated hand position, objects, and size of objects in the real world, the system can guide blind people to pre-determine hand shape and hand size to perform safe handling, or some applications in HCI, healthcare, etc. Therefore, the 3D-HPE step is very important to build intuitive applications. Our paper performs an empirical study on 3D R-HPE based on PCD input with typical CNNs for 3D-HPE: P2PR PointNet, Hand PointNet, V2V-PoseNet, and HFNet. The 3D R-HPE model is pre-trained from data collected from four cameras of the HOI4D dataset. Before performing the fine-tuning of the estimation and evaluation model, we normalized the data of the HOI4D dataset. The 3D R-HPE result with the error distance of P2PR PointNet is $(Erra = 32.71mm)$, Hand PointNet is $(Erra = 35.12mm)$, V2V-PoseNet is $(Erra = 26.32mm)$ and HFNet is $(Erra = 20.49mm)$. 3D R- HPE results with HFNet are the best. This result can be applied to building practical applications in robotics and assisting visually impaired people. The short-term plan is that the accuracy of 3D-HPE will improve. Apply this result to end-to-end model building in automatically detecting, estimating, and recognizing HA in 3D space.

Acknowledgment

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

Declarations

Author contribution. This article is my own work and is not affiliated with any individual organization
Funding statement. This research is funded by Tan Trao University in Tuyen Quang province, Vietnam.
Conflict of interest. The authors declare no conflict of interest.
Additional information. No additional information is available for this paper.

References

[1] C. Bandi and U. Thomas, “Regression-Based 3D Hand Pose Estimation for Human-Robot Interaction,” in *Communications in Computer and Information Science*, vol. 1474 CCIS, Springer, Cham, 2022, pp. 507–529, doi: 10.1007/978-3-030-94893-1_24.

[2] Q. Gao, Y. Chen, Z. Ju, and Y. Liang, “Dynamic Hand Gesture Recognition Based on 3D Hand Pose Estimation for Human–Robot Interaction,” *IEEE Sens. J.*, vol. 22, no. 18, pp. 17421–17430, Sep. 2022, doi: 10.1109/JSEN.2021.3059685.

[3] S. Tsutsui, Y. Fu, and D. Crandall, “Whose hand is this? Person Identification from Egocentric Hand Gestures,” in *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2021, pp. 3398–3407, doi: 10.1109/WACV48630.2021.00344.

[4] M.-F. Tsai, R. H. Wang, and J. Zariffa, “Recognizing hand use and hand role at home after stroke from egocentric video,” *PLOS Digit. Heal.*, vol. 2, no. 10, p. e0000361, Oct. 2023, doi: 10.1371/journal.pdig.0000361.

[5] K. Delloul and S. Larabi, “Egocentric Scene Description for the Blind and Visually Impaired,” in *2022 5th International Symposium on Informatics and its Applications (ISIA)*, Nov. 2022, pp. 1–6, doi: 10.1109/ISIA55826.2022.9993531.

[6] A. Bandini and J. Zariffa, "Analysis of the Hands in Egocentric Vision: A Survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 6, pp. 6846–6866, Jun. 2023, doi: 10.1109/TPAMI.2020.2986648.

[7] 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 *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, Jun. 2018, pp. 409–419, doi: 10.1109/CVPR.2018.00050.

[8] V.-H. Le and H.-C. Nguyen, "A Survey on 3D Hand Skeleton and Pose Estimation by Convolutional Neural Network," *Adv. Sci. Technol. Eng. Syst. J.*, vol. 5, no. 4, pp. 144–159, Jul. 2020, doi: 10.25046/aj050418.

[9] L. Fan, H. Rao, and W. Yang, "3D Hand Pose Estimation Based on Five-Layer Ensemble CNN," *Sensors*, vol. 21, no. 2, p. 649, Jan. 2021, doi: 10.3390/s21020649.

[10] J. H. R. Isaac, M. Manivannan, and B. Ravindran, "Single Shot Corrective CNN for Anatomically Correct 3D Hand Pose Estimation," *Front. Artif. Intell.*, vol. 5, p. 759255, Feb. 2022, doi: 10.3389/frai.2022.759255.

[11] J. Cheng *et al.*, "Efficient Virtual View Selection for 3D Hand Pose Estimation," *Proc. AAAI Conf. Artif. Intell.*, vol. 36, no. 1, pp. 419–426, Jun. 2022, doi: 10.1609/aaai.v36i1.19919.

[12] Y. Liu *et al.*, "HOI4D: A 4D Egocentric Dataset for Category-Level Human-Object Interaction," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 20981–20990, doi: 10.1109/CVPR52688.2022.02034.

[13] L. Ge, Z. Ren, and J. Yuan, "Point-to-Point Regression PointNet for 3D Hand Pose Estimation," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11217 LNCS, Springer Verlag, 2018, pp. 489–505, doi: 10.1007/978-3-030-01261-8_29.

[14] 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*, 2017, pp. 5079–5088. [Online]. Available at: https://openaccess.thecvf.com/content_cvpr_2018/papers/Moon_V2V-PoseNet_Voxel-to-Voxel_Prediction_CVPR_2018_paper.pdf.

[15] W. Cheng, J. H. Park, and J. H. Ko, "HandFoldingNet: A 3D Hand Pose Estimation Network Using Multiscale-Feature Guided Folding of a 2D Hand Skeleton," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 11240–11249, doi: 10.1109/ICCV48922.2021.01107.

[16] L. Huang, B. Zhang, Z. Guo, Y. Xiao, Z. Cao, and J. Yuan, "Survey on depth and RGB image-based 3D hand shape and pose estimation," *Virtual Real. Intell. Hardw.*, vol. 3, no. 3, pp. 207–234, Jun. 2021, doi: 10.1016/j.vrih.2021.05.002.

[17] T. Ohkawa, R. Furuta, and Y. Sato, "Efficient Annotation and Learning for 3D Hand Pose Estimation: A Survey," *Int. J. Comput. Vis.*, vol. 131, no. 12, pp. 3193–3206, Dec. 2023, doi: 10.1007/s11263-023-01856-0.

[18] C. R. Q. Li, Y. Hao, S. Leonidas, and J. Guibas, "PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space," in *NIPS'17: Proceedings of the 31st International Conference on Neural Information Processing Systems*, 2017, pp. 5105–5114. [Online]. Available at: 10.5555/3295222.3295263.

[19] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-Fine Volumetric Prediction for Single-Image 3D Human Pose," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jul. 2017, vol. 2017-Janua, pp. 1263–1272, doi: 10.1109/CVPR.2017.139.

[20] F. Choi, S. Mayer, and C. Harrison, "3D Hand Pose Estimation on Conventional Capacitive Touchscreens," in *Proceedings of the 23rd International Conference on Mobile Human-Computer Interaction*, Sep. 2021, pp. 1–13, doi: 10.1145/3447526.3472045.

[21] D. Drosakis and A. Argyros, "3D Hand Shape and Pose Estimation based on 2D Hand Keypoints," in *Proceedings of the 16th International Conference on PErvasive Technologies Related to Assistive Environments*, Jul. 2023, pp. 148–153, doi: 10.1145/3594806.3594838.

[22] M. Ivashechkin, O. Mendez, and R. Bowden, "Denoising Diffusion for 3D Hand Pose Estimation from Images," in *2023 IEEE/CVF International Conference on Computer Vision Workshops (ICCVW)*, Oct. 2023, pp. 3128–3137, doi: 10.1109/ICCVW60793.2023.00338.

[23] Y. Wen, H. Pan, L. Yang, J. Pan, T. Komura, and W. Wang, "Hierarchical Temporal Transformer for 3D Hand Pose Estimation and Action Recognition from Egocentric RGB Videos," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, vol. 2023-June, pp. 21243–21253, doi: 10.1109/CVPR52729.2023.02035.

[24] E. Kazakos, J. Huh, A. Nagrani, A. Zisserman, and D. Damen, "With a Little Help from my Temporal Context: Multimodal Egocentric Action Recognition," in *32nd British Machine Vision Conference, BMVC 2021*, 2021, pp. 1–16, [Online]. Available at: https://www.bmvc2021-virtualconference.com/assets/papers/0610.pdf.

[25] V.-D. Le *et al.*, "Hand Activity Recognition From Automatic Estimated Egocentric Skeletons Combining Slow Fast and Graphical Neural Networks," *Vietnam J. Comput. Sci.*, vol. 10, no. 01, pp. 75–100, Feb. 2023, doi: 10.1142/S219688882250035X.

[26] B. Doosti, S. Naha, M. Mirbagheri, and D. J. Crandall, "HOPE-Net: A Graph-Based Model for Hand-Object Pose Estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2020, pp. 6607–6616, doi: 10.1109/CVPR42600.2020.00664.

[27] V.-H. Le, "Automatic 3D Hand Pose Estimation Based on YOLOv7 and HandFoldingNet from Egocentric Videos," in *2022 RIVF International Conference on Computing and Communication Technologies (RIVF)*, Dec. 2022, pp. 161–166, doi: 10.1109/RIVF55975.2022.10013903.

[28] A. Prakash, R. Tu, M. Chang, and S. Gupta, "3D Hand Pose Estimation in Egocentric Images in the Wild," *arXiv Comput. Vis. Pattern Recognit.*, pp. 1–11, 2023, [Online]. Available at: http://arxiv.org/abs/2312.06583.

[29] C. Plizzari *et al.*, "An Outlook into the Future of Egocentric Vision," *Int. J. Comput. Vis.*, pp. 1–57, May 2024, doi: 10.1007/s11263-024-02095-7.

[30] S. Pramanick *et al.*, "EgoVLPv2: Egocentric Video-Language Pre-training with Fusion in the Backbone," in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2023, pp. 5262–5274, doi: 10.1109/ICCV51070.2023.00487.

[31] L. Zhang, S. Zhou, S. Stent, and J. Shi, "Fine-Grained Egocentric Hand-Object Segmentation: Dataset, Model, and Applications," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 13689 LNCS, Springer, Cham, 2022, pp. 127–145, doi: 10.1007/978-3-031-19818-2_8.

[32] X. Gong *et al.*, "MMG-Ego4D: Multi-Modal Generalization in Egocentric Action Recognition," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, vol. 2023-June, pp. 6481–6491, doi: 10.1109/CVPR52729.2023.00627.

[33] L. Khaleghi, A. Sepas-Moghaddam, J. Marshall, and A. Etemad, "Multiview Video-Based 3-D Hand Pose Estimation," *IEEE Trans. Artif. Intell.*, vol. 4, no. 4, pp. 896–909, Aug. 2023, doi: 10.1109/TAI.2022.3195968.

[34] C. Plizzari *et al.*, "E 2 (GO)MOTION: Motion Augmented Event Stream for Egocentric Action Recognition," in *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2022, vol. 2022-June, pp. 19903–19915, doi: 10.1109/CVPR52688.2022.01931.

[35] T. Ohkawa, K. He, F. Sener, T. Hodan, L. Tran, and C. Keskin, "AssemblyHands: Towards Egocentric Activity Understanding via 3D Hand Pose Estimation," in *2023 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2023, vol. 2023-June, pp. 12999–13008, doi: 10.1109/CVPR52729.2023.01249.

[36] X. Wang, L. Zhu, H. Wang, and Y. Yang, "Interactive Prototype Learning for Egocentric Action Recognition," in *2021 IEEE/CVF International Conference on Computer Vision (ICCV)*, Oct. 2021, pp. 8148–8157, doi: 10.1109/ICCV48922.2021.00806.

[37] M. Planamente, C. Plizzari, E. Alberti, and B. Caputo, "Domain Generalization through Audio-Visual Relative Norm Alignment in First Person Action Recognition," in *2022 IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, Jan. 2022, pp. 163–174, doi: 10.1109/WACV51458.2022.00024.

[38] Y. Wu, L. Zhu, X. Wang, Y. Yang, and F. Wu, "Learning to Anticipate Egocentric Actions by Imagination," *IEEE Trans. Image Process.*, vol. 30, pp. 1143–1152, 2021, doi: 10.1109/TIP.2020.3040521.

[39] H. Liu, R. Song, X. Zhang, and H. Liu, "Point cloud segmentation based on Euclidean clustering and multi-plane extraction in rugged field," *Meas. Sci. Technol.*, vol. 32, no. 9, p. 095106, Sep. 2021, doi: 10.1088/1361-6501/abead3.