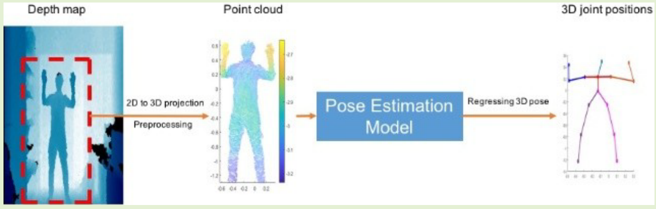
Learning to Estimate 3D Human Pose From Point Cloud

Yufan Zhou, Haiwei Dong, *Senior Member, IEEE*, and Abdulmotaleb El Saddik, *Fellow, IEEE*

***Abstract*—3D pose estimation is a challenging problem in computer vision. Most of the existing neural-network-based approaches address color or depth images through convolution networks (CNNs). In this paper, we study the task of 3D human pose estimation from depth images. Different from the existing CNN-based human pose estimation method, we propose a deep human pose network for 3D pose estimation by taking the point cloud data as input data to model the surface of complex human structures. We first cast the 3D human pose estimation from 2D depth images to 3D point clouds and directly predict the 3D joint position. Our experiments on two public datasets show that our approach achieves higher accuracy than previous state-of-art methods. The reported results on both ITOP and EVAL datasets demonstrate the effectiveness of our method on the targeted tasks.**



***Index Terms*—Edge feature, pose regression network, depth image.**

I. Introduction

POSE estimation aims to identify and locate the key points of all human bodies in an image [1]. This is a basic research topic for many visual applications, such as human motion recognition and human-computer interaction. With the recent development of neural networks, many methods perform well using CNN, including stacked hourglass networks [2] and multiscale structure-aware networks [3]. Some researchers have estimated 2D human poses based on heat maps. Others approaches [4]–[6] regressed the heat maps for 3D pose estimation. In addition, Shotton *et al.* [7] and Junget *et al.* [8] took depth images as the input to process 3D estimation. Some methods are not only applicable in hand pose estimation but also to regress the human body joints, such as the region ensemble network [9] and the voxel-to-voxel network [10]. Compared to color images, depth images contain 3D information about the distance between the scene object and the camera viewpoint. With this considerable advantage, using 3D information can make a better 3D prediction.

3D interaction devices such as the Intel RealSense, Microsoft Kinect sensor, and other types of depth/IR cameras have a positive impact on the technology of human-computer interaction (HCI) [11], [12]. Images captured by depth sensors are widely used in robotics and autonomous driving. Point cloud maps, as a form of data structure of depth images, have both geometric positions and intensity information. The study of point cloud maps has gradually improved from geometric feature extraction to high-level understanding, such as point cloud segmentation [13] and recognition [14]. Different from ordinary image perception, pose estimation from a point cloud is challenging. First, the data structure of the point cloud is a set of points that are composed of point coordinates in three-dimensional space. These points only provide one-sided geometric information. The limited information cannot be used to extract enough features. Second is the sparsity of the point cloud. When the same object is scanned by different 3D devices in different positions, the order of the three-dimensional points varies widely. Such data are difficult to directly process through an end-to-end model. Third, we need to consider the time and space complexities in performing real-time tasks. Recent networks that directly process each point individually from the point cloud are computationally expensive. Fourth, 3D skeleton joints of most datasets are captured from a depth camera instead of using a Vicon camera to track the markers with high accuracy.

What we achieve by converting the depth image into a point cloud is not the actual 3D data, but the 2.5D data from the object surface, which is not suitable for directly processing the point cloud by 2D CNN or 3D CNN. However, the time and space complexities of the 3D CNN grow cubically with the resolution of the input 3D volume. As Figure 1 shows,

Manuscript received April 19, 2020; revised May 27, 2020; accepted May 31, 2020. Date of publication June 3, 2020; date of current version September 17, 2020. The associate editor coordinating the review of this article and approving it for publication was Prof. Kazuaki Sawada. *(Corresponding author: Haiwei Dong.)*

The authors are with the Multimedia Computing Research Laboratory, School of Electrical Engineering and Computer Science, University of Ottawa, Ottawa, ON K1N 6N5, Canada (e-mail: yzhou158@uottawa.ca; haiwei.dong@ieee.org; elsaddik@uottawa.ca).

Digital Object Identifier 10.1109/JSEN.2020.2999849

1558-1748 © 2020 IEEE. Personal use is permitted, but republication/redistribution requires IEEE permission.
See https://www.ieee.org/publications/rights/index.html for more information.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

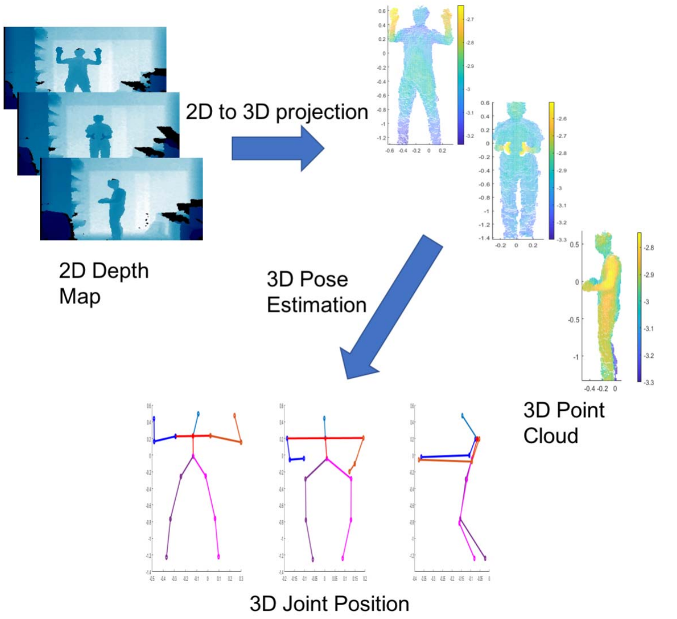


Fig. 1. The 3D point cloud has a one-to-one relation with a 3D pose. Our approach is based on point clouds, converting depth images into point clouds before pose estimation.

we aim to learn 3D human key body joint positions directly from the 3D point cloud. PointNet++ [15] and DGCNN (Dynamic Graph CNN) [16] have been motivated by the recent works on PointNet [14] that perform both 3D object classification and 3D object segmentation on point clouds directly.

In this approach, our idea to perform pose estimation is that first, we take the images as input and convert them into 3D point clouds. Depending on the depth thresholding and Euclidean cluster extraction, we extract the person from these 3D points. Points of the human body are normalized by the height and width of this person before being transferred into the deep learning network. Our network is designed on a modified dynamic graph CNN model and PointNet, whose output is a low dimensional representation of the 3D key points. This work captures 3D structures of the human body and accurately estimates 3D human poses.

Our contributions can be summarized as follows.

- To the best of our knowledge, we proposed estimating the human body key joints directly from 3D point clouds based on the network architecture of DGCNN and PointNet for the first time. Unlike other methods that regress the 3D key points from the depth images, we cast the problem of 3D pose estimation from a single depth image to the point clouds.
- We processed the experiment using two existing representative 3D human pose datasets–EVAL [17] and ITOP [18]. We compared our approach with other CNN-based [18] and RF/RTW-based [7], [8], [19] human pose estimation methods from depth images. Experimental results show that our network for 3D pose estimation has a significantly accurate performance.

II. Related Work

A. Depth-Based 3D Pose Estimation

The methods of depth-based 3D pose estimation are divided into two parts: generative and discriminative models. The generative models are similar to the template matching. The human body templates are required to find correspondence between the inputs and templates in generative models. The iterative closest point algorithm [17] is commonly used to track the 3D human body. Different from generative models, discriminative models directly estimate the pose. In the following, we focus on conventional discriminative models.

Most of the discriminative models are based on random forests (RF). Shotton *et al.* [7] classified body parts from a single depth image based on the random forest classifier and estimated the 3D joint locations. Jung *et al.* [8] used a random tree walk algorithm (RTW) to regress the joint position and reduce the running time. In the field of deep learning, Haque *et al.* [18] proposed a viewpoint-invariant model using CNN and recurrent networks for human pose estimation, while Guo *et al.* [9] introduced a tree-structured region ensemble network for 3D position regression. These deep learning models are based on the image features. In addition, a voxel-to-voxel network (V2V-PoseNet) is proposed in [10], which takes the point cloud as input. For each voxel, the network estimates the likelihood of each body joint. V2V-PoseNet extracts the 3D joint positions from the generated heatmaps.

B. 3D Deep Learning for Point Cloud

A series of PointNet models, including PointNet [14] and PointNet++ [15] are the recently proposed methods for point cloud classification and segmentation. Point clouds are fundamentally irregular, and it is not sensitive to the order of the data. This means that the model for processing point cloud data needs to be invariant to different permutations of the data. A spatial transform network named T-Net (part of PointNet [14]) has been designed to ensure the invariance of the model to a specific spatial transformation. The points are first aligned by multiplying it with a transformation matrix learned by a spatial transform network. PointNet models treat each point individually, learning the mapping from 3D to potential features without taking advantage of geometry. A single maxpooling layer is used for all the features of sampled points in PointNet. This operation loses many local features and only maintains global features. Therefore, the network's ability to extract local information from the model is far less satisfactory than that of the convolutional neural network. PointNet++ extracts features on different scales and obtains deep features through a multilayer cascade network. There are three main modules, sampling, grouping and feature learning, for extracting both local and global features. However, PointNet++ still considers individual points in local point sets instead of the relationships between a pair of points.

In contrast to previous networks, Wang *et al.* [16] successfully developed a dynamic graph CNN model (DGCNN) to process point clouds based on a thinking way of image analysis using a convolution neural network (CNN). The method is the same as the principle of a convolutional network

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

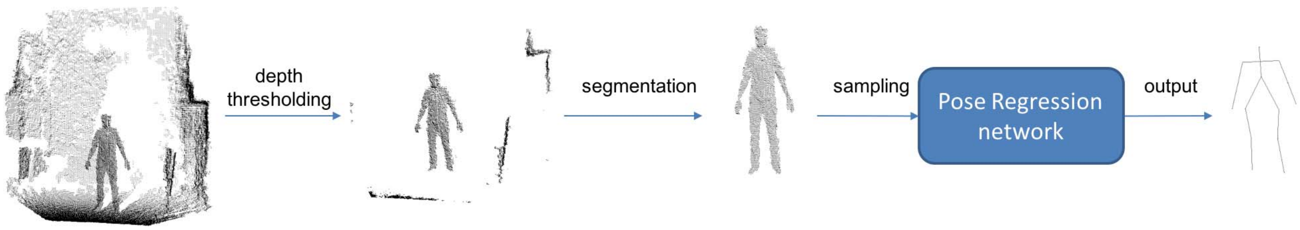


Fig. 2. The point cloud is converted from the depth image. We define distance thresholds to identify the subject and segment the human body from the background. After sampling the point cloud into the same size, it is fed into the regression network. The output is the 3D coordinates of the skeleton joints.

that considers the set of interconnected pixels instead of a single pixel. EdgeConv, as the main part of DGCNN, takes a center point with its nearest neighbor points as an input to the multilayer perceptron (MLP) layer. EdgeConv models the distance and geometric structure between points when constructing the neighborhood. Local features are formed by searching neighbor points, while global features can be extracted through stacking or recycling EdgeConv layers. However, unlike PointNet and PointNet++, not only global features but also local features are considered in EdgeConv layers.

III. Deep Learning Model

Our method for pose estimation takes the point clouds converted from the depth images as input, considering this F-dimensional point cloud with N points. The point cloud is downsampled to N points, which is defined as a set of vectors $P = \{p_1, p_2, \ldots, p_N\}$. Outputs are a set of 3D key body joint locations $J = \{j_1, \ldots, j_M\}$, where M is the number of key body joints and j_i contains x_i, y_i, z_i in the camera coordinate system where $i \in \{1, \ldots, M\}$. The data input to the deep learning model is (P, J). Considering the resolution of the depth image, we set N as $5,000$ and F as 3. Our model takes these sampled point clouds to regress the 3D body pose by extracting the global and local features. To further improve the results, we modified the DGCNN model. In this section, we introduce our preprocessing and then present our 3D pose estimation method.

A. Preprocess

First, our approach segments the human body from the background as clearly as possible. Figure 2 shows the process of segmentation and sampling. Since the data are captured by the depth camera, many points with invalid depth values or zero-depth values may be included. We removed all invalid and zero points. Given the depth information, most of the background points can be easily removed by defining depth thresholding. However, only setting the depth thresholding is not enough to extract the human body. Point clouds still include noise and other background objects, which may affect the results. These are mostly due to the photon shot noise and long distance between the human body and the camera.

Setting a bounding box and depth thresholding can eliminate many background objects from Figure 2. In a more general case, we can make use of nearest neighbors and implement a clustering technique that is essentially similar to a flood fill algorithm [20]. The main idea for removing background subjects in point clouds is using the Euclidean cluster extraction filter as shown in Algorithm 1. $Search(n_i, d_{th})$ is to search for the neighbors of n_i with a radius $r < d_{th}$. The distance thresholding d_{th} is set to $10cm$. $Size(x)$ returns the size of the cluster x. Each point from the point cloud is checked whether the distance between it and its neighbors is below thresholding. If the conditions are satisfied, point and its neighbors belong to the same cluster. Since point cloud data provide higher dimensional data, there is considerable information that can be extracted. In our proposed approach, segmentation based on Euclidean distance was performed by removing the small noise cluster.

After the previous steps, considering the different numbers of points, we downsampled the number of original point clouds into the same size. Normalization of the point cloud is executed by the person's height (b_h) and width (b_w) in this point cloud to address the data of different sizes. Since joint coordinates are in absolute image coordinates, it proves beneficial to normalize them with a bounding box b, which includes the person's height (b_h) and width (b_w). In a trivial case, the box can denote the full image. Such a box is defined by its center. The center in the point cloud is the human body center b_c. b_c can be derived from the average of all joints from the ground truth. As shown in Equations 1 and 2, $NOR(p_i, b)$ is the function for point cloud normalization.

$$NOR(p_i, b) = (p_i - b_c)\begin{bmatrix} \frac{1}{b_w} & 0 & 0 \\ 0 & \frac{1}{b_h} & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{1}$$

$$b_c = \frac{\sum_{i=0}^{N} p_i}{N} \tag{2}$$

where N is the number of points from the human body and p_i is the point from point cloud P of the human body.

B. Pose Regression Network

We designed a 3D pose regression network that can directly estimate the 3D coordinates of the pose from the input point clouds by modifying DGCNN and PointNet. In our model, we use T-Net as the spatial transform network to achieves permutation invariance of points and stack EdgeConv layers to extract and learn features from the transformed point clouds.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

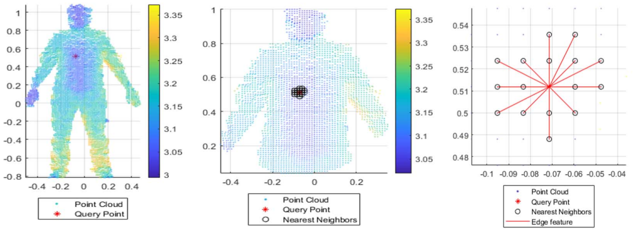


Fig. 3. An example of edge features from the chest. We define an arbitrary center query point from the chest and its neighborhoods. Depending on the distance, we can find the *K* neighbor points. The edge features are composed of these *K*+1 points. The point clouds with edge features are the input to the neural network. We set *K* to 16 in this work.

Algorithm 1 Euclidean Cluster Extraction

```
Input: Point cloud $P$
Output: A point set of human body $C_{body}$
set an empty queue $Q$ and an empty list $C$
for $p_i \in P$ do
  add $p_i$ to $Q$
  for $q_i \in Q$ do
    $Neighbors = Search(q_i, d_{th})$
    for $n_i \in Neighbors$ do
      if $n_i \notin Q$ then
        add $n_i$ into $Q$
      end if
    end for
  end for
  add $Q$ to $C$
  set $Q$ to empty
end for
$C_{body} = Argmax_{x \in C} Size(x)$
return $C_{body}$
```

The original DGCNN model is used for classification and segmentation. We modified the architecture of the classification model by changing the middle layers and fully connected layers. The architecture of the modified model is shown in Figure 4. Different from the original model, we use one EdgeConv layer with 128 filters instead of three 64 filters. EdgeConv is a basic part of the DGCNN, which captures the local structures of the human body by combining the points and their neighborhoods. In our model, EdgeConv has been used as a main part for feature extraction. As Figure 3 shows, we visualize an edge feature that is considered a local feature. The feature is composed of the *K* nearest neighbors by calculating the Euclidean distance. EdgeConv takes the local feature as input, considering the coordinates of points and the distance from the domain points as local domain information. Global shape information can be extracted by stacking or recycling EdgeConv layers.

For each point $p_i \in P$, we find its K neighbor points and concatenate the neighbor points with the original point cloud as the edge features, which are calculated in the EdgeConv layers. There are two units in our network, which are composed of EdgeConv layers. The first unit is composed of two EdgeConv (128 filters) layers that are connected with MLP layers (1024 filters). The input to the MLP layers is the concatenation of the outputs from the first two EdgeConv layers. The second unit is the same as the first unit. To regress the joint positions, we modify the last three fully connected layers into 1024, 512, $3 \times M$. Since we target the single human object, the transform network is considered to be very suitable. Although the point clouds converted from depth images are ordered, points become an unordered form after preprocessing. The spatial transform network aligns an input point set to canonicalize into a specific space by applying an estimated 3×3 transformation matrix. To estimate the 3×3 matrix, the network uses the coordinates of each point in the point cloud and the coordinate difference of its K neighbors. After matrix multiplication, the pose network takes the transformed point cloud as input.

C. Network Training

The process of training is composed of two parts. First, after preprocessing, the point cloud is sampled into a set of 5, 000 points. We trained the whole model with the randomly arranged point cloud as the input in each step. In this way, the spatial transform network is trained in a good situation. Second, we maintain the weights of the spatial transform network and train the remaining networks. The input to the model is also a set of 5,000 points but not randomly arranged every time. The input to the pose regression network is a set of normalized points $X^{nor} = \{x_i^{nor}\}_{i=1}^{N} = \{p_i^{nor}\}_{i=1}^{N}$, where p_i^{nor} is 3D coordinates of the normalized point and the ground truth $Y^{nor} = \{y_i^{nor}\}_{i=1}^{M} = \{j_i^{nor}\}_{i=1}^{M}$, where j_i^{nor} is the corresponding key body joints after normalization. Given T training samples with normalized point clouds, we minimize the following objective function:

$$\omega^* = \underset{\omega}{\operatorname{argmin}} \sum_{t=0}^{T} ||Y_t^{nor} - \digamma(X_t^{nor}, \omega)||^2 + \lambda||\omega||^2 \quad (3)$$

where ω denotes the parameters of the pose regression network, $\digamma$ represents the pose regression network, and λ is the regularization strength.

IV. Experiments

In this section, we evaluate our proposed approach on two public 3D human pose estimation datasets: EVAL [17] and ITOP [18], and then compare our method with state-of-the-art methods.

A. Implementation Details

All experiments were performed on a computer with one Intel Core i7-9700K CPU, dual Nvidia GTX1080 Ti GPUs with 24 GB memory, and 64 GB of RAM. The pose regression network and preprocessing were implemented by the TensorFlow framework and point cloud library (PCL). The optimizations of both training steps were Adam [21]. For the first step to train the spatial transform network, the initial learning rate was 0.001, and decay rates (β_1, β_2) were separately 0.9 and 0.999. Epsilon of Adam was $1e - 08$. The learning rate was divided by 10 after 50 epochs. After approximately 80 epochs,

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

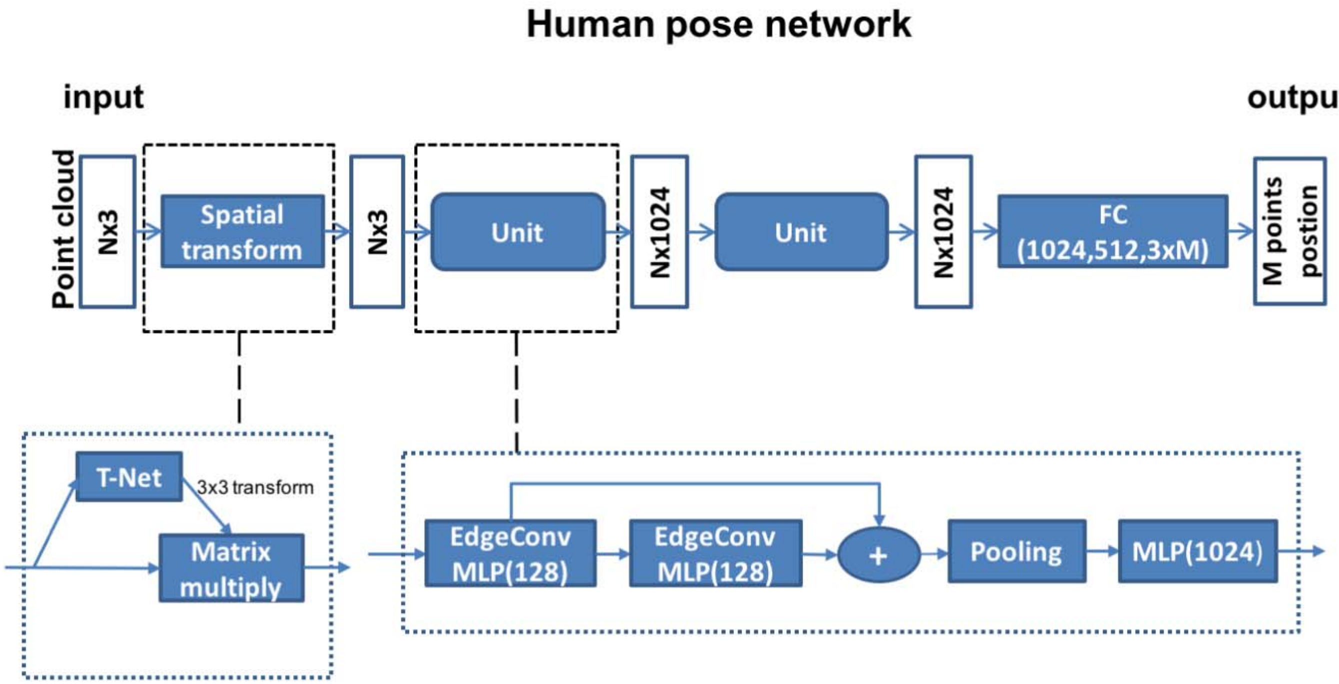


Fig. 4. The architecture of the pose regression network contains a spatial transform network and two units which are composed of EdgeConv layers. The normalized 3D point clouds are fed into the regression network. The dimension of input is the $N \times 3$ point clouds while the dimension of output is $M \times 3$. M represents the number of keypoint positions. The normalized 3D point clouds are input to $N \times 3$. $\oplus$ represents the process of concatenation. Spatial transform networks are part of PointNet [14], while EdgeConv is a part of the DGCNN [16]. FC is the fully connected layer. Our network is trained in an end-to-end manner to extract body features and regress 3D joint locations.

we stopped the first training and started the second training step. For the second step of training, the initial learning rate was 0.00001. Regularization strength was set to 0.0005. The number of neighbor points K was 16. All MLP and fully connected layers included the ReLU activation function, except the last layer. We trained the network with a batch size of 4. For a more in-depth training of the spatial transform network, we randomly sampled the data in each training step. When we performed the evaluation, the estimated 3D key body joint locations were reconstructed from the network outputs:

$$J = F(X_t^{nor}, \omega^*) \begin{bmatrix} b_w & 0 & 0 \\ 0 & b_h & 0 \\ 0 & 0 & 1 \end{bmatrix} + b_c \qquad (4)$$

B. Data Preparation and Evaluation Metrics

We evaluated the performance of our point cloud-based human pose regression network on the EVAL and ITOP datasets. Examples of ITOP and EVAL datasets are shown in Figure 5. Figure 5(a) represents the top-view of the ITOP dataset, while (b) and (c) represent the front-view of the ITOP dataset. Depending on the different datasets, we estimate the main 14 key body joints on the EVAL dataset and estimate 15 on the ITOP dataset. To produce the most convincing test results and verify the generalization ability of our model, all training and testing data are divided by the subjects.

The ITOP 3D human pose estimation dataset consists of two angles of view–front-view and top-view tracks. Each track contains a list of approximately 40 k training and 4 k testing depth images. The resolution of the image is 320 × 240 (width × height). This dataset consists of 20 actors who perform 15 sequences each and is recorded by two Asus Xtion Pro cameras. The ground truth of this dataset is the 3D coordinates of 15 body joints, including head, neck, shoulders, elbows, hands, torso, hips, knees, and feet. The 3D joint position is directly from the camera interface. After we removed invalid data, the dataset has approximately 17k and 4k depth data separately for training and testing. We take 3k of the training data for validation. Both front-view and top-view depth data with their corresponding joint data are used for evaluation.

The EVAL 3D human pose estimation dataset has 3 subjects: 1 female and 2 males. It contains approximately 10 k frames. Each frame is combined with the Vicon data of 30 markers. The resolution of the image is 320 × 240 (width × height). The dataset is divided into three parts (training, validation, and testing). We remove some invalid data that are out of the Vicon data. Finally, we have 8,158 front-facing depth images. We choose one subject for the test (approximately 2k frames) and the other two subjects (approximately 5k frames) for training, retaining approximately 800 frames as a validation dataset. The ground truth of this dataset is the 3D coordinates of 14 body joints, including chest, hips, shoulders, elbows, knees, feet, head, hands. Each joint is set up with different Vicon markers, chest with 1 marker, hip with 2 markers, shoulder with 2 markers, elbow with 2 markers, knee with 3 markers, feet with 3 markers, head with 1 marker, and hand with 3 markers. We select the average position of markers for each joint. Different from the ITOP dataset, the coordinates of joints are calculated from the Vicon system and markers. These joints are of high accuracy.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

TABLE I

PERFORMANCE COMPARISON OF THE PROPOSED METHOD WITH STATE-OF-THE-ART METHODS. (RF [7], RTW [8], IEF [19], VI [18]) ON THE ITOP DATASET AND EVAL DATASET

Dataset	ITOP(front-view)					ITOP(top-view)					EVAL		
Body part	RF	RTW	IEF	VI	Ours	RF	RTW	IEF	VI	Ours	RTW	VI	Ours
Head	63.8	97.8	96.2	**98.1**	96.73	95.4	**98.4**	83.8	98.1	96.13	90.9	**93.9**	88.93
Neck/Chest	86.4	95.8	85.2	97.5	**98.05**	**98.5**	82.2	50.0	97.6	97.61	87.4	94.7	**96.87**
Shoulders	83.3	94.1	77.2	**96.5**	94.38	89.0	91.8	67.3	**96.1**	93.08	**87.8**	87.0	86.14
Elbows	73.2	**77.9**	45.4	73.3	73.67	57.4	80.1	40.2	**86.2**	70.83	27.5	45.5	**75.11**
Hands	51.3	**70.5**	30.9	68.7	54.95	49.1	76.9	39.0	**85.5**	48.41	32.3	39.6	**63.07**
Torso	65.0	93.8	84.7	85.6	**98.35**	80.5	68.2	30.5	72.9	**95.58**	–	–	–
Hips	50.8	80.3	83.5	72.0	**91.77**	20.0	55.7	38.9	61.2	**84.50**	–	–	83.20
Knees	65.7	68.8	81.8	69.0	**90.74**	2.6	53.9	54.0	51.6	**79.19**	83.4	**86.0**	82.68
Feet	61.3	68.4	80.9	60.8	**86.30**	0.0	28.7	62.4	51.5	**67.76**	90.0	**92.3**	82.94
Upper Body	70.7	**84.8**	61.0	84.0	80.10	73.1	84.8	51.7	**91.4**	77.30	59.2	73.8	**79.31**
Lower Body	59.3	72.5	82.1	67.3	**89.60**	7.5	46.1	53.3	54.7	**77.15**	86.7	**89.2**	82.94
Mean	65.8	80.5	71.0	77.4	**85.11**	47.4	68.2	51.2	75.5	**78.46**	68.3	74.1	**80.86**

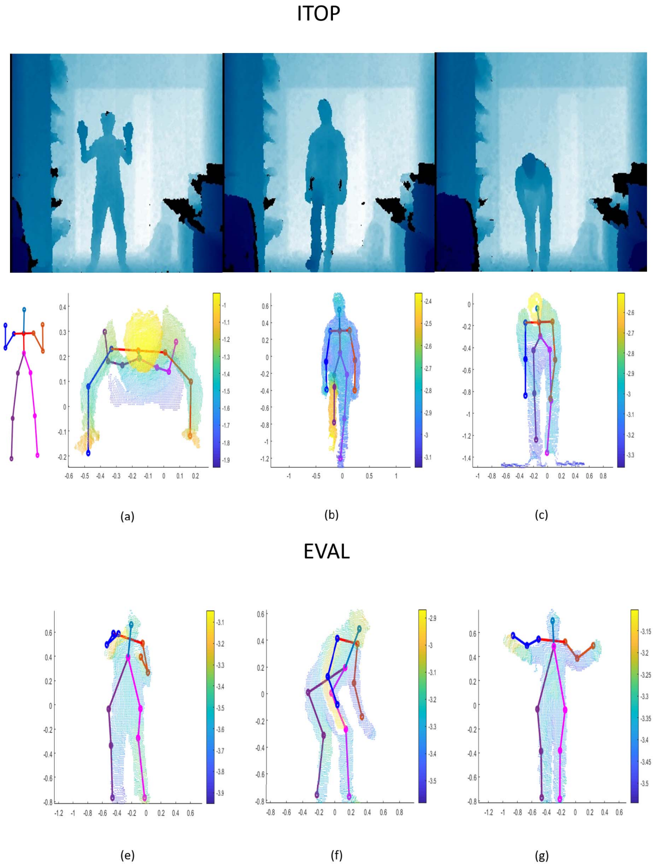


Fig. 5. An example of ITOP and EVAL dataset. There are 15 key joints in the ITOP dataset, which are head, neck, torso, shoulders, elbows, hands, hips, knees and feet. There are 14 key joints in the EVAL dataset, which are head, chest, shoulders, elbows, hands, hips, knees and feet.

We use the mean average precision (mAP) which is defined as the average precision for all human body parts. We set a $10cm$ rule following [8], [18] to show the result of each kind of joint, which means there is a successful detection when the predicted joints are less than $10cm$ from the ground truth. Here are the evaluation metrics.

$$AP(x, y) = \begin{cases} 1, & \text{if } distance(x, y) < 10cm \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

$$mAP = \frac{\sum_{i=0}^{M} AP(J_i, GT_i)}{M} \tag{6}$$

where $distance(x, y)$ denotes the Euclidean distance between x and y in 3D space; J and GT are predicted joints and ground truth separately. M is the total number of joints.

C. Comparison With State-of-the-Art Methods

We compared our proposed approach with state-of-the-art methods, which include random forest (RF) [7], the random tree walk algorithm (RTW) [8], iterative error feedback (IEF) [19], and the viewpoint-invariant feature-based method (VI) [18] on the ITOP dataset. Next, we conducted training on the EVAL dataset and compared our approach with RTW and VI. The mAP of the upper body and lower body is provided in Table I. The upper part of the body includes head, neck/chest, shoulders, elbows, and hands, while the lower body includes hips, knees, and feet. The qualitative results of our approach on the ITOP front-view, ITOP top-view, and EVAL datasets are shown in Figure 6. The motions in Figure 6 are turning around, standing, raising hands, boxing and shaking.

Most of the methods mentioned above performed well on the main parts of the human body, which are head, neck/chest, shoulders, and torso. These main parts provide the richest depth information. Predicting the 3D position of feet and hands is the hardest task in 3D human pose estimation since the proportion of hands and feet to the body is relatively small. In contrast to previous methods, our approach has excellent performance on the joints of the feet and knees. Some kinds of joints, such as knees, hands, elbows, hips, and feet, may be invisible from the view of the depth camera depending on the human motions, which results in deviation of the estimation. However, we still perceive from Table I that our method corresponds to this problem well.

For the front-view part of the ITOP dataset, RTW has the best score on estimating both upper body joints with a mAP of 84.8. Our predictions of other parts except hands are not much different from the state-of-the-art approaches because parts of the edge of the body, such as the hand, always convey limited depth information. The scores of RF, RTW, IEF, and VI are lower than ours in predicting feet and knees. Our approach has a high score with a mAP of 89.60 on predicting the lower body joints. From Table I, our method performs better than the other methods in predicting the whole body with a mAP of 85.11.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

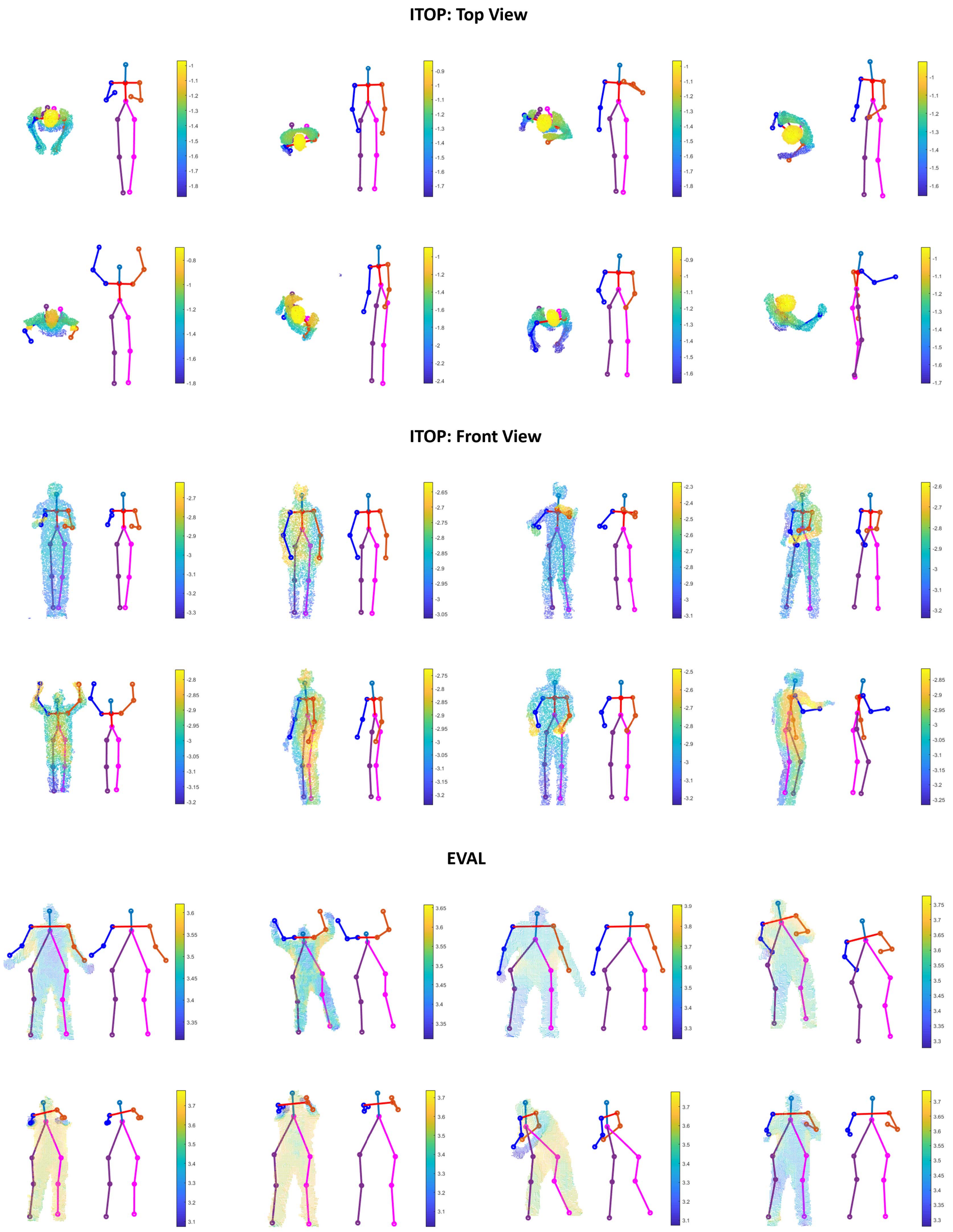


Fig. 6. Qualitative results of our pose net on EVAL and ITOP (both front-view and top-view) datasets. A segmented point cloud of a human is shown with its corresponding joints. The motions in ITOP are turning around, standing, raising hands and boxing, while the motions in EVAL are standing, shaking, and raising hand.

For the top-view of the ITOP dataset, the view angle of the collected dataset results in a considerable loss of depth information. The top-view of the ITOP dataset suffers more serious depth information loss than the front-view and EVAL. Figure 5 (a) can be taken as a representative example of such a situation. Therefore, among all experiments, the results from

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

the top-view part of the ITOP dataset are the least accurate. Although VI has the best score in the assessment of the upper body with a mAP of 91.4, our human pose net has a score with a mAP of 77.15, which is higher than other approaches. On estimating the whole body joints, our approach has a good mAP score performance of 78.46, which is higher than the second-ranking method VI. From the results of our approach, the estimation accuracy of the upper body part equals the lower part, and the main difference exists in the prediction of the hand part.

Although both ITOP and EVAL have front-view data, the EVAL dataset has more complex motion than ITOP. In the experiments on the EVAL dataset, VI achieved a higher score on the evaluation of the lower body with a mAP of 89.2. For the whole body, our approach was the first one with the whole body mAP of 80.86, which was much higher than the others (RTW and VI). This comes from the accurate estimations in the hand and elbow joints.

In our experiment, the most challenging joints are hand and elbow. The mean average precision from both datasets are below 75. The main reason is the lack of enough depth information. The motion range of both of the two joints is large, leading that the two parts beyond the capturing scope of the camera. Moreover, these two joints are easily occluded by other body parts from the view of the camera. Once these two parts are partially occluded, it results in that the extracted body parts in the point cloud image are not connected with each other and cause the depth information to be filtered out in the processing of segmentation.

D. Discussion

The application scenarios of our proposed approach are wide, including rehabilitation training, exercise coaching, sport player tracking, etc. In the rehabilitation training, most of the environments are controlled structured environments and the designed training is typically a slow tracking movement for a single patient. In this situation, our proposed approach well suits the task. In the exercise coaching, the movement can be fast which needs our proposed approach to be efficient enough to track the movement. In this case, we can tailor the model to be a light-weight simplified one by eliminating the neural links with small weights and increasing the number of sharing weights between clustered links. In the sport player tracking, the tasks are typically multi-player tracking which often involves occlusions. To overcome the occlusion issue, we can use maximum likelihood principle or posterior estimation to make correct joint association and pose estimation by fully utilizing the prior knowledge of the constraint between adjacent joints. More specifically, the occlusion can be modeled as a crossing of two segments based on computational geometry principle. Different combinations of movement sequences are then compared in the sense of probability by taking human movement restrictions into account [22].

V. Conclusion

We proposed a human pose model that estimates human pose from point clouds with a single depth image. A 2.5D depth image is converted to a 3D point cloud and processes it in the directly modified dynamic graph CNN network and PointNet to regress the 3D positions of joints. To handle the influence of background objects, we segment the point cloud of the human body using Euclidean cluster extraction. After being normalized by its height and width, the point cloud is fed into our modified dynamic graph CNN network. Experimental results compared to other state-of-the-art methods on two public depth-based datasets show that our method performs well for 3D human pose estimation. Our future work will be further improving accuracy for fast and occluded movements. Moreover, we will testify our approach in the task-oriented datasets, such as rehabilitation, excise coaching, and sport team players' tracking [23], [24].

References

[1] L. Yang, B. Yang, H. Dong, and A. E. Saddik, "3-D markerless tracking of human gait by geometric trilateration of multiple Kinects," *IEEE Syst. J.*, vol. 12, no. 2, pp. 1393–1403, Jun. 2018.

[2] Y. Zhang, J. Liu, and K. Huang, "Dilated hourglass networks for human pose estimation," in *Proc. Chin. Autom. Congr. (CAC)*, Nov. 2018, pp. 483–499.

[3] L. Ke1, M.-C. Chang, H. Qi, and S. Lyu, "Multi-scale structure-aware network for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 731–746.

[4] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3D human pose estimation in the wild: A weakly-supervised approach," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 398–407.

[5] M. Wang, X. Chen, W. Liu, C. Qian, L. Lin, and L. Ma, "DRPose3D: Depth ranking in 3D human pose estimation," in *Proc. 27th Int. Joint Conf. Artif. Intell.*, Jul. 2018, pp. 978–984.

[6] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[7] J. Shotton *et al.*, "Real-time human pose recognition in parts from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. CVPR*, Jun. 2011, pp. 1297–1304.

[8] H. Y. Jung, S. Lee, Y. S. Heo, and I. D. Yun, "Random tree walk toward instantaneous 3D human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 2467–2474.

[9] H. Guo, G. Wang, X. Chen, C. Zhang, F. Qiao, and H. Yang, "Towards good practices for deep 3D hand pose estimation," in *Proc. IEEE Int. Conf. Image Process.*, Jul. 2017, pp. 4512–4516.

[10] J. Y. Chang, G. Moon, and K. M. Lee, "V2 V-PoseNet: Voxel-to-Voxel prediction network for accurate 3D hand and human pose estimation from a single depth map," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Jun. 2018, pp. 5079–5088.

[11] L. Yang, L. Zhang, H. Dong, A. Alelaiwi, and A. E. Saddik, "Evaluating and improving the depth accuracy of Kinect for Windows v2," *IEEE Sensors J.*, vol. 15, no. 8, pp. 4275–4284, Mar. 2015.

[12] C. Chen, R. Jafari, and N. Kehtarnavaz, "A real-time human action recognition system using depth and inertial sensor fusion," *IEEE Sensors J.*, vol. 16, no. 3, pp. 773–781, Feb. 2016.

[13] B. Nagy and C. Benedek, "3D CNN-based semantic labeling approach for mobile laser scanning data," *IEEE Sensors J.*, vol. 19, no. 21, pp. 10034–10045, Nov. 2019.

[14] R. Q. Charles, H. Su, M. Kaichun, and L. J. Guibas, "PointNet: Deep learning on point sets for 3D classification and segmentation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 77–85.

[15] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 5099–5108.

[16] Y. Wang, Y. Sun, Z. Liu, S. E. Sarma, M. M. Bronstein, and J. M. Solomon, "Dynamic graph CNN for learning on point clouds," *ACM Trans. Graph.*, vol. 1, no. 1, pp. 1–13, 2019.

[17] V. Ganapathi, C. Plagemann, D. Koller, and S. Thrun, "Real-time human pose tracking from range data," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 738–751.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.

[18] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, “Towards viewpoint invariant 3D human pose estimation,” in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 160–177.

[19] J. Carreira, P. Agrawal, K. Fragkiadaki, and J. Malik, “Human pose estimation with iterative error feedback,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2016, pp. 4733–4742.

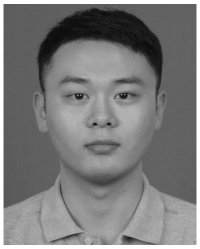
[20] P. M. Chu, S. Cho, Y. W. Park, and K. Cho, “Fast point cloud segmentation based on flood-fill algorithm,” in *Proc. IEEE Int. Conf. Multisensor Fusion Integr. Intell. Syst. (MFI)*, Nov. 2017, pp. 656–659.

[21] D. P. Kingma and J. L. Ba, “Adam: A method for stochastic optimization,” in *Proc. Int. Conf. Learn. Representations*, 2015, pp. 1–15.

[22] B. Yang, H. Dong, and A. El Saddik, “Development of a self-calibrated motion capture system by nonlinear trilateration of multiple Kinects v2,” *IEEE Sensors J.*, vol. 17, no. 8, pp. 2481–2491, Apr. 2017.

[23] M. Capecci *et al.*, “The KIMORE dataset: KInematic assessment of MOvement and clinical scores for remote monitoring of physical REhabilitation,” *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 27, no. 7, pp. 1436–1448, Jul. 2019.

[24] F. Negin, F. Özdemir, K. A. Yüksel, C. B. Akgül, and A. Ercil, “A decision forest based feature selection framework for action recognition from RGB-Depth cameras,” in *Proc. 21st Signal Process. Commun. Appl. Conf. (SIU)*, Apr. 2013, pp. 648–657.



Yufan Zhou received the B.Eng. degree in railway traffic signaling and control from Southwest Jiaotong University, China, in 2017. He is currently pursuing the M.A.Sc. degree in electrical and computer engineering with the University of Ottawa. His research interests include artificial intelligence and multimedia.

Haiwei Dong (Senior Member, IEEE) received the Dr.Eng. degree in computer science and systems engineering from Kobe University, Kobe, Japan, and the M.Eng. degree in control theory and control engineering from Shanghai Jiao Tong University, Shanghai, China, in 2008 and 2010, respectively. He was a Research Scientist with the University of Ottawa, Ottawa, ON, Canada, a Postdoctoral Fellow of New York University, New York City, NY, USA, a Research Associate with the University of Toronto, Toronto, ON, Canada, and a Research Fellow (PD) of the Japan Society for the Promotion of Science, Tokyo, Japan. He is currently a Principal Engineer with Huawei Technologies Canada, Ottawa, and a licensed Professional Engineer in Ontario. His research interests include artificial intelligence, robotics, and multimedia.

Abdulmotaleb El Saddik (Fellow, IEEE) is a Distinguished University Professor and the University Research Chair of the School of Electrical Engineering and Computer Science, University of Ottawa. He has coauthored ten books and more than 550 publications, and chaired more than 50 conferences and workshops. He has received research grants and contracts totaling more than 20 M. His research focuses on the establishment of digital twins to facilitate the well-being of citizens using AI, the IoT, AR/VR, and 5G, hence allowing people to interact in real-time with one another as well as with their smart digital representation. He has supervised more than 120 researchers and received several international awards, among others. He is an ACM Distinguished Scientist, and a Fellow of the Engineering Institute of Canada and the Canadian Academy of Engineers. He has received the IEEE I&M Technical Achievement Award, the IEEE Canada C. C. Gotlieb (Computer) Medal, and the A. G. L. McNaughton Gold Medal for important contributions in the fields of computer engineering and science.

Authorized licensed use limited to: FUDAN UNIVERSITY. Downloaded on May 05,2022 at 11:55:55 UTC from IEEE Xplore. Restrictions apply.