3-D Human Pose Estimation in Traditional Martial Art Videos

Van-Hung Le

***Abstract*—Preserving, maintaining and teaching traditional martial arts are very important activities in social life. That helps preserve national culture, exercise and self-defense for practitioners. However, traditional martial arts have many different postures and activities of the body and body parts are diverse. The problem of estimating the actions of the human body still has many challenges, such as accuracy, obscurity, etc. In this paper, we survey several strong studies in the recently years for 3-D human pose estimation. Statistical tables have been compiled for years, typical results of these studies on the Human 3.6m dataset have been summarized. We also present a comparative study for 3-D human pose estimation based on the method that uses a single image. This study based on the methods that use the Convolutional Neural Network (CNN) for 2-D pose estimation, and then using 3-D pose library for mapping the 2-D results into the 3-D space. The CNNs model is trained on the benchmark datasets as COCO dataset, Human 3.6M, MPII dataset, LSP, etc. From this comparative study, we can see when there are good 2-D human pose estimation results, then there will be good 3-D human pose estimation results. Quantitative results are presented and evaluated.**

***Index Terms*—2-D key points estimation, 3-D key points estimation, 3-D human pose estimation, convolutional neural network (CNN).**

I. Introduction

Estimating and predicting the actions of the human body is a well-studied problem in the robotics and computer vision community. 3-D human pose estimation is also applied in many other applications such as sports analysis, evaluation analysis and playing games with 3-D graphics, or in health care and protection. Especially, 3-D human pose estimation has the estimated results that can fully see human actions in the real world, and addresses cases when human parts are obscured. However, 3-D human pose estimation have many challenges. The estimation in the 3-D space is very difficult to extract and train the features vector because 3-D data is much more complex than data in 2-D space (image space), or estimate many people in the outdoor environment, noise of data (data missing parts of the human body). There are two methods to do recovering 3-D human pose: The first is recovering 3-D human pose from a single image; The second is recovering 3-D human pose from a sequence of images [1]. Regarding the first method 3-D human pose estimation using a single image usually performs 2-D human pose estimation and then maps to 3-D space. The second method using a sequence of images is the combination of its 2-D pose human estimation and based on geometric transformations (affine transformations)/mapping to build the skeleton in the 3D space of the person [2].

To address 2-D human pose estimation can be based on a set of methods such as analyzing people in the images, locating people in the images, locating key points on human bodies and identifying joints on points represented on the body (skeleton). In recent years, studies of these methods are often based on the CNN models. 2-D human pose estimation is usually based on color images and depth images or it is based on objects and action context [3]. The above studies often use color images, depth images [4], or skeleton [5] obtained from different types of sensors (e.g, Microsoft (MS) Kinect version 1, MS Kinect version 2, Time-of-Flight-Sensors).

In this paper, we survey on recent 3-D human pose estimation techniques in the recently years. We also propose a comparative study for 3-D human pose estimation based on the method that uses a single image. We utilized the CNNs CPM (Convolutional Pose Machines) [6] and ResNet50 [7] for estimating 2-D human pose. The methods in this study are evaluated on the MADS (Martial Arts, Dancing and Sports) dataset [8].

II. Related Works

3-D human pose estimation is often using most computer vision techniques. These studies can be based on a single image or a sequence of images. The human poses and actions estimation is applied in many application such as: human interaction (such as body language or gesture recognition), human interaction with robots, video surveillance (use to convey human actions) [1]. To address 3-D human pose estimation from a single image, these studies are often performed from 2-D pose estimation and then mapping into the 3-D space. The model often applied to estimating 3-D human pose is shown in Fig. 3 of [1]. In this section, we examine in detail the studies that estimate 3-D human pose following two above methods. Especially in the last few years a number of studies on 3-D human pose estimation have been published on many prestigious conferences and journals of computer science and computer vision. This is shown in Fig. 1.

Most studies of 3-D human pose estimation use the CNN models to train and estimate 2-D human pose (first method)(studies by Pavllo *et al.* [9], Wang *et al.* [10], etc) or use the 2-D human pose annotation (second method) (studies by Karim *et al.* [11], Hossain *et al.* [12], etc). These studies use color or depth images as input. The first method projected the 2-D human pose results into the 3-D space by

Manuscript received December 7, 2019; revised February 3, 2020.
The author is with Tan Trao university, Vietnam (e-mail: Van-hung.le@mica.edu.vn).

doi: 10.18178/ijmlc.2020.10.2.943

3-D pose library as [13] and then find the most suitable 3-D pose; The second method projected the 3-D space by the parameters of captured sensors [14] or using a CNN model [15].

In particular, most studies of 3-D human pose estimation are evaluated on the Human3.6m dataset [13] with the following common measurements: MPJPE (Mean Per Joint Position Error) [9], PCK (Percentage of Correct Keypoints), and AUC (Area Under Curve) [16], PMPJPE (Procrustes Aligned Mean Per Joint Position Error) [14], etc. These studies are often evaluated on datasets such as: Human3.6m [13], LSP [17], 3DHP [18], MPII [19], HumanEva-I [20], Football [21], Invariant-Top View [22], [23], MPI-INF3DHP [18], MuPoTS-3D [24], AIChallenger [10].

3-D human pose estimation result was based on MPJPE measurement, as shown in Table I.

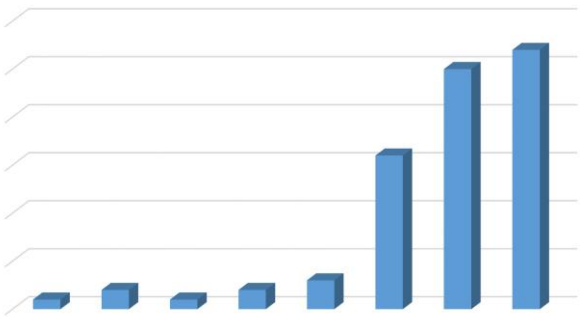

Fig. 1. Statistics of published studies on the 3-D human pose estimation following each year.

A. 3-D Human Pose Estimation from a Single Image

As reported in the survey of Sarafianos *et al.* [1], 3-D human pose estimation from a single image is performed based on two steps: 2-D human pose estimation and then estimate its depth by matching to a library of 3-D poses as Fig. 2.

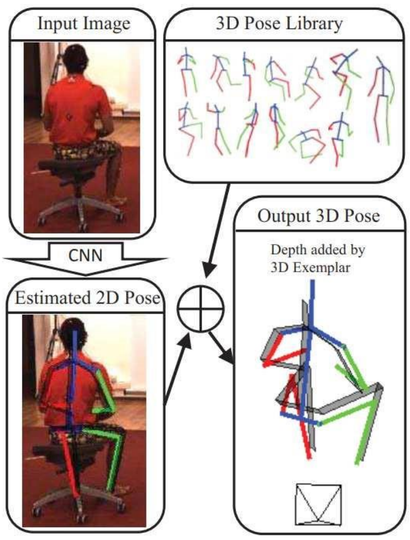

Fig. 2. Illustration of method for 3-D human pose estimation [36]: the input is a RGB image, the first estimate a 2-D pose and then estimate its depth by matching to a library of 3-D poses. The final prediction is given by the colored skeleton based on the 3-D poses library, while the ground-truth is shown in gray.

B. 3-D human Pose Estimation from a Sequence of Images

Especially estimating 3-D skeleton and posture of human is an essential skill in rebuilding the actual environment and estimating joints in the field of the parts of the human limbs is obscured.

III. 3-D Pose Estimation

The activity of the human body is detected and recognized as well as predicted and estimated, based on parts of the human body. Parts are based on the link between the key points. Each part is represented by a L_c vector in 2-D space (image space) in a set of vectors on human body S, where the set of vectors $L = \{L_1, L_2, ..., L_C\}$, has **C** vectors on human body **S**. The body of S is represented by the key points J, $S = \{S_1, S_2, ..., S_J\}$. For an input image of size ($w \times h$) pixels, the position of the key points can be $S_j \in R^{w \times h}, j \in \{1, 2, ..., J\}$. CNN architecture is shown in Fig. 5. As can be seen in Fig. 5, this CNN consists of two branches performing two different jobs. From input data, a set of feature maps F is created from analyzing image, then these confidence maps and affinity fields are detected at the first stage. The key points on the training data are displayed on confidence maps as shown. These points are trained to estimate key points on color images. The first branch (top branch) is used to estimate key points; the second branch (bottom branch) is used to predict the affinity fields matching joints on many people. As shown in Fig. 5, this CNN consists of two branches performing two different jobs. From the input data, a set of feature maps F is created from the image analysis; these confidence maps and affinity fields are detected at the first stage. Branch in Fig. 5 is the CNN that called "CPM - Convolutional Pose Machines" [6] to estimate 2-D human pose.

TABLE I: Statistics of the Result of Studies Based on the MPJPE(mm) Measurement on the Human 3.6m Dataset [13] for 3-D Human Pose Estimation

Method	Results of Mean Per Joint Position Error (MPJPE) (mm)
Pavllo *et al.* [9]	Protocol 1: 51.8 Protocol 2: 40.0
Nibali *et al.* [16]	57.0
Veges *et al.* [25]	Protocol #1: 61.1
Wang *et al*. [26]	Protocol #1: 63.67
Martinez *et al*. [27]	protocol #1: 45.5
Pavlakos *et al.* [28]	51.9
Wang *et al.* [10]	Protocol#1: 40.8
Hossain *et al.* [12]	Protocol #1: 39.2
Li *et al*. [29]	Protocol #1: 52.7 Protocol #2: 42.6
Karim *et al.* [11]	Protocol 1: 49.9
Fang *et al.* [30]	Protocol #1: 60.4 Protocol #2: 45.7 Protocol #3: 72.8
Tekin *et al*. [31]	50.12
Omran *et al.* [32]	59.9
Pavllo *et al.* [33]	36
Bastian *et al.* [15]	Protocol #1: 50.9
Kocabas *et al*. [14]	51.83
Rhodin *et al.* [2]	131.7
Mehta *et al.* [34]	ResNet 100: 82.5 ResNet 50: 80.5
Tome *et al.* [35]	Protocol #1: 88.39 Protocol #2: 70.4 Protocol #3: 79.6

The detailed model of training and predicting (Fig. 3) of Zhe's study [40] is shown as follows: The input image at stage 1 is an image with 3 color channels (R,G,B) and has a size of $h \times w$ and features extracted from multiplication with masks that have the size $9\times9, 2\times, 5\times,...$ for training set X as shown in the Fig. 4. The number of convoluational layers of CPM is 5, shown in Fig 5. For each mask, there will be a patch and training model g_1, g_2 at each stage, which will predict the heatmaps such as b_1, b_2 at each stage as shown in Fig. 3. As shown in the Fig. 3, 4, Convolutional Pose Machines consist of at least 2 stages and the number of phases is a super parameter (usually 3 stages). The second stage takes the results of the heatmaps of the first stage as the input.

Therein, each heatmap indicates the location confidence (x,y) of the key points. Therefore, the key points on the training data are displayed on confidence maps as shown in Fig. 3. These points are trained to estimate the key points on color images. The first branch (top branch) is used to estimate the key points, and the second branch (bottom branch) is used to predict the affinity fields matching joints.

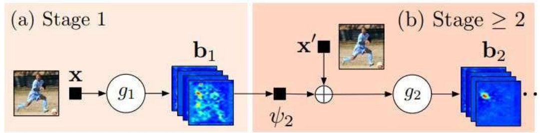

Fig. 3. Illustration of the detail model to predict the heatmaps [41].

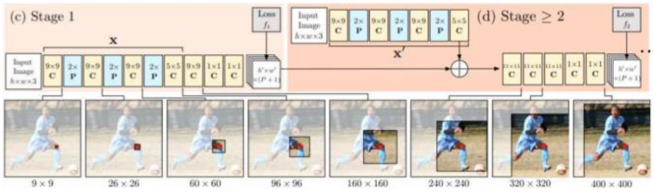

Fig. 4. Illustration of the detail model to extract the feature for training model and to predict the heatmaps at each stage [41].

TABLE II: SURVEY: 3-D HUMAN POSE ESTIMATION FROM A SINGLE IMAGE

Year	Main Author/ reference	3- D pose library	Method Highlights	Evaluation dataset	Evaluation matrix
2019	Pavllo *et al.* [9]	Yes	2D human pose estimation use Mask R-CNN with a ResNet-101-FPN, using its reference implementation in Detectron, as well as cascaded pyramid network (CPN) (trained models on COCO); 3D human pose estimation: As optimizer authors use Amsgrad and train for 80 epochs in Human3.6m dataset	Human3.6m HumanEva-I	MPJPE
2019	Nibali *et al.* [16]	No	In 2D human pose estimation, coordinates predicted by the model are in the same xy coordinate space as the input, making it straightforward to construct a simple fully convolutional network which maps RGB inputs to xy heatmaps. 3D coordinate prediction which avoid the aforementioned undesirable traits by predicting 2D marginal heatmaps under an augmented soft-argmax scheme.	MPII Human3.6m	PCK MPJPE AUC
2019	Wang *et al.* [26]	Yes	2D pose sub-network by borrowing the architecture of the convolutional pose machines. From 2D pose sub-network, the 3D pose transformer module is employed to adapt the 2D pose-aware features in an adapted feature space for the later 3D pose prediction.	Human3.6m HumanEva-I	MPJPE
2019	Veges *et al.* [37]	No	The 2D pose detector is the state-of-the-art multi-person pose detector OpenPose on the depth image; the 2D-to-3D component is called 3D PoseNet.	MuPoTS-3D	MPJPE
2019	Wang *et al.* [10]	Yes	The significant advances have been achieved in 2D human pose estimation due to the powerful deep Convolutional Neural Networks (CNNs) and the availability of large-scale in-the-wild 2D human pose datasets with manual annotations. The authors propose a novel stereo inspired neural network to generate high quality 3D pose labels for in-the-wild images.	MPII LSP AIChallenger Human3.6m	MPJPE
2019	Li *et al.* [29]	No	The authors adopt the state-of-the-art stacked hour glass network as the 2D joint estimation; Propose a novel approach to generate multiple feasible hypotheses of the 3D pose from 2D joints	Human3.6M MPII MPI-INF 3 DHP	MPJPE
2018	Veges *et al.* [25]	Yes	2D pose is determined with an off-the-shelf component and then the 3D position is predicted from the 2D skeleton. 3D pose estiamtion: using the Adam optimizer with a learning rate of 0.001 and an exponential decay with a rate of 0.96. The batch size was set to 256. The training ran for 100 epochs.	Human3.6m	MPJPE
2018	Sun *et al.* [28]	Yes	First, a person box detection component roughly localizes the person in the input RGB image. Second, a camera projection component is used to project 3D ground truth to the image coordinate system, as done in per-pixel/voxel classification based learning methods.	COCO MPII	MPJPE
2018	Fang *et al.* [30]	Yes	For 2D pose estimation, existing large-scale pose estimation datasets (Andriluka *et al.* 2014; Charles *et a*l. 2016); Authors develop a deep grammar network that incorporates both powerful encoding capabilities of deep neural networks and high-level dependencies and relations of human body	Human3.6m HumanEva-I MPII	MPJPE
2018	Omran et *al.* [32]	No	The authors propose a novel approach (Neural Body Fitting (NBF)). It integrates a statistical body model within a CNN, leveraging reliable bottom-up semantic body part segmentation and robust top-down body model constraints.	UP-3D HumanEva-I Human3.6m	MPJPE
2018	Pavllo *et al.* [33]	No	QuaterNet, represents rotations with quaternions and our loss function performs forward kinematics on a skeleton to penalize absolute position errors instead of angle errors; it reduce proning to error accumulation along the kinematic chain	Human3.6m	MPJPE
2017	Martinez *et al.* [27]	Yes	2D pose detections using the state-of-the-art stacked hourglass network which pre-trained on the MPII dataset; we can train data-hungry algorithms for the 2d-to-3d problem with large amounts of 3D mocap data captured in controlled environments	Human3.6m HumanEva MPII	MPJPE
2017	Pavlakos *et al.* [28]	Yes	For 2D human pose estimation, authors discretize the space around the subject and use a ConvNet to predict per voxel likelihoods for each joint from a single color image; a subsequent optimization step to recover 3D pose.	Human3.6m HumanEva-I KTH Football II MPII	MPJPE
2017	Tekin *et al.* [31]	No	For 2D human pose estimation: The authors employed the stacked hourglass network design, which carries out repeated bottom-up, top-down processing to capture spatial relationships in the image; a discriminative fusion framework to simultaneously exploit 2D joint location confidence maps and 3D image cues for 3D human pose estimation.	Human3.6m HumanEva-I KTH Football II LSP	MPJPE
2016	Haque *et al.* [39]	No	The authors propose a viewpoint invariant model for 3D human pose estimation from a single depth image. To achieve this, our discriminative model embeds local regions into a learned viewpoint invariant feature space	Stanford EVAL Invariant-Top View	PCKh

TABLE III: SURVEY: 3-D HUMAN POSE ESTIMATION FROM A SEQUENCE OF IMAGES

Using 3-D pose library	Method Highlights	Evaluation dataset	Evaluation matrix
No	The authors present two novel solutions for multi-view 3D human pose estimation based on new learnable triangulation methods that combine 3D information from multiple 2D views.	Human3.6m CMU Panoptic	MPJPE
Yes	One part of the proposed reprojection network (RepNet) learns a mapping from a distribution of 2 D poses to a distribution of 3D poses using an adversarial training approach.	Human3.6m MPI-INF-3DHP LSP	MPJPE
Yes	EpipolarPose estimates 2D poses from multi-view images, and then, utilizes epipolar geometry to obtain a 3D pose and camera geometry which are subsequently used to train a 3D pose estimator.	Human3.6m MPI-INF-3DHP	MPJPE PMPJPE PCK PSS @50 PSS@100
Yes	The authors propose to overcome this problem by learning a geometry-aware body representation from multi-view images without annotations.	Human3.6m	MPJPE N-MPJPE P-MPJPE
Yes	The authors take a sequence of 2-D poses and encodes them in a fixed size high dimensional vector in the hidden state of its final LSTM unit; utilize the temporal information across a sequence of 2-D joint locations to estimate a sequence of 3-D poses	Human3.6m	MPJPE

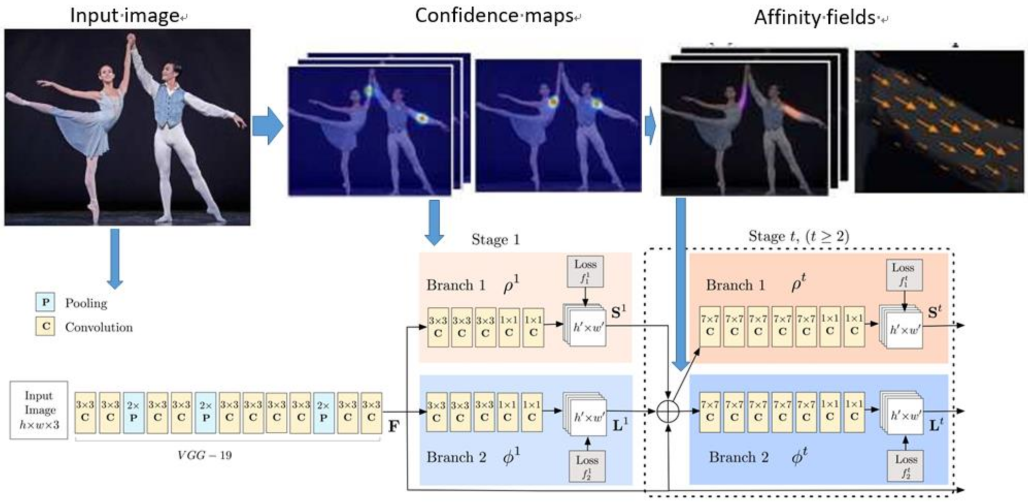

Fig. 5. The architecture of the two-branch multi-stage CNN for training the model estimation [40].

To visualize the results step by step during 3-D human pose estimation process, we propose a comparative study on 2-D human pose estimation (**2-D Comparative Study**), it is shown in Fig. 6.

In Fig. 6, we evaluate on two methods: The first method (**Method 1**) is using the trained model that uses CPM network on COCO dataset [42]; The second method (**Method 2**) is using the trained model that uses CPM network on Human 3.6M dataset [13]. The result of 2-D human pose estimation with coordinates of each estimated key point (x_i,y_i) on the color image, then they are combined with the pixel that has the coordinates (x_i,y_i) on the depth image according to the equation (1) to generate a key point (X_p,Y_p,Z_p) in the 3-D space.

$$\begin{aligned} X_p &= \frac{(x_i - c_x)*\text{depthvalue}(x_i,y_i)}{f_x} \\ Y_p &= \frac{(y_i - c_y)*\text{depthvalue}(x_i,y_i)}{f_y} \\ Z_p &= \text{depthvalue}(x_i,y_i) \\ C(r,g,b) &= \text{colorvalue}(x_i,y_i) \end{aligned} \quad (1)$$

where depthvalue(x_i,y_i) is the depth value of a pixel (x_i,y_i) on the depth image, colorvalue(r,g,b) is the color value of a pixel (x_i,y_i) on the color image, (c_x,c_y) are the principle point (usually the image center), f_x, f_y are the focal lengths. Next, we conduct a comparative study of 3-D human pose estimation **(3-D Comparative Study)**, as is shown in Fig. 7. In which the methods are presented as follows: The first method is called **"3-D_COCO_Method"**: 2-D human pose estimation by using CPM that was trained on the COCO [42] dataset + mapping to 3-D space by 3-D pose library of Human 3.6m dataset base on the method of Tome *et al.* [35]. The second method is called **"3-D_HUMAN3.6_Method"**: 2-D human pose estimation by using CPM that was trained on the Human 3.6m [13] + mapping to 3-D space by 3-D pose library of Human 3.6m dataset base on the method of Tome *et al.* [35]. The third method is called **"3-D_VNECT_Method"**: 2-D, 3-D human pose estimation using the VNect in study of Mehta *et al.* [34].

The method of Tome *et al.* [35] implemented the process of 3-D human pose estimation based on mapping the 2D human pose estimation results into the 3-D space. This process is of finding a 3-D human pose model with an optimal rotation, the approximate model found based on a Gaussian distribution (the smallest error function). The optimization is to optimize a set of variables, from a set of N 3-D human pose, each representation is a matrix $P_i(3\times L)$ 3-D joints, where $i \in 1,2,...,N$ and L is the number of joints in 3-D space.

This method finds global estimates of an average 3-D pose μ, a set of J orthonormal basis matrices $\boldsymbol{e}$ and noise variance $\boldsymbol{\sigma}$, along with each per sample rotations R_i and basis coefficients $\boldsymbol{a_i}$ to minimize the following estimate as Eq. 2.

$$\underset{R,\mu,a,e,\sigma}{\operatorname{argmin}} \sum_{i=1}^{N} (|| P_i - R_i(\mu + a_i e) ||_2^2 + \sum_{j=1}^{J} (a_{i,j\sigma_j})^2 + ln \sum_{j=1}^{J} \sigma_j^2) \quad (2)$$

where, $a_i e = \sum_j a_{i,j} e_j$ is the tensor analog of a multiplication between a vector and a matrix, and $|| \, . \, ||_2^2$ is the squared Frobenius norm of the matrix, y axis is assumed to point up and the rotation matrix R_i is considered to be rotated against the ground plane.

In the comparative study, the third method is based on the method of Mehta *et al.* [34], The authors use the regression CNN model to predict the heatmaps by method of Tompson *et al.* [43] Especially the training of features for learning and predicting the map highlights is based on ResNet (Deep Residual Networks) network [44], which provides a breakthrough idea for building Characteristic and training. The ResNet in [44] is built on the platform of Tensorflow library of [45].

The model in this network uses the MPII dataset [19], LSP [17] for the training of estimating the key points on the image. To estimate the 3-D human pose, the authors employed the method of Ionescu *et al.* [47] with the use of Human3.6m dataset [13] and MPI-INF-3DHP [48] for projecting 2-D human pose estimation to the 3-D space.

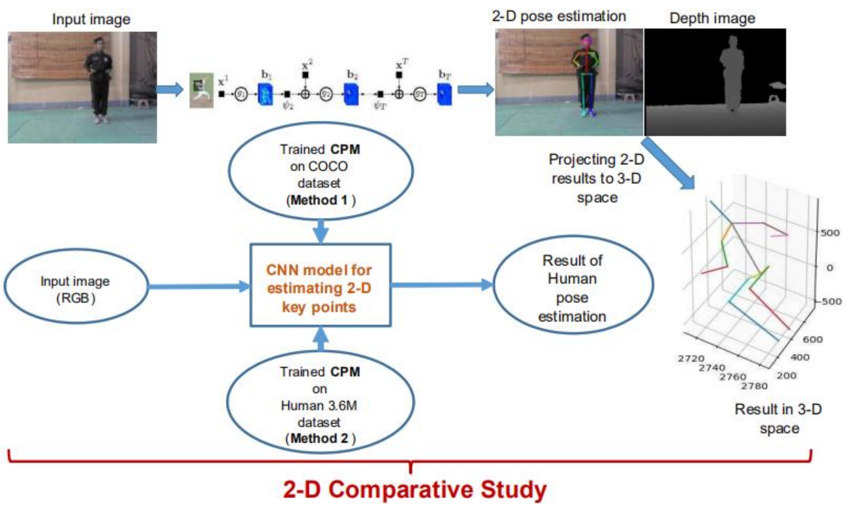

Fig. 6. Comparative study for evaluating 2-D human pose.

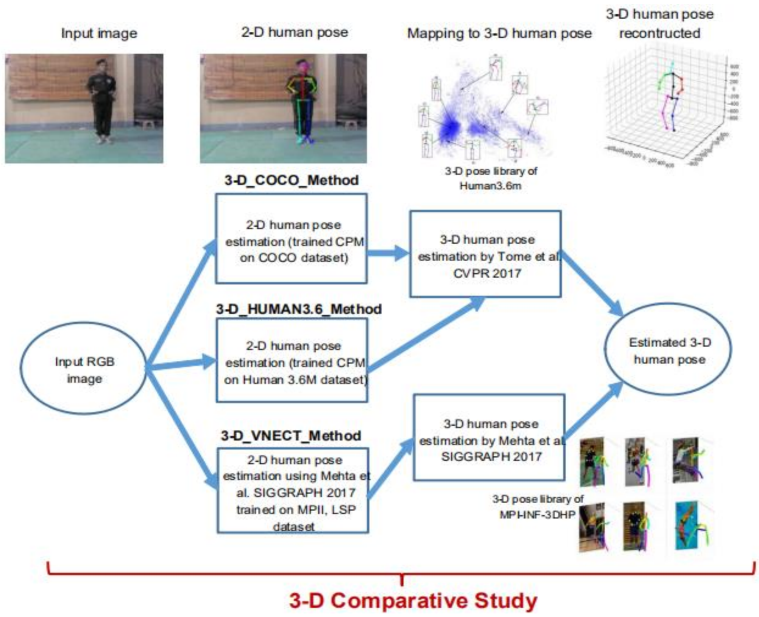

Fig. 7. Comparative study for evaluating 3-D human pose estimation.

IV. Experimental Results

A. Data Collection and Evaluation

Recently, Zhang *et al.* [8] published the benchmark dataset that called "MADS - Martial Arts, Dancing and Sports", which consists of both multi-view RGB videos and depth videos. This dataset contains 5 challenging actions types: Tai-chi, Karate, Hip-hop dance, Jazz dance and sports, with the total of approximately 53,000 frames. The frame rate is used to capture the video (10 fps for Tai-chi and Karate and 20 fps for jazz, hip-hop and sports). The ground truth pose data is prepared in the 3-D pose, using a MOCAP (MOtion CAPture) system [20] by Motion Analysis. Seven MOCAP cameras are placed on the walls around the capture space to record the positions of markers on the human body.

The MOCAP system works at frame rate of 60 fps. The 3-D pose includes 19 key points, sorted and labeled as follows: neck, pelvis, left hip, left knee, ankle left, right hip, right knee, right ankle, left shoulder, left elbow, left wrist, left hand, right shoulder, right elbow, right wrist, right hand, head.

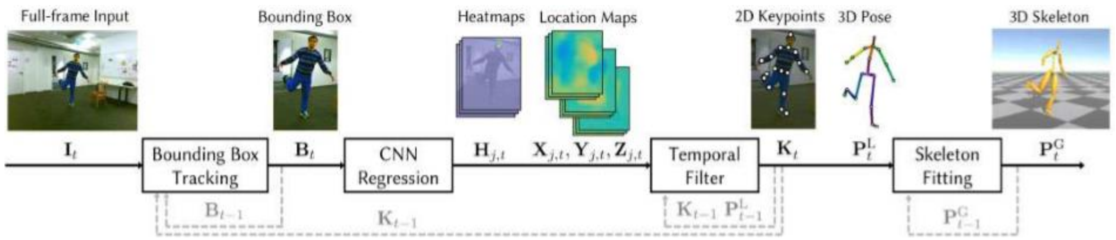

Fig. 8. The architecture of VNect network [34].

In this paper, we use a trained model on the COCO dataset [42] for 2-D human pose estimation of the first method "**3-D_COCO_Method".**

We also use a trained model on the Human 3.6m dataset [13] for 2-D human pose estimation of the second method "**3-D_HUMAN3.6_Method**".

The models trained based on the published OpenPose [49]. The parameters of training the whole CNN network are as follows: the size of the input image is (width: 368× height: 368 × channel: 3), *batchSize* = 16, *stacks* = 4, the number of stages is 6 for pooling, the number of convolutional layers is 5, etc. The detail of the parameters is shown in the link: https://github.com/ZheC/Realtime Multi-Person Pose Estimation/blob/master/training/ example proto/pose train test.prototxt. The parameters of training the whole CPM are shown in the link: https://github.com/DenisTome/Lifting-from-the-Deep-release/ blob/master/packages/lifting/utils/cpm.py.

The parameters of mapping 2-D human pose estimation result to the 3-D space of "**3-D COCO Method**" and "**3-D_HUMAN3.6_Method**" methods are shown in the link: https://github.com/DenisTome/ Lifting-from-the-Deep-release/blob/master/packages/lifting/ utils/prob model.py. The parameters of the third method "**3-D_VNECT_Method**" are shown in the link: https://github. com/XinArkh/VNect/blob/master/src/vnect model.py.

The output of 2-D human pose estimation based on the Method 2 in Fig. 6 is 14 key points, therefore when evaluating the 2-D human pose estimation results, we only evaluate on 14 key points. The output of 3-D human pose estimation based on the method of Tome *et al.* [35] (the methods: "**3-D_COCO_Method**", "**3-D_HUMAN3.6_Method**") are 17 key points, as shown in Fig. 9. The output of 3-D human pose estimation based on the method of Mehta *et al.* [34] ("**3-D_VNECT_Method**") is 21 key points. We calculate the assignment of the output data of the above three methods and the 3-D ground truth data, only 15 key points are considered. Therefore we evaluate 3-D human pose estimation results on 15 key points. However, the 3-D ground truth data follows the MOCAP system [8] and the estimated data is based on the coordinate system of the training data to estimate the 3-D human pose as the coordinate system of Human 3.6M [13] and MPI-INF-3DHP [48] datasets are different. Therefore take steps to the synchronized the coordinate system.

In this study we combine the findings of the rotation and the translation matrix into a process, in which the rotation and translation matrices are represented in the 3-D space [50] as Eq. 3

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3)$$

where $P(x,y,z)$ is the estimated point of 3-D human pose estimation result; $P'(x',y',z')$ is the estimated point of 3-D human pose estimation result after transform to the same coordinate system with the 3-D ground truth data. Therefore, we have a formulation in Eq. (4).

$$\left\{ \begin{array}{lcl} x' & = & R_{11}x + R_{12}y + R_{13}z + T_1 \\ y' & = & R_{21}x + R_{22}y + R_{23}z + T_2 \\ z' & = & R_{31}x + R_{32}y + R_{33}z + T_3 \end{array} \right\} \quad (4)$$

From the coordinates of the estimated key points in the 3-D human pose, we define the coordinates of a estimated 3-D pose including n points as in Eq. (5).

$$\begin{bmatrix} 1 & z_1 & y_1 & x_1 \\ 1 & z2 & y_2 & x_2 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 1 & z_n & y_n & x_n \end{bmatrix} \quad (5)$$

In particular, the rotation matrix and translation according to the *x,y,z* axes are presented in the order $\theta_1,\theta_2,\theta_2$ as in the Eq. (6).

$$\theta_1 = \begin{bmatrix} T_1 \\ R_{13} \\ R_{12} \\ R_{11} \end{bmatrix} \theta_2 = \begin{bmatrix} T_2 \\ R_{23} \\ R_{22} \\ R_{21} \end{bmatrix} \theta_3 = \begin{bmatrix} T_3 \\ R_{33} \\ R_{32} \\ R_{31} \end{bmatrix} \quad (6)$$

The results of rotation and translation are shown in the vector X',Y',Z' as in the Eq. (7).

$$X' = \begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ . \\ x'_n \end{bmatrix} Y' = \begin{bmatrix} y'_1 \\ y'_2 \\ . \\ . \\ . \\ y'_n \end{bmatrix} Z' = \begin{bmatrix} z'_1 \\ z'_2 \\ . \\ . \\ . \\ z'_n \end{bmatrix} \quad (7)$$

where, x_i,y_i,z_i is the coordinate value on the 3-D pose ground truth data (which is the coordinate system destination that the 3-D human pose estimated to be rotated and translated to it);

x_j, y_j, z_j is the coordinates of key points of the 3-D human pose estimated data, which is expected to rotate and translate to the same coordinate system with the 3-D human pose ground truth data.

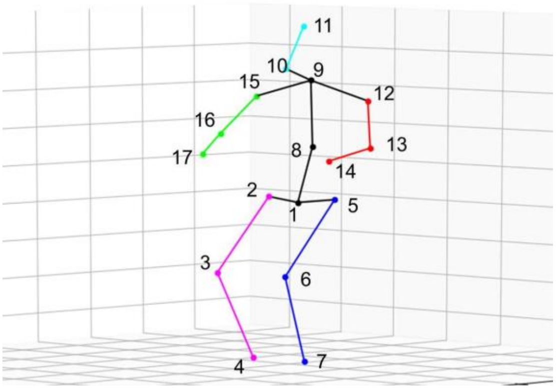

Fig. 9. The output of 3-D human pose estimation based on the method of Tome *et al*. [35].

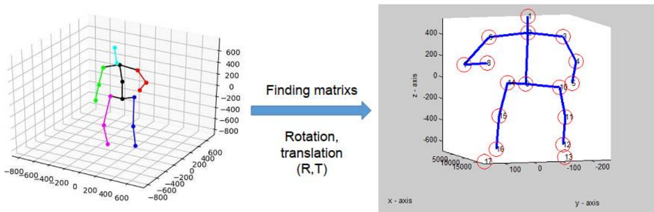

Fig. 10. Illustration of finding the rotation, translation matrix in the 3-D space.

From this, we have a system of linear equations presented in the Eq. (8).

$$\begin{aligned} X' &= M\theta_1 \\ Y' &= M\theta_2 \\ Z' &= M\theta_3 \end{aligned} \quad (8)$$

In which the estimation $\boldsymbol{\theta_i}$ is the using the Least Squares method (LS) [51], [52] as in Eq. (9).

$$\begin{aligned} \theta_1 &= (M^T M)^{-1} M^T X' \\ \theta_2 &= (M^T M)^{-1} M^T Y' \\ \theta_3 &= (M^T M)^{-1} M^T Z' \end{aligned} \quad (9)$$

The entire source of the rotation and translation is stored in the path: https://drive.google.com/file/d/1dIHgal63TcGn0-6hnTJsEDfh8qkNOsE/view?usp=sharing and explained in detail in the "Readme.md" file in this link. Finally we have the transformation matrix in the form ($\theta_1, \theta_2, \theta_3$). The testing process is performed on workstation computer with Intel (R) Xeon (R) CPU E5-2420 v2 @ 2.20GHz 16GB RAM, GPU GTX 1080 TI-12GB Memory. In this paper, we choose 15 common points between the 3-D ground truth data, the output key points of Tome *et al.* [35] method and the output key points of Mehta *et al.* [34] method, shown in Fig. 11.

We use the MPJPE (Mean Per Joint Position Error) (mm) for evaluating 3-D human pose estimation. This measure is the Euclidean distance between the two key points corresponding to the 3-D ground truth data and the estimated 3-D pose, the distance is calculated as in Eq. 10.

$$D(p_g, p_e) = \sqrt{(x_g - x_e)^2 + (y_g - y_e)^2 + (z_g - z_e)^2} \quad (10)$$

where (x_g, y_g, z_g) is the coordinates of the ground-truth key points p_g in the 3-D space, (x_e, y_e, z_e) is the coordinates of the estimated key points p_e in the 3-D space.

The input data of this study is the color images in the video. The output data is the 3-D human pose estimation results.

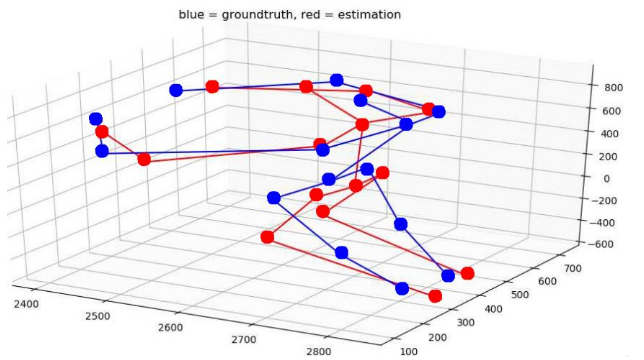

Fig. 11. Illustrating 3-D human pose for evaluating 3-D human pose estimation. The blue key points are ground truth data, the red key points are the estimated data which transformed the same coordinate system.

B. Results of Estimation and Discussion

We first evaluated the results of 2-D human pose estimation (**2-D Comparative Study**) on the 3-D space with the MADS dataset. This dataset published 3-D ground truth pose data [8]. The estimated results are shown in Table IV, and the number of frames used for evaluating, is shown in Table V.

TABLE IV: THE RESULT OF 2-D HUMAN POSE ESTIMATION THEN PROJECTED TO THE 3-D SPACE ON MADS DATASET WITH 14 KEY POINTS

#Video	MPJPE (mm)	
	Method 1	Method 2
Kata_F2	167.0256	170.9718
Kata_F3	92.8588	122.0557
Kata_F4	169.6934	169.5459
Kata_N2	90.6843	118.5762
Kata_N3	131.483	166.6152
Kata_P3	136.4613	151.514
Tai_chi_S1	121.4755	145.6657
Tai_chi_S2	107.303	141.7948
Tai_chi_S3	140.8937	177.942
Tai_chi_S4	137.6644	163.3607
Tai_chi_S5	147.1612	160.3719
Tai_chi_S6	124.4179	156.7291
Average	**130.5935083**	153.7619

Table IV and Fig. 12, CPM training on the COCO dataset (the average of MPJPE is 130.59 mm) is better than CPM when training on the Human 3.6m dataset (the average of MPJPE is 153.76 mm).

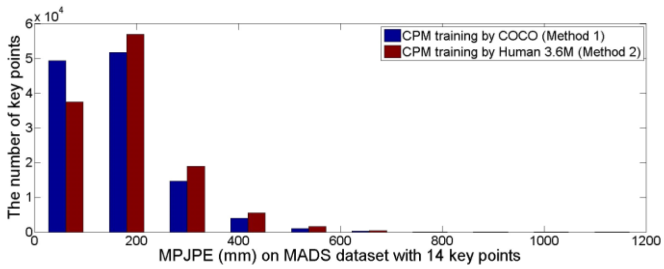

Fig. 12. Distribution of error distance MPJPE of the pair of key points between the ground truth data and the estimated data on the MADS dataset.

Table V shows the number of frames that used for evaluating 2-D human pose estimation with **Method 1** and **Method 2**, they are lower than the number of frames on the ground truth data, because when projecting the results of 2-D human pose estimation to the 3-D space is missing the depth data. As in Fig. 13 (left), the estimated key point (1) is outside the data of head, which on the depth image has only the data of human, and the other areas have the depth value equal to 0, as shown in Fig. 13(right). Although we have calculated the average depth of area that has the size of 10×10 pixels, there are still many frames that have lower 14 key points in the 3-D space. We do not evaluate on these frames.

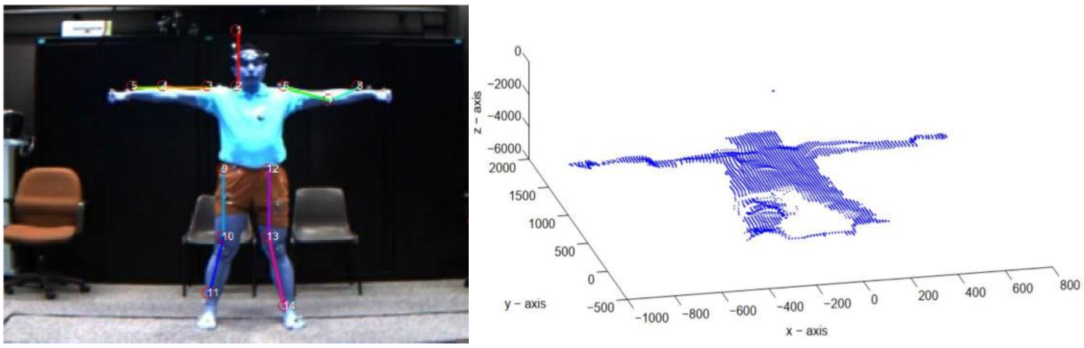

Fig. 13. Distribution of error distance MPJPE of the pair of key points between the ground truth data and the estimated data on the MADS dataset

TABLE V: THE NUMBER OF FRAMES FOR EVALUATING 3-D HUMAN POSE ESTIMATION RESULTS AT TABLE IV

#Video	The number of frames for evaluating		The number of frames on ground truth data
	Method 1	Method 2	
Kata_F2	1186	1207	1300
Kata_F3	874	812	1400
Kata_F4	1106	1106	1400
Kata_N2	875	872	1400
Kata_N3	1299	1148	1400
Kata_P3	961	822	1400
Taichi_S1	494	493	500
Taichi_S2	462	461	500
Taichi_S3	369	321	400
Taichi_S4	484	485	500
Taichi_S5	424	425	500
Taichi_S6	488	478	500
Sum	**9022**	8630	11200

We second evaluated the results of 3-D human pose estimation (**3-D Comparative Study**). The results of 3- D human pose estimation on MADS dataset are shown in Table VI.

TABLE VI: THE AVERAGE DEVIATION OF THE ESTIMATED KEY POINTS AND THE KEY POINTS OF THE GROUND TRUTH DATA ON THE MADS DATASET (15 KEY POINTS IN THE 3-D SPACE) (MM)

#Video	MPJPE (mm)		
	3-D_COCO_ Method	3-D_HUMAN3.6_ Method	3-D_VNECT_ Method
Kata_F2	102.0685	147.1236	168.0953
Kata_F3	78.0681	102.4019	122.2993
Kata_F4	105.8182	133.6986	152.3534
Kata_N2	79.0682	113.4793	165.0814
Kata_N3	34.7923	135.7989	168.1528
Kata_P3	101.3404	113.9912	129.7044
Tai_chi_S1	80.0703	106.2125	107.9224
Tai_chi_S2	79.3635	118.2341	114.8655
Tai_chi_S3	99.99	127.516	161.056
Tai_chi_S4	95.3349	124.6166	136.334
Tai_chi_S5	99.2752	120.4779	122.3163
Tai_chi_S6	100.1354	123.6235	124.6892
Average	**87.94375**	122.2645	139.4058

Fig. 14 shows the distribution of error distance when estimating 3-D human pose on the MADS dataset with 15 key points to evaluate in each frame.

Table VI and Fig. 14 are shown the results of 3-D human pose estimation on the first method "3-D_COCO_Method", is much better the second method "**3-D_HUMAN3.6_Method**" [35] and the thirs method "**3-D_VNECT_Method**" [34]. The average error value (MPJPE) of "**3-D_COCO_Method**" method is 87.94375 mm. This method uses the output of 2-D human pose estimation (**Method 1** in Fig. 6 as the input for 3-D human pose estimation in the method of Tome *et al.* [35]. By measuring, when the 2-D human pose estimation results are good, then the results of estimation, recovering 3-D human pose is good. The "**3-D_VNECT_Method**" method has the lowest result, the average error value (MPJPE) is 139.4058 mm.

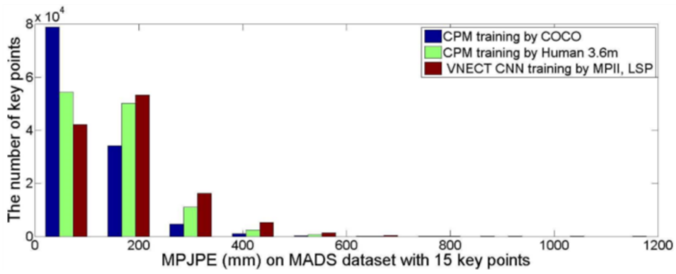

Fig. 14. The distribution of error distance between the estimated key points and the key points of the ground truth data in the 3-D space on the MADS dataset. Where "CPM training by COCO" is "3-D_COCO-Method", "CPM training by Human 3.6m" is "3-D_HUMAN3.6_Method", "VNECT CNN training by MPII, LSP" is "3-D VNECT Method".

Fig. 15. Illustration of 2-D human pose estimation result of the "3-D VNECT" method on the image of the MADS dataset with 21 key points.

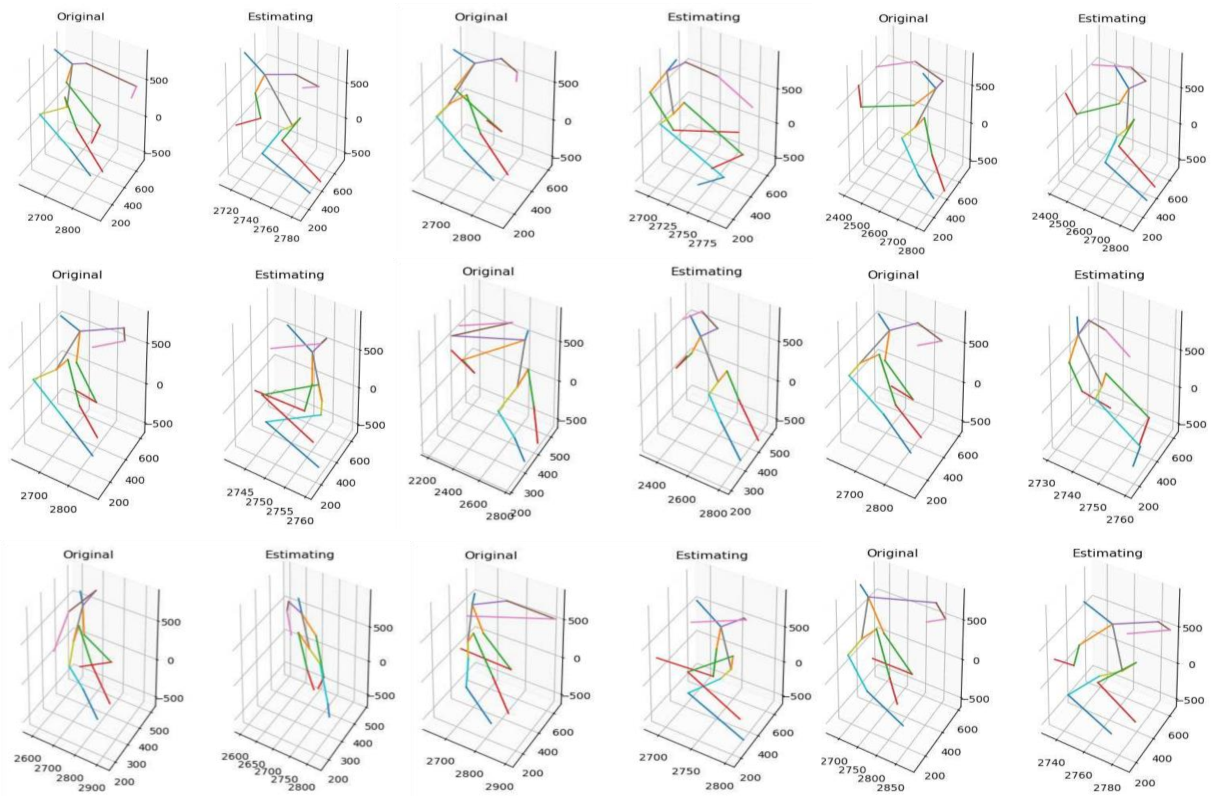

Fig. 16. The results of 3-D human pose estimation. Each block is a pair of correspondences between the 3-D pose of the ground truth data (ground truth - original) and the estimated 3-D human pose (estimating). Each pair of frames in a block has been synchronized to the coordinate system.

The process of checking every step of the implementation of the "3-D VNECT Method" method, we found that the results of 2-D human pose estimation are low, illustrated in Fig. 15, the estimated results of key points is the outside of human data.

Fig. 16 shows several 3-D human pose estimation results on the MADS dataset with 17 key points.

V. Conclusion and Future Work

The preservation, storage and teaching of traditional martial arts are very important in preserving national cultural identities and training health and self-defense of people. However, the actions of the body (body, arms, legs) of a martial arts instructor are not always clear. There are many hidden joints. In this paper, we surveyed, summarized the studies on the 3-D human pose estimation in two methods: 3-D human pose estimation from an image or a sequence of images. We proposed two comparative studies for 2-D human pose estimation and 3-D pose estimation. In comparative studies, when there are good 2-D human pose estimation results, then there will be good 3-D human pose estimation results. The average of errors distance of 3-D human pose estimation when using CPM trained on the COCO [42] dataset is 87 mm.

Conflict of Interest

The article is a private result of the author, not owned by any organization or individual. It is part of a series of studies for 3-D human pose estimation problem.

Author Contributions

The article was written based on the author's long-time understanding of 3-D human pose estimation. This paper has only authors and no additional participants.

Acknowledgment

This research is funded in part by Vietnam National Foundation for Science and Technology (NAFOSTED).

References

[1] N. Sarafianos, B. Boteanu, B. Ionescu, and I. A. Kakadiaris, "3D human pose estimation: A review of the literature and analysis of covariates," *Computer Vision and Image Understanding*, vol. 152, no. vii, pp. 1–20, 2016.

[2] H. Rhodin, M. Salzmann, and P. Fua, "Unsupervised geometry-aware representation for 3D human pose estimation," *Lecture Notes in Computer Science*, vol. 11214, pp. 765–782, 2018.

[3] W. Gong, X. Zhang, J. Gonzalez, A. Sobral, T. Bouwmans, C. Tu, and E. H. Zahzah, "Human pose estimation from monocular images: A comprehensive survey," *Sensors*, vol. 16, no. 12, pp. 1–39, 2016.

[4] M. Rantz, T. Banerjee, E. Cattoor, S. Scott, M. Skubic, and M. Popescu, "Automated fall detection with quality improvement "rewind" to reduce falls in hospital rooms," *J Gerontol Nurs*, vol. 40, no. 1, pp. 13–17, 2014.

[5] R. IgualCarlos, M. Carlos, and I. Plaza, "Challenges, issues and trends in fall detection systems," *BioMedical Engineering OnLine*, vol. 12, no. 1, pp. 147–158, 2013.

[6] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, 2016,.

[7] S. Jian *et al.*, "Convolutional pose machines," in *Proc.* I*EEE Conference on Computer Vision and Pattern Recognition*, 2016.

[8] W. Zhang, Z. Liu, L. Zhou, H. Leung, and A. B. Chan, "Martial Arts, dancing and sports dataset: A challenging stereo and multi-view dataset for 3d human pose estimation," *Image and Vision Computing*, vol. 61, 2017.

[9] D. Pavllo, C. Feichtenhofer, D. Grangier, and M. Auli, "3D human pose estimation in video with temporal convolutions and semi-supervised training," in *Proc. Conference on Computer Vision and Pattern Recognition*, 2019.

[10] L. Wang, Y. Chen, Z. Guo, K. Qian, M. Lin, H. Li, and J. S. Ren, "Generalizing monocular 3d human pose estimation in the wild," arXiv preprint arXiv:1904.05512, 2019.

[11] K. Iskakov, E. Burkov, V. S. Lempitsky, and Y. Malkov, "Learnable triangulation of human pose," *CoRR*, vol. abs/1905.05754, 2019.

[12] M. R. I. Hossain and J. J. Little, "Exploiting temporal information for 3D human pose estimation," *Lecture Notes in Computer Science*, vol. 11214, pp. 69–86, 2018.

[13] 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 Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 7, pp. 1325–1339, July 2014.

[14] M. Kocabas, S. Karagoz, and E. Akbas, "Self-supervised learning of 3D human pose using multi-view geometry," *IEEE Computer Vision and Pattern Recognition*, 2019.

[15] B. Wandt and B. Rosenhahn, "Repnet: Weakly supervised training of an adversarial reprojection network for 3d human pose estimation," *CoRR*, vol. abs/1902.09868, 2019.

[16] A. Nibali, Z. He, S. Morgan, and L. Prendergast, "3D human pose estimation with 2D marginal heatmaps," in *Proc. 2019 IEEE Winter Conference on Applications of Computer Vision*, 2019, pp. 1477–1485.

[17] S. Johnson and M. Everingham, "Clustered pose and nonlinear appearance models for human pose estimation," in *Proc. BMVC*, 2010, pp. 1–11.

[18] D. Mehta, H. Rhodin, D. Casas, P. Fua, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation in the wild using improved cnn supervision," in *Proc. 2017 Fifth International Conference on 3D Vision (3DV)*, 2017.

[19] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2014, pp. 3686–3693.

[20] L. Sigal, A. O. Balan, and M. J. Black, "HUMANEVA: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *International Journal of Computer Vision*, vol. 87, no. 1, 2010.

[21] M. Burenius, J. Sullivan, and S. Carlsson, "3D pictorial structures for multiple view articulated pose estimation," in *Proc. 2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[22] C. Plagemann, "Real time motion capture using a single time-of-flight camera," in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 2010.

[23] D. K. S. T. V. Ganapathi and C. Plagemann, "Real-time human pose tracking from range data," in *Proc. ECCV*, 2012.

[24] D. Mehta, O. Sotnychenko, F. Mueller, W. Xu, S. Sridhar, G. Pons-Moll, and C. Theobalt, "Single-shot multi-person 3d pose estimation from monocular rgb," in *Proc. 2018 Sixth International Conference on3D Vision (3DV)*.

[25] M. Veges, V. Varga, and A. Lorincz, "3d human pose estimation with siamese equivariant embedding," arXiv preprint arXiv:1809.07217, 2018.

[26] K. Wang, L. Lin, C. Jiang, C. Qian, and P. Wei, "3D human pose machines with self-supervised learning," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[27] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 2659–2668.

[28] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 1263–1272.

[29] C. Li and G. H. Lee, "Generating multiple hypotheses for 3d human pose estimation with mixture density network," in *Proc. The IEEE Conference on Computer Vision and Pattern Recognition*, June 2019.

[30] H.-S. Fang, Y. Xu, W. Wang, X. Liu, and S.-C. Zhu, "Learning pose grammar to encode human body configuration for 3D Pose estimation," in *Proc. Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.

[31] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," in *Proc. the IEEE International Conference on Computer Vision*, 2017, pp. 3961–3970.

[32] M. Omran, C. Lassner, G. Pons-Moll, P. Gehler, and B. Schiele, "Neural body fitting: Unifying deep learning and model based human pose and shape estimation," in *Proc.* 2018 *International Conference on 3D Vision*, 2018, pp. 484–494.

[33] D. Pavllo, D. Grangier, and M. Auli, "Quaternet: A quaternion-based recurrent model for human motion," in *Proc. British Machine Vision Conference (BMVC)*, 2018.

[34] D. Mehta, S. Sridhar, O. Sotnychenko, H. Rhodin, M. Shafiei, H.-P. Seidel, W. Xu, D. Casas, and C. Theobalt. (2017). Vnect: Real-time 3d human pose estimation with a single rgb camera. [Online]. 36(4). Available: http://gvv.mpi-inf.mpg.de/projects/VNect/

[35] D. Tome, C. Russell, and L. Agapito, "Lifting from the deep: Convolutional 3D pose estimation from a single image," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5689–5698.

[36] C. H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 5759–5767.

[37] M. Veges and A. Lorincz, "Absolute human pose estimation with depth prediction network," *CoRR*, vol. abs/1904.05947, 2019.

[38] X. Sun, C. Li, and S. Lin, "An integral pose regression system for the ECCV2018 posetrack challenge," in *Proc. ECCV*, 2018, pp. 1–5.

[39] A. Haque, B. Peng, Z. Luo, A. Alahi, S. Yeung, and L. Fei-Fei, "Towards viewpoint invariant 3D human pose estimation," *Lecture Notes in Computer Science*, vol. 9905 LNCS, pp. 160–177, 2016.

[40] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh, *Realtime Multi-Person 2D Pose Estimation Using Part Affinity Field*, 2017.

[41] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. CVPR*, 2016.

[42] COCO. (2019). Observations on the calculations of COCO metrics. [Online]. Available: https: //github.com/cocodataset/cocoapi/issues/56

[43] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," *Advances in Neural Information Processing Systems*, pp. 1799–1807, 2014.

[44] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," *Lecture Notes in Computer Science*, vol. 9908, pp. 630–645, 2016.

[45] J. Kim. (2019). ResNet-Tensorflow. [Online]. Available: https://github.com/taki0112/ ResNet-Tensorflow

[46] S. Johnson and M. Everingham, "Learning effective human pose estimation from inaccurate annotation," in *IEEE Proc. CVPR*, 2011.

[47] C. Ionescu, J. Carreira, and C. Sminchisescu, "Iterated secondorder label sensitive pooling for 3d human pose estimation," in *Proc. 30th IEEE Conference on Computer Vision and Pattern Recognition*, , 2014.

[48] D. Mehta, H. Rhodin, D. Casas, O. Sotnychenko, W. Xu, and C. Theobalt, "Monocular 3d human pose estimation using transfer learning and improved CNN supervision," *CoRR*, vol. abs/1611.09813, 2016.

[49] Z. Cao, T. Simon, S.-E. Wei, and Y. Sheikh. (2019). Realtime multi-person pose estimation. [Online]. Available: https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation

[50] Geometric. (2019). *Geometric Transformations*. [Online]. Available: https://pages.mtu.edu/shene/COURSES/cs3621/NOTES/geometry/geo-tran.html

[51] G. Forgeeks. (2019). *Linear Regression (Python Implementation)*. [Online]. Available: https://www.geeksforgeeks.org/linear-regression-python-implementation/

[52] Linear. (2019). *Linear Regression*. [Online]. Available: https://machinelearningcoban.com/2016/ 12/28/linearregression/

Van-Hung Le received his M.Sc. at the Faculty Information Technology- Hanoi National University of Education in 2013. He is a PhD at International Research Institute MICA - HUST, CNRS/UMI 2954 - INP Grenoble (2018). He is now a lecturer of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; Deep learning; Estimation algorithm.