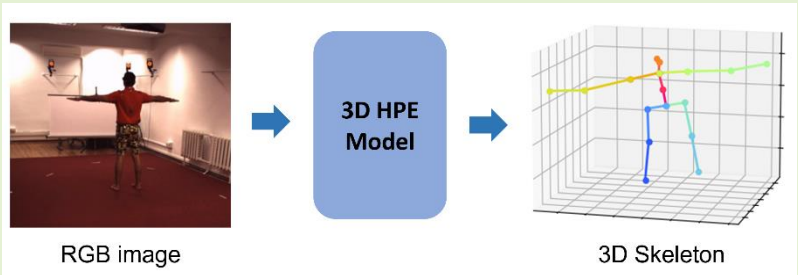
Monocular Based 3D Human Pose Estimation with Refinement Block and Special Loss Function

Tsung-Han Tsai, Senior Member, IEEE, and Yi-Jhen Luo

***Abstract*—Recently, 3D human pose estimation (HPE) from a monocular RGB image has attracted much attention following the success of a deep convolution neural network. Many algorithms take 2.5D heatmaps as the 3D coordinate, whose X-axis and Y-axis correspond to the image coordinate, and the Z-axis corresponds to the camera coordinate. Therefore, the camera matrix or the distance between the root skeleton and the camera (the ground-truth information) is usually adopted to transform the 2.5D coordinate to 3D space. Since 2.5D heatmaps ignore the conversion between 2D and 3D positions, they lose some conversion features and limit their applicability in the real world. In this paper, we present an end-to-end framework that can utilize the contextual information in RGB images to directly predict 3D space skeletons from a monocular image. Specifically, we use the multi-loss method that depends on 2D heatmaps and volumetric heatmaps and a refinement block to locate the root-relative 3D human pose. Our approach takes 2D heatmaps and volumetric heatmaps as features to compute the loss and combine the loss from relative 3D locations to generate the total loss. The model can learn the 2D heatmap feature and 3D location jointly and focus on the root-relative 3D position in the camera coordinate. The experimental result shows that our model can predict relative 3D human pose well on Human3.6M.**



***Index Terms*—3D human pose estimation, deep convolution neural network, root-relative 3D human pose, volumetric heatmap**

I. Introduction

THIS purpose of 3D human pose estimation (HPE) is to locate single-person or multiple-person skeletons in 3D coordinates. In recent years, 3D HPE has progressed significantly under the great development of neural networks. Estimating the accurate 3D human pose from a monocular image attracts lots of attention because HPE is important in many applications, such as action recognition [1], human-computer interaction [2], surveillance [3], and sports analysis [4]. However, it is a challenging task due to the inherent ambiguity of the skeleton, the self-occlusion, or the occlusion by other objects with various human poses. Traditional pose estimation methods use specialized equipment or wearable devices with high-precision systems to mark human skeletons. These methods require complex setup processes and incur high costs, limiting the application of skeletal tracking. Besides wearable devices, various sensors such as radio frequency (RF), radar [5], and RGB cameras are also used in this field. However, we prefer using RGB cameras because they are cost-effective. Compared to RF and IR sensors, RGB cameras are cheaper, making them suitable for large-scale applications and allowing for more economical system deployment and expansion. Additionally, RGB cameras have the advantage of widespread availability. They are commonly found in devices such as smartphones, laptops, and surveillance cameras. This widespread presence allows us to utilize existing hardware resources for pose estimation without needing to purchase specialized sensors, thus reducing overall costs and deployment complexity.

HPE can mainly be divided into two types, one is to estimate 2D pose [6], [7] and the other is to estimate 3D pose [8]. Although the difference between these two types is only the dimension issue, the 3D human pose estimation is more difficult than the 2D human pose estimation. In the 3D processing

domain, the spatial relationships in the depth axis are difficult to express on a plane. It can utilize a camera matrix (including camera extrinsic and camera intrinsic parameters) to convert image coordinates and camera coordinates to each other, whose coordinates are completely different. Because the camera matrix of each RGB image is different, the range of camera coordinate distribution is large. Therefore, this greatly increases the difficulty of estimating 3D coordinates.

As for 3D human pose estimation, there are single-view methods and multi-view methods [9]. The single-view method means that it takes one monocular image as input. It only adopts one camera image and outputs the human skeleton position. The multi-view method means that it will have a multi-camera system, and capture synchronized images from each camera. Then, the model will output the human skeleton in each view

and overlap the skeleton to calculate the 3D human skeleton. The main disadvantage of the multi-view method is the requirements of specific devices to establish the multi-camera system. Besides, the multi-camera system is assumed to be synchronized and calibrated in most multi-view approaches.

With the rapid growth of the convolution neural network (CNN), there are many methods to estimate 2D or 3D human pose based on deep neural network (DNN). Most of the CNN methods use 2D heatmaps and volumetric heatmaps as features to represent the possible locations of a human skeleton and then regress these features to 3D spatial coordinates by convolution neural network.

In this paper, we propose the multi-loss method combined with 2D heatmaps to construct a 3D HPE. We use the volumetric heatmaps and root-relative camera coordinates to locate relative space coordinates based on the CNN method. The meaning of root-relative coordinates is to subtract the root coordinate from each skeleton joint, with most root coordinates being the central skeleton joints of the human body, such as the chest or pelvis. First, calculate the depth of the root skeleton joint and the value of the root-relative coordinates. Then, by adding the relative value to the root coordinate, the absolute coordinates in space can be obtained. Finally, the depth of each coordinate is determined using the known camera distance. Although volumetric heatmaps have been adopted in many algorithms, few methods focus on the loss generated by volumetric heatmaps. We take the loss from volumetric heatmaps and combine it with pixel loss and camera coordinate loss. Also, we present a refinement block to fine-tune the 3D skeleton. We evaluate this model on the Human3.6M dataset and achieve a better result than other algorithms. It means that the multi-loss and refinement blocks are efficient methods for 3D human pose estimate.

The rest of the paper is as follows: Section 2 introduces related work; Section 3 presents the proposed method; Section 4 presents the experiment result and Section 5 concludes this paper.

II. Related Work

Human pose estimation, both in 2D and 3D, has been greatly studied in these years as it is useful for many applications. There are many solutions to locate the position of the human skeleton. In this section, we will introduce deep learning approaches for 2D HPE and 3D HPE.

A. 2D human pose estimation

DeepPose [10] is the first paper that utilized a deep neural network (DNN) to estimate 2D human pose. It takes 2D human pose estimation as a regression problem and defines how to use DNN for 2D HPE. The overall model of DeepPose only includes convolution operation and fully connected layers. Furthermore, the authors adopt a cascaded DNN-based pose detector. The concept of cascading also influences the latter methods, such as stacked hourglass networks [11]. Stacked hourglass networks process features between different scales to capture spatial relationships between joints efficiently. Besides, the stacked architecture integrates the information between different scales and refines the skeleton position well. A cascaded pyramid network (CPN) [12] uses the concatenation of a variety of feature maps of different scales to learn the human keypoints. The method of CPN integrates rich feature information, including high-level feature maps with low resolution and low-level feature maps with fewer features but high image resolution to obtain a target detection system with accurate identification and positioning.

Bottom-up and top-down methods all can predict multi-person 2D skeletons well. The bottom-up approach localizes the skeleton of all subjects and then associates them with individuals. It is usually used for multi-person pose estimation. The bottom-up method [13] utilizes part affinity fields (PAF) to find out all skeletons in the picture and group them to output an individual subject. PAFs are 2D vectors that can link each joint to its parent joints. Through PAFs, the neural network can quickly and greedy part association to achieve real-time multi-person 2D pose estimation. [14] is another bottom-up method that directly regresses human keypoints instead of detecting each keypoint and grouping them into individuals. It presents a disentangled keypoint regression (DEKR) to estimate human pose. Different from the bottom-up approach, the top-down approach [15] crops the people from an image first and then uses heatmaps to match the skeleton position. HRNet [16] is another top-down method that maintains the high-resolution branch. And it achieves rich high-resolution representation by repeat fusing each of the high-to-low resolution features. Therefore, the predicted keypoint is potentially more accurate and more precise in spatial. Heatmap matching is another popular and effective solution to obtain skeleton features. It gives a probability to each pixel position in the picture, which represents the probability of the skeleton. Compared to regressing the skeleton coordinate directly, heatmap matching is easier to converge and heatmap matching can be simulated by the corresponding function. In addition, Gaussian heatmaps can provide a better correlation between joints in pixel coordinates resulting in higher precision results [17].

B. 3D human pose estimation from single-view

In 2D HPE, most of the RGB images are used as input and the heatmaps are used to directly regress X and Y coordinates in the plane coordinates of the same dimension. However, in 3D HPE, it is rare to directly regress the RGB image to the camera coordinate, because there is an additional Z-axis spatial coordinate. There are many approaches for 3D HPE, including the top-down and bottom-up methods. Besides, it also can be distinguished by input. It can be divided into multi-view images single-view images or video as the input of the neural network. The multi-view method takes RGB images from multiple angles as input and outputs a set of 3D skeletons. However, the multi-view method is not easy to adapt to most applications. Using video as input means that the time sequence is used as the input of the neural network, and the output is a series of joints. Here we mainly discuss the method of generating a 3D skeleton with a single view.

The 3D pose estimation by single-view image is divided into the end-to-end method and the two-stage lifting method. The end-to-end method can directly predict skeleton position through neural network operations without generating 2D coordinates in the middle. One representative work is [18], which uses hourglass architecture to gradually increase the

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3510728

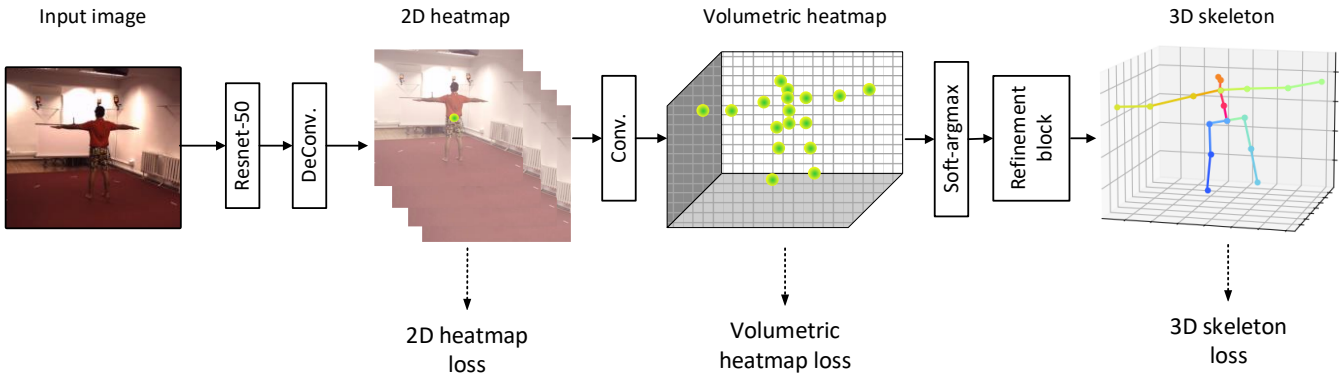


Fig. 1. The overall neural network architecture of proposed method. It contains a ResNet-50 architecture to extract RGB image feature, a set of deconvolution layer and convolution layer to scale the small dimension feature, and utilize soft-argmax and refinement block to generate 3D human skeleton.

resolution of the depth dimension and finally regress to the spatial coordinates. [19] is a framework for training with root and root-relative coordinates respectively, and uses the camera parameters to return the skeleton coordinates. [20] present a tiny-HourglassNet that can estimate 3D human pose with smaller hourglass architecture and guarantee the performance. It combines two types of ShuffleNet blocks and develops two feature enhancement modules to improve the accuracy of 3D human pose. [21] present a volume representation to transform the highly nonlinear 3D coordinate regression problem into a prediction problem form in discrete space. The voxel likelihood of each joint in the volume is predicted by the convolutional network. The ordinal depth relationship of the human joints is used to alleviate the need for accurate 3D ground truth poses. Although it is feasible to directly predict coordinates in space, this still is a challenging problem because the mapping of RGB images to 3D space is a highly nonlinear and difficult problem.

The two-stage lifting method is to predict the 2D skeleton and then convert it to 3D space. [22] proves that it is possible to convert from planar joints to spatial coordinates. It directly inputs 2D coordinates and then maps them to three-dimensional space through the residual hierarchy. [23] applies CNN architecture to generate 2D skeletons and 3D skeletons sequentially through a complex network. [24] proposed a new transform re-projection loss, which is an effective method to explore consistency from different views for training the 2Dto-3D lifting network. It only input multi-view during training, and input single-view in inference time. [25] focused on learning mapping 2D pose to 3D pose and it used the SMPL model [26] as an intermediate feature to suppress unreasonable 3D pose prediction. Specifically, it regresses the parameters of the low-dimensional SMPL model which are used to compose a 3D pose. This kind of lifter module largely relies on the accuracy of the 2D skeleton. Once a wrong 2D skeleton is generated, it will cause serious damage to the subsequent 2D-3D lifter.

C. 3D human pose estimation from multi-view

The 3D pose estimation from a multi-view means that there are multiple camera views to capture images. In general, multi-view 3D HPE combines the information and features from 2D images to generate a 3D skeleton. [27] present new solutions for multi-view 3D HPE based on learnable triangulation. The volumetric triangulation can improve the performance of multi-view pose estimation. [28] present a new fusion algorithm to combine 2D keypoints from different camera views and lift to 3D coordinate with a differentiable Direct Linear Transform (DLT) layer. This method reduces the computational complexity and achieves real-time 3D pose estimation from multiple cameras. VoxelPose [29] proposed a method to directly infer in the 3D coordinate instead of solving the challenging association problems in the 2D space. This method can estimate the human pose stably in the scenes with a lot of occlusions.

Although the multi-view approaches can estimate the 3D pose well, this method usually requires an advanced multi-view environment and a camera synchronization system. In general scenarios, such settings are rarely complete. In most applications, there is only a single view. Multi-view methods provide another way to generate a 3D skeleton, but at the same time, it also has many restrictions, which greatly affect the generality of this method.

III. Proposed Method

To our knowledge, there are seldom methods that combine 2D heatmaps and volumetric heatmaps as feature loss. We especially focus on the volumetric heatmaps and adopt combinational loss to calculate the total loss. In summary, we present an end-to-end approach based on multi-loss and refinement blocks to estimate 3D human pose from a sing-view RGB image. In this section, we explain the proposed method in detail.

A. Network architecture

3D HPE mainly outputs the coordinates of the human body in 3D space. Since we aim to perform single-person pose estimation from a monocular RGB image, we only need to focus on the relationships between the skeletons in a person. It is natural to estimate the root-relative 3D position.

Our network architecture is shown in Fig. 1. Initially we input an RGB image I and use ResNet-50 to extract the features from the input. Because there are only some positions in an image that we need to focus on, we encode the RGB image to a smaller size and then decode it to the same size as the 2D heatmaps through the deconvolution layers. There are efficient residual architectures in ResNet-50. Therefore, we can obtain robust 2D heatmap features at the beginning. After ResNet-50, we utilized three de-convolutional layers to decrease the

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3510728

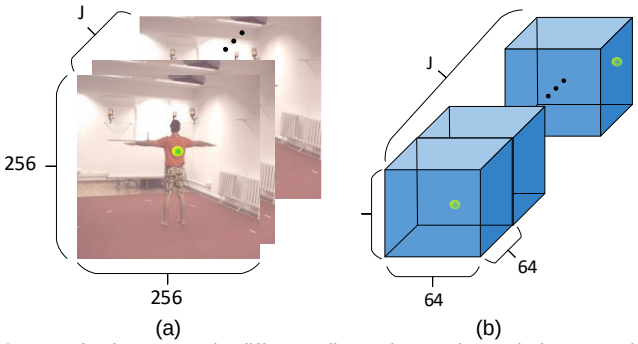


Fig. 2. The heatmaps in different dimension. Volumetric heatmap has depth dimension of z axis. (a) 2D heatmap (b) Volumetric heatmap.

channel number and increase the feature map size to fit the dimensions of 2D heatmaps. Next, instead of directly outputting the 2D heatmaps into X and Y coordinates, we keep the original 2D heatmaps information and use convolution operations to lift our features to three-dimensional space.

As shown in Fig. 2., the initial 2D heatmaps $H_{2D} \in \mathbb{R}^{J\times h\times w}$ consists of J joint coordinates where J is the total number of joints 18 (we add another joint as thorax between the left shoulder and right shoulder as an additional joint manually). Fig. 3. shows the 17 skeleton joints. Through a series of convolution operations, we can resize the feature map to $H_{3D} \in \mathbb{R}^{J\times d\times w\times h}$, where d is the dimension along the depth axis of the volumetric heatmaps of 64 in our method. We expand the number of channels to $J \times d$ to represent the depth in 3D space. Then we can predict the human pose of X, Y, and Z coordinates in the camera coordinate through the soft-argmax, and finally, use the refinement block as shown in Fig. 4. to calibrate the generated 3D coordinates.

B. Volumetric heatmaps

Assume X, Y, and Z are random vectors corresponding to the x, y, and z coordinates of the predicted particular human pose joint in 3D space. By thinking of a voxelization of X, Y, and Z coordinates, we call it a volumetric heatmap with the size $D \times H \times W$, where D is the depth dimension, while H and W represent the height and width of the image respectively. The volumetric heatmaps represent the confidence map of the probability of the human joints. Through the soft-argmax calculation of pose estimation, we can transmit the volumetric heatmaps into estimated joints, which is the expectation of the random vectors. All values in the volumetric heatmaps must be positive to present the probability. To create the volumetric feature, we utilize convolution to enhance the feature dimensions.

The advantage of using volumetric heatmap representation is that it transforms highly non-linear problems into simple predictive problems in discrete spaces. In other words, human pose estimation does not predict the position of the skeleton directly, instead predicts per voxel confidence, which makes it easier for neural networks to learn the target function. It is a difficult task for deep neural networks to regress the skeleton position directly in the image because the connection between each skeleton is not as tight as general pixel-to-pixel image tasks. Therefore, it is more intuitive to adopt volumetric heatmaps for the 3D HPE task.

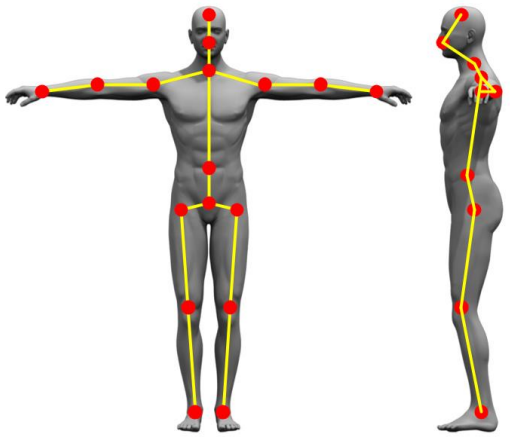


Fig. 3. Skeleton definition and connection relationship. The amount of skeleton is 17 provides from Human3.6M as usual.

Although volumetric heatmap is useful in 3D HPE, it still causes another problem. The major drawback of volumetric representation is the amount of computation and memory, leading to some limitations during implementation. For example, we utilize smaller heatmap resolution which has low quantization errors, or complex training strategies with coarse-to-fine predictions through the refining of network outputs.

C. Refinement block

Although the soft-argmax can express the 3D joints from a volumetric heatmap, the predicted 3D coordinate outputs are not accurate enough. Because soft-argmax directly converts the low-resolution voxel probability into the 3D coordinate, it will lose most information, resulting in inaccurate skeleton points. Therefore, we propose a refinement block to refine the predicted skeleton generated from the soft-argmax function to obtain a more accurate result.

Fig. 4. shows the architecture of the refinement block. After the soft-argmax initially generates the 3D skeleton position, we use the refinement block to make the skeleton more accurately represented in 3D space. First, we flatten the 3D skeleton into a representation vector of dimension $d = 54$ (18×3 joints) and then refine the predicted skeleton by fully-connected layers to generate a vector of dimension $d = 512$. After each fully-connected layer, we apply the batch normalization (BN in Fig. 2.), ReLU, and dropout, with parameter 0.5 and 2 residual blocks to regress the feature to the 3D skeleton position. Although the operation of the fully connected layer is very simple, we found that this can effectively calibrate the predicted human skeleton in camera coordinates. It means that we can predict 3D human pose more precisely through the simple refinement block.

D. Combinational loss function

In terms of calculating loss, we use mean square error (MSE) to calculate multiple losses for different features to compose the final required loss. In general, the 3D coordinate loss is adopted in the pose estimation task. Some algorithms also include loss from 2D heatmap in the loss calculation. Most of the methods use the 2D loss to estimate position in image coordinates and then convert it to camera space. Despite our main task being to estimate the human skeleton in camera space and we do not

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and
content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3510728

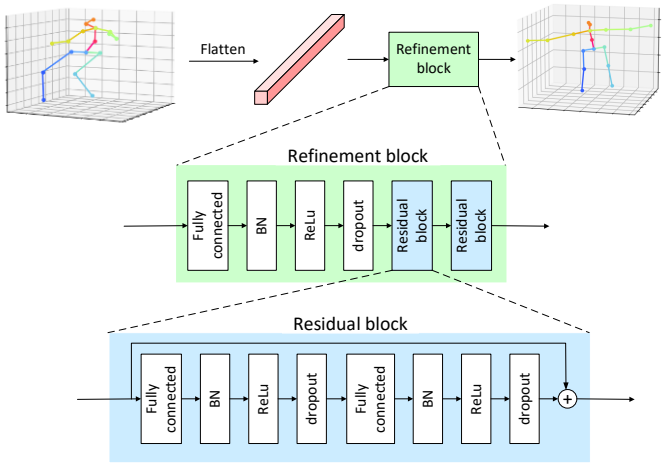


Fig. 4. The operation after soft-argmax and the detail of refinement block. The top: Flatten all joints and input them to a refinement block. The middle: The architecture of the refinement block. The bottom: The architecture of residual block.

generate image coordinates, we still adopt 2D loss from the 2D heatmap. There is a translation between the image coordinate and camera coordinate, as shown in (1), x_i and y_i are the x and y coordinates in the image coordinate, x_c, y_c and z_c are the x, y, and z coordinates in the camera coordinate(root-relative skeleton coordinate), and F and C are the focal length and principal point coordinates, respectively. We utilize 2D heatmaps to produce volumetric heatmaps, so we retain the 2D feature which has rich feature information, and calculate the 2D loss. Besides, we also take the loss from volumetric heatmaps which can present the space relationship as well as our loss. As for the 3D coordinate loss from the 3D skeleton, we multiply it by a parameter λ. Because the skeleton position is our final target output, we set a weight to increase the importance of the calculation process.

We will explain in detail how the volumetric heatmap is calculated. First, we convert the camera coordinate in the dataset into image coordinates through the camera parameters by (1) and then generate 2D Gaussian heatmaps. As shown in (2), H_{2D} is the 2D heatmap ground truth, and $\widehat{H}_{2D}$ is the predicted 2D heatmap with the size of $J \times h \times w$, where J is the number of joints, h, and w are the height and width of 2D heatmaps. Besides, we also calculate the volumetric loss from volumetric heatmaps as shown in (3). In (3), H_{3D} is the volumetric heatmap ground truth, and $\widehat{H}_{3D}$ is the predicted volumetric feature with the size of $J \times d \times h \times w$, where d is the depth axis we defined. Finally, we keep the initial values of the camera coordinates to calculate loss with the features we refined in (4). K is the ground truth of the 3D skeleton, and $\widehat{K}$ is the predicted skeleton. As in (5), our total loss L_{total} is composed of three kinds of loss.

It is hard for neural networks to regress the human skeleton position directly because it is a non-linear problem. In most studies, volumetric heatmap loss is not specifically calculated. However, we found that the volumetric heatmap can represent the correlation between the depth axis well and it can solve the inherent non-linear problem of 3D HPE. Accordingly, we employ the volumetric heatmap as one of the features to be computed. In addition to the common loss term (2D heatmap loss and 3D coordinate loss), we further utilize volumetric heatmap loss which is rich in spatial information and translates the non-linear problem into a probability problem. Finally, we combine these loss terms and take L_{total} in (5) as our neural network goal. In summary, we especially calculate the loss of the volumetric heatmap to have better results.

$$x_i = \frac{x_c}{z_c} \times F + C, \quad y_i = \frac{y_c}{z_c} \times F + C \tag{1}$$

$$L_{2DHM} = \frac{1}{J}\sum_i^J \left\| {H_{2D}}^i - {\widehat{H}_{2D}}^i \right\|^2 \tag{2}$$

$$L_{3DHM} = \frac{1}{J}\sum_i^J \left\| {H_{3D}}^i - {\widehat{H}_{3D}}^i \right\|^2 \tag{3}$$

$$L_{3DPOSE} = \frac{1}{J}\sum_i^J \left\| K^i - \widehat{K}^i \right\|^2 \tag{4}$$

$$L_{total} = L_{2DHM} + L_{3DHM} + \lambda * L_{3DPOSE} \tag{5}$$

*L_{2DHM}: 2D heatmap loss
*L_{3DHM}: volumetric heatmap loss
*L_{3DPOSE}: 3D coordinate loss

IV. Experiment Results

A. Humam3.6M dataset

We validate our approach on the Human3.6M dataset [30] which is a large dataset including 3.6 million single-person RGB images with accurate 3D human skeletons annotated by a high-speed motion capture system. There are 11 subjects in total (6 males and 5 females,) and we generally select subjects 1, 5, 6, 7, and 8 as training data, 9, and 11 as testing data. Besides, there are 15 action annotations such as walking, eating, directions, making a phone call, etc. in this dataset. For each action, accurate 2D and 3D skeleton locations and the camera parameters are provided. Furthermore, they also include a multi-view of the human pose. In this paper, we only take a 3D skeleton from the dataset.

In testing time, we do not utilize 2D skeleton position and camera parameters. The 3D pose estimation containing 32 joints is to be applied in the Human3.6M dataset. We follow the standard training and testing strategies and we adopt a 17-joint skeleton as our estimated skeleton target. In the testing stage, we calculate all joints except “thorax”, which is added at training. We use the standard protocol for the evaluation as usual. Protocol-I is the mean per-joint position error (MPJPE), in millimeters which is the mean Euclidean distance between predicted joint positions and ground-truth joint positions. In (6), N_S is the number of joints in skeleton, m_p is the predicted skeleton, and m_{gt} is the skeleton ground truth. Protocol-II employs a rigid alignment to the estimated pose first, then computes the MPJPE. Protocol-I directly measures the model error in the camera coordinate system, suitable for raw predictions; whereas Protocol-II performs rigid alignment (including translation and rotation) before measurement, eliminating viewpoint effects and more accurately assessing the model's generalization across different positions.

$$MPJPE = \frac{1}{N_S}\sum_{i=1}^{N_S} \left\| m_p(i) - m_{gt}(i) \right\|_2 \tag{6}$$

B. Training and Implementation

During the training stage, we use the common random color jitter on RGB images for the pre-processing strategy. We use camera parameters to convert the 3D skeleton into a 2D skeleton to generate a 2D heatmap and utilize a 3D skeleton to

This article has been accepted for publication in IEEE Sensors Journal. This is the author's version which has not been fully edited and content may change prior to final publication. Citation information: DOI 10.1109/JSEN.2024.3510728

TABLE I
COMPARING MPJPE AND P-MPJPE VALUES BETWEEN DIFFERENCE APPROACHES ON HUMAN3.6M USING RGB IMAGE AS INPUT. BEST RESULTS ARE SHOW IN **BLOD**.

	Dir.	Diss.	Eat.	Gre.	Phn.	Pose	Pur	Sit.	SitD.	Smo.	Pht.	Wait	Walk	WD.	WT.	Protocol-I Avg..	Protocol-II Avg.
[31]	71.6	**66.6**	74.7	79.1	70.0	67.6	89.3	90.7	195.6	83.5	93.2	71.2	55.7	85.9	62.5	82.7	114.2
[32]	62.8	69.2	79.6	78.8	80.8	72.5	73.9	96.1	106.9	88.0	86.9	**70.7**	71.9	76.5	73.2	79.5	97.5
[33]	62.6	78.1	63.4	72.5	88.3	63.1	74.8	106.6	138.7	78.8	93.8	73.9	55.8	82.0	**59.6**	80.5	NA
[34]	69.2	75.2	75.8	73.6	75.4	99.6	76.1	**73.6**	**75.0**	109.6	**73.7**	88.9	71.8	**55.6**	73.5	77.8	NA
[35]	68.4	77.3	70.2	71.4	75.1	86.5	69.0	76.7	88.2	103.4	73.8	72.1	83.9	58.1	65.4	76.0	NA
[36]	77.5	85.2	82.7	93.8	93.9	101	82.9	102.6	100.5	125.8	88.0	84.8	72.6	78.8	79.0	89.9	65.1
[37]	70.4	83.6	76.6	77.9	85.4	106.1	72.3	102.9	115.8	164.9	82.4	74.3	94.6	60.2	70.7	88.8	NA
[38]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	120.9	90.8
[39]	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	97.8	86.2
OURS	**58.7**	71.3	**58.4**	**67.7**	**71.8**	**62.2**	**66.6**	90.04	128.4	**66.32**	94.88	72.63	58.4	84.96	73.2	**74.7**	**65.1**

*Note: NA means these metrics are not provided by the original paper with the source code

TABLE II
THE ABLATION STUDY ON HUMAN3.6M WITH DIFFERENT MODULES. A: BASELINE WITHOUT VOLUMETRIC HEATMAP LOSS AND REFINEMENT BLOCK. B: ADD VOLUMETRIC HEATMAP LOSS. C: ADD REFINEMENT BLOCK. PROTOCOL-I AND PROTOCOL-II ARE IN MM.

Method	Protocol-I	Protocol-II
A	142.23	123.02
A + B	100.24	90.95
A + C	135.8	110.85
A + B + C	74.73	65.1

TABLE III
THE NUMBER OF PARAMETERS AND COMPUTATIONS IN THE NEURAL NETWORK. A: BASELINE WITHOUT VOLUMETRIC HEATMAP LOSS AND REFINEMENT BLOCK. B: ADD VOLUMETRIC HEATMAP LOSS. C: ADD REFINEMENT BLOCK.

Method	Parameters	Operations (FLOPs)
A	24.46M	7.00045G
A + B	24.46M	7.00045G
A + C	25.5M	7.00156G
A + B + C	25.5M	7.00156G

generate a volumetric heatmap. We take 256×256 as the size of the input image. In terms of loss functions, as we mentioned, L_{total} is our final loss which is composed of multiple losses. As described in Section 3, our 3D HPE network is ResNet-50, and other modules, ResNet-50 are mainly to learn the image feature, and the following modules are computing the 3D human skeleton position and refining the prediction. About the implementation details, we train our model on NVIDIA Geforce RTX 3080Ti GPU with PyTorch deep learning framework. And we train 25 epochs with a batch size of 32. We set a learning rate of 0.001 at the beginning, and then at epoch 11 the learning will decrease by a factor of 10 with the Adam optimizer. Besides, our model can run at 48 frame-per-second (fps) and an execution time of 20.1 ms on an NVIDIA Geforce RTX 3080Ti GPU.

TABLE IV
ABLATION STUDIES FOR DIFFERENT HYPERPARAMETERS DURING NETWORK LEARNING. WEIGHT DENOTES THE WEIGHT COEFFICIENT λ IN (5). THE MPJPE ON HUMAN3.6M WITH DIFFERENT λ.

Weight	MPJPE (mm)
10	91.7
15	74.99
20	74.73
25	78.82
30	90.74
35	77.35
40	85.41
45	82.93
50	86.36

C. Comparison Results

Table I reports the MPJPE without rigid alignment for each action on the Human3.6M dataset. The action "SitD" (sitting down) exhibits the highest error, primarily due to its complexity and the significant changes in body posture involved in the motion. During the sitting process, various joints, such as the hips, knees, and ankles, undergo rapid movements, which can lead to misinterpretations of joint positions. Moreover, self-occlusion is a major issue; as the legs can obscure the torso and other body parts, the model struggles to accurately predict the positions of hidden joints. Since 2D RGB images provide only planar information and lack depth cues, this self-occlusion complicates the estimation. Additionally, the model may not fully capture the relationships between joints during the dynamic action. For example, if the torso is obscured, the model might fail to correctly estimate the angles of the knees in relation to the hips.

To improve performance, future approaches could incorporate multi-view perspectives to provide more context or utilize depth information from sensors to enhance spatial awareness. These strategies could help mitigate the challenges posed by occlusions and improve the accuracy of joint

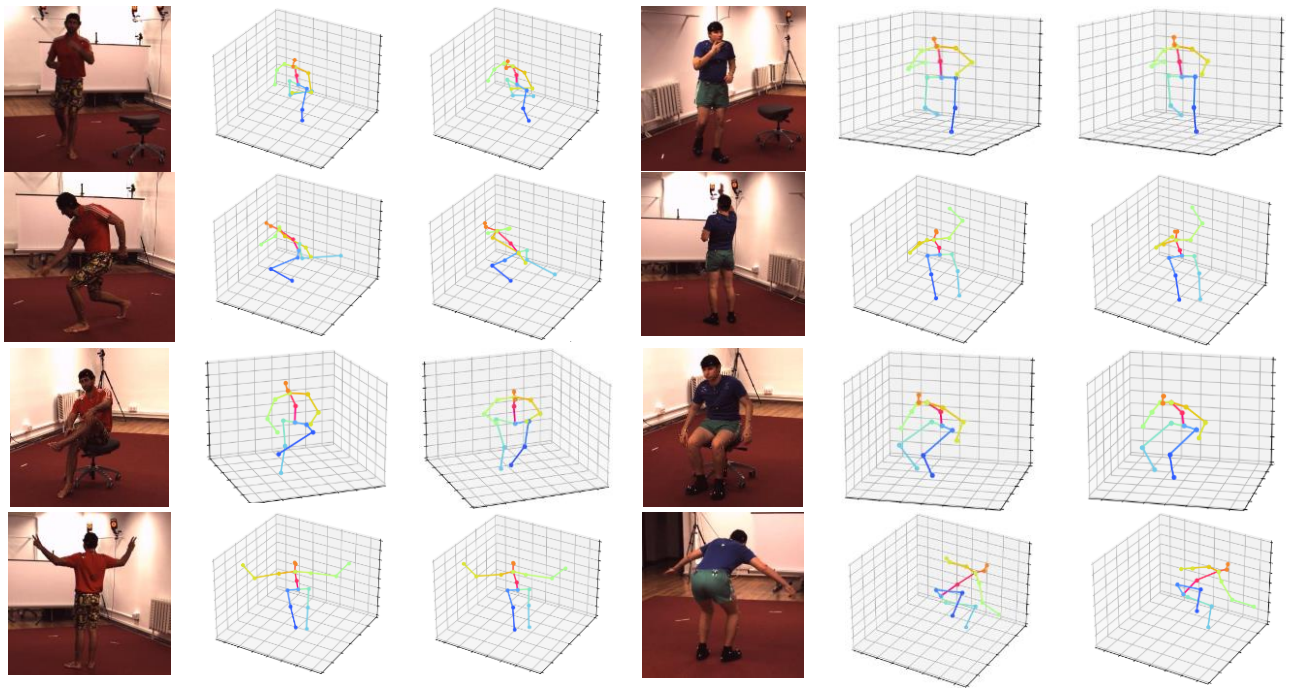


Fig. 5. The visualization results for Human3.6M dataset. The left: RGB images of neural network inputs. The middle: The 3D skeleton of ground truth. The right: The predicted 3D skeleton from out model.

estimation in actions like "SitD." As for actions "Walk", "WD" and "WT", we conjecture that the length feature of the body is effective for these actions. In Table I, we also compare the average of MPJPE and the average of P-MPJPE values with other methods. In the results, it can be seen that our model outperforms other models. It is beneficial that our model uses multiple losses and refinement blocks to predict the root-relative skeleton in the camera coordinate well. We also visualize the outputs of our model as shown in Fig. 3. In most cases, our model can predict the 3D skeleton well except for some unusual actions.

D. Ablation Study

Our ablation study is conducted on Human3.6M and the results are shown in Table II. All the ablation study uses $\lambda = 20$. It shows that adding volumetric heatmap loss (B) and adding refinement block (C) outperform the baseline (A). Table II shows that volumetric heatmap loss is useful for 3D HPE. Unlike most methods that regress the skeleton location in the spatial coordinate directly, we do not ignore the volumetric heatmap feature but regard it as one of the loss functions. Although there are three kinds of loss in our loss function, volumetric heatmap loss is still an important element in our method. It proves that the volumetric heatmap well represents the relationship between the skeleton joints in the space. As for the refinement block, the architecture without the refinement block can estimate the human pose, but the results are not good enough. The predicted skeleton has a larger error than the architecture with the refinement block. It proves that our refinement block can correct the human skeleton and output more accurate results. In summary, our volumetric heatmap loss and refinement block can improve the performance of 3D HPE.

Table III shows the number of parameters and operations in our neural network architecture. The additional calculation of volumetric heatmap loss does not increase the number of parameters in the overall architecture. Therefore, methods A and A+B in Table III have the same number of parameters and operations because they differ only in whether they compute volumetric heatmap loss or not, and methods A+C and A+B+C have the same number of parameters and operations. Because our refinement block has only a small calculation of the fully connected layer, the number of parameters and operations generated doesn't increase significantly, and it doesn't cause a burden to the overall neural network architecture. This shows that our refinement block is very effective for the task of estimating human pose.

To make the neural network pay attention to learning the 3D human skeleton, we set a weight on 3D skeleton loss. From Table IV, we found that the parameter, λ in (5) has a slight effect on accuracy. In general, the more important the loss term is, the larger the hyperparameter is. We made an experiment with hyperparameter settings, and the result is shown in Table IV We respectively set the magnification of 10, 15, 20, 25, 30, 35, 40, 45, and 50 times for the 3D skeleton loss term. Because the 3D skeleton position is the main target in pose estimation, we only set the hyperparameter on this loss function. As shown in Table IV, we obtain the lowest error when $\lambda = 20$. As λ is adjusted, the error also changes but the error does not scale up or down regularly. This may be because the neural network needs to learn the 2D heatmap feature and 3D volumetric feature and the skeleton location together. These three kinds of information are necessary for HPE. Finally, we get 74.73 MPJPE (mm) on average to prove that our model can predict the 3D human skeleton well and the visualization as shown in Fig. 5.

V. Conclusion

In this paper, we propose a refinement block that is composed

of the fully-connected layer and two residual blocks based on deep learning to estimate root-relative 3D human pose in camera coordinates. We utilize the loss from 2D heatmaps and volumetric heatmaps and 3D skeletons to calculate the total loss. We take these heatmap features as a part of our loss, allowing the neural network to learn more feature weights. The experiment results show that the volumetric heatmap loss and refinement block are effective for the 3D HPE task. Then we evaluate our model on the Human3.6M dataset, which is one of the biggest datasets with RGB images and a corresponding 3D skeleton. Finally, we get the 74.73 MPJPE (mm) on average.

Since 3D HPE can be applied in many fields such as action recognition, virtual reality, human-computer interaction, and sports analysis. By connecting an action recognition task, we can develop our architecture into a complete system. In the future, we will develop other tasks such as action recognition, and combine it with our 3D HPE to form a fully functional human-computer interactive system.

REFERENCES

[1] B. Ren, M. Liu, R. Ding, and H. Liu, ''A survey on 3D skeleton-based action recognition using learning method,'' 2020, arXiv:2002.05907. [Online]. Available: https://arxiv.org/abs/2002.05907

[2] Z. A. Kahar, P. S. Sulaiman, F. Khalid, and A. Azman, "Skeleton Joints Moment (SJM): A Hand Gesture Dimensionality Reduction for Central Nervous System Interaction," in IEEE Access, vol. 9, pp. 146640-146652, 2021, doi: 10.1109/ACCESS.2021.3123570.

[3] J. Hayakawa and B. Dariush, "Recognition and 3D Localization of Pedestrian Actions from Monocular Video," 2020 IEEE 23rd International Conference on Intelligent Transportation Systems (ITSC), 2020, pp. 1-7, doi: 10.1109/ITSC45102.2020.9294551.

[4] Lewis Bridgeman, Marco Volino, Jean-Yves Guillemaut, and Adrian Hilton. "Multi-person 3d pose estimation and tracking in sports," in The IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Workshops, June 2019.

[5] WU, Xuan, et al. LIDAR-based 3D human pose estimation and action recognition for medical scenes. IEEE Sensors Journal, 2024.

[6] Q. Dang, J. Yin, B. Wang, and W. Zheng, "Deep learning based 2D human pose estimation: A survey," in Tsinghua Science and Technology, vol. 24, no. 6, pp. 663-676, Dec. 2019, doi: 10.26599/TST.2018.9010100.

[7] ZHANG, Shihao, et al. Efficient pose estimation via a lightweight single-branch pose distillation network. IEEE Sensors Journal, 2023.

[8] Y. Chen, Y. Tian, and M. He, ''Monocular human pose estimation: A survey of deep learning-based methods,'' in Comput. Vis. Image Understand., vol. 192, Mar. 2020, Art. no. 102897, doi: 10.1016/j.cviu.2019.102897.

[9] HE, Haoyang, et al. Interacting multiple model-based human pose estimation using a distributed 3D camera network. IEEE Sensors Journal, 2019, 19.22: 10584-10590.

[10] Alexander Toshev, Christian Szegedy, "DeepPose: Human Pose Estimation via Deep Neural Networks," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014, pp. 1653-1660

[11] Newell, A., Yang, K., & Deng, J. (2016). "Stacked hourglass networks for human pose estimation," In ECCV (pp. 483–499).

[12] Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., & Sun, J. (2018). Cascaded pyramid network for multi-person pose estimation. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 7103-7112).

[13] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. "Realtime Multi-Person 2D Pose Estimation using Part Affinity Fields," In Computer Vision and Pattern Recognition (CVPR), 2017

[14] Geng, Z., Sun, K., Xiao, B., Zhang, Z., & Wang, J. (2021). Bottom-up human pose estimation via disentangled keypoint regression. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (pp. 14676-14686).

[15] Papandreou, G., Zhu, T., Kanazawa, N., Toshev, A., Tompson, J., Bregler, C., & Murphy, K. (2017). Towards accurate multi-person pose estimation in the wild. In Proceedings of the IEEE conference on computer vision and pattern recognition (pp. 4903-4911).

[16] Ke Sun, Bin Xiao, Dong Liu, and Jingdong Wang, "Deep high-resolution representation learning for human pose estimation," In CVPR, 2019, pp. 5693-5703

[17] Adrian Bulat and Georgios Tzimiropoulos, "Human pose estimation via convolutional part heatmap regression," In ECCV, volume 9911 of Lecture Notes in Computer Science, pages 717–732. Springer, 2016.

[18] Georgios Pavlakos, Xiaowei Zhou, Konstantinos G Derpanis, and Kostas Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3d human pose," in (CVPR, 2017, pp. 7025-7034

[19] Gyeongsik Moon, Ju Yong Chang, Kyoung Mu Lee, "Camera Distance-aware Top-down Approach for 3D Multi-person Pose Estimation from a Single RGB Image," in ICCV, 2019, pp. 10133-10142

[20] Shi, B., Xu, Y., Dai, W., Wang, B., Zhang, S., Li, C., ... & Xiong, H. (2020, October). Tiny-Hourglassnet: An efficient design for 3D human pose estimation. In 2020 IEEE International Conference on Image Processing (ICIP) (pp. 1491-1495). IEEE.

[21] Georgios Pavlakos, Xiaowei Zhou, and Kostas Daniilidis, "Ordinal Depth Supervision for 3D Human Pose Estimation," in CVPR, 2018, pp. 7307-7316.

[22] Julieta Martinez, Rayat Hossain, Javier Romero, and James J Little, "A simple yet effective baseline for 3d human pose estimation," in ICCV, 2017, pp. 2640-2649

[23] Li, S., Ke, L., Pratama, K., Tai, Y. W., Tang, C. K., & Cheng, K. T. (2020). Cascaded deep monocular 3d human pose estimation with evolutionary training data. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6173-6183).

[24] Li, Y., Li, K., Jiang, S., Zhang, Z., Huang, C., & Da Xu, R. Y. (2020, April). Geometry-driven self-supervised method for 3d human pose estimation. In Proceedings of the AAAI Conference on Artificial Intelligence (Vol. 34, No. 07, pp. 11442-11449).

[25] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in CVPR, 2018, pp. 7122-7131

[26] Loper, M., Mahmood, N., Romero, J., Pons-Moll, G., & Black, M. J. (2015). SMPL: A skinned multi-person linear model. ACM transactions on graphics (TOG), 34(6), 1-16.

[27] K. Iskakov, E. Burkov, V. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), 2019, pp. 7718-7727

[28] Remelli, E., Han, S., Honari, S., Fua, P., & Wang, R. (2020). Lightweight multi-view 3d pose estimation through camera-disentangled representation. In Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (pp. 6040-6049).

[29] Hanyue Tu, Chunyu Wang, and Wenjun Zeng. "Voxelpose: Towards multi-camera 3d human pose estimation in wild environment." In ECCV, pages 197–212. Springer, 2020.

[30] 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. Machine Intell., vol. 36, no. 7, pp. 1325–1339, 2014.

[31] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in CVPR, 2017, pp. 7035-7043

[32] B. Xiaohan Nie, P. Wei, and S.-C. Zhu, "Monocular 3D human pose estimation by predicting depth on joints," in Proceedings of the IEEE International Conference on Computer Vision (ICCV), 2017, pp. 3447-3455

[33] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H. P., ... & Theobalt, C. (2017). Vnect: Real-time 3d human pose estimation with a single rgb camera. Acm transactions on graphics (tog), 36(4), 1-14.

[34] Zhou, X., Zhu, M., Pavlakos, G., Leonardos, S., Derpanis, K. G., & Daniilidis, K. (2018). Monocap: Monocular human motion capture using a cnn coupled with a geometric prior. IEEE transactions on pattern analysis and machine intelligence, 41(4), 901-914.

[35] C. Luo, X. Chu, and A. L. Yuille, "Orinet: A fully convolutional network for 3d human pose estimation," in British Machine Vision Conference 2018, BMVC 2018, Northumbria University, Newcastle, UK, September 3-6, 2018, page 92, 2018.

[36] B. Wandt and B. Rosenhahn, "RepNet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation," in CVPR, 2019, pp. 7782–7791.

[37] Zhi Li, Xuan Wang, Fei Wang, and Peilin Jiang. "On boosting single-frame 3d human pose estimation via monocular videos," in ICCV, October 2019.

[38] Mitra, R., Gundavarapu, N. B., Sharma, A., & Jain, A. (2020). Multiview-consistent semi-supervised learning for 3d human pose estimation. In Proceedings of the ieee/cvf conference on computer vision and pattern recognition (pp. 6907-6916).

[39] Kundu, J. N., Seth, S., Jamkhandi, A., YM, P., Jampani, V., & Chakraborty, A. (2021). Non-local latent relation distillation for self-adaptive 3D human pose estimation. Advances in Neural Information Processing Systems, 34, 158-171.

Tsung-Han Tsai received the B.S., M.S., and Ph.D. degrees in electrical engineering from the National Taiwan University, Taipei, Taiwan, in 1990, 1994, and 1998 respectively. From 1999 to 2000, he was an Associate Professor of electronic engineering at Fu Jen University. He joined National Central University in 2000. Since 2008 he was a full Professor in the department of electrical engineering at National Central University. Currently he is also the director of Intelligent Chip and System Center in National Central University, and also serves as the Principal Investigator of the National Program for Intelligent Electronics. Dr. Tsai has been awarded more than 40 patents and 280 refereed papers published in international journals and conferences. Dr. Tsai received the Industrial Cooperation Award in 2003 from the Ministry of Education, Taiwan. He received the Best Paper Award from the IEEE International Conference on Innovations in Bio-inspired Computing and Applications (IBICA) in 2011, and IEEE International Conference on Innovation, Communication and Engineering (ICICE) in 2015. His research team has won many international IC related student design contest awards including ISOCC in 2015, TI DSP Asia Design Contest in 2008, and ISSCC in 2011. He has served as a Guest Editor of special issues for Journal of VLSI Signal Processing Systems. He was a General Co-Chair of IEEE International Conference on Internet of Things 2014, and a General Chair of IEEE International Conference on Consumer Electronics-Taiwan 2020 (ICCE-TW). He has been an IEEE member for over 20 years, and serves as Technical Program Committee member or Session Chair of several international conferences. His research interests include VLSI signal processing, video/audio coding algorithms, DSP architecture design, wireless communication and System-On-Chip design.

Yi-Jhen Luo received her bachelor degree from National Changhua University of Education, Taiwan in 2019, and received her Master degree from National Central University, Taiwan in 2022. Her research interests are focused on deep learning and human pose estimation.