VOLUME: 3 | ISSUE: 3 | 2019 | September

3D HUMAN POSE ESTIMATION IN VIETNAMESE TRADITIONAL MARTIAL ART VIDEOS

Tuong Thanh NGUYEN [1,2], *Van-Hung LE* [*,3], *Duy-Long DUONG* [1], *Thanh-Cong PHAM* [1], *Dung LE* [1]

[1]Hanoi University of Science and Technology
[2]Quy Nhon University
[3]Tan Trao University

*Corresponding Author: Van-Hung Le (email: van-hung.le@mica.edu.vn)
(Received: 3-Aug-2019; accepted: 25-Sep-2019; published: 30-Sep-2019)
DOI: http://dx.doi.org/10.25073/jaec.201933.252

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 MSCOCO Keypoints Challenge dataset [1], Human 3.6m [2], MPII dataset [3], LSP [4], [5], etc. We final publish the dataset of Vietnamese's traditional martial arts in Binh Dinh province for evaluating the 3-D human pose estimation. Quantitative results are presented and evaluated.*

Keywords

3-D Key points estimation, 3-D Human Pose estimation, Convolutional Neural Network (CNN), Conserving and teaching traditional martial arts.

1. 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 [6]. 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 [7].

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 [8]. The above studies often use color images, depth images [9], or skeleton [10] obtained from different types of sensors (e.g, Microsoft (MS) Kinect version 1, MS Kinect version 2, Time-of-Flight-Sensors).

In particular, Microsoft (MS) Kinect sensor version 1(v1) is a common and cheap sensor that can collect information such as color images, depth images, skeleton and acceleration vectors [11].

In this paper, the main contributions are: (1) We survey on recent 3D human pose estimation techniques in the recently years by 3-D human pose estimation; (2) We propose a comparative study for 3-D human pose estimation based on the method that uses a single image, they captured MS Kinect sensor v1; (3) We propose measures to evaluate and publish the dataset of Vietnamese's traditional martial arts in Binh Dinh province. This paper is structured as follows: the first is the introduction of 3-D human pose estimation (Section 1.); the second is the literature review of some studies of 3-D human pose estimation in recently years (Section 2.); the third is the comparative study of 3-D human pose estimation on the Vietnamese's traditional martial arts dataset (Section 3.); the final is some conclusions and discussions (Section 5.).

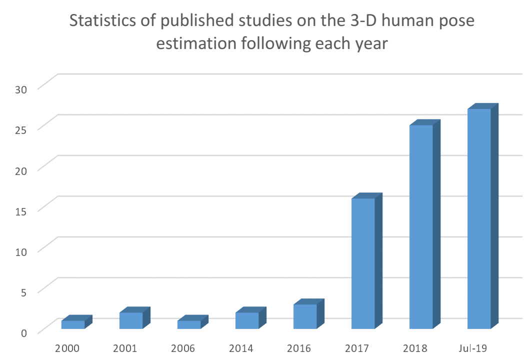

Fig. 1: Statistics of published studies on the 3-D human pose estimation following each year

2. 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) [6]. 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 Figure 3 of [6]. 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. [12], Wang et al. [13], etc) or use the 2-D human pose annotation (second method) (studies by Karim et al. [14], Hossain et al. [15], 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 3-D pose library as [2] and then find the most suitable 3-D pose;

The second method projected the 3-D space by the parameters of captured sensors [16] or using a CNN model [17].

In particular, most studies of 3-D human pose estimation are evaluated on the Human3.6m dataset [2] with the following common measurements: MPJPE (Mean Per Joint Position Error) [12], PCK (Percentage of Correct Keypoints), and AUC (Area Under Curve) [18], PMPJPE (Procrustes Aligned Mean Per Joint Position Error) [16], etc. These studies are often evaluated on datasets such as: Human3.6m [2], LSP [19], 3DHP [20], MPII [3], HumanEva-I [21], Football II [22], Invariant-Top View [23, 24], MPI-INF3DHP [20], MuPoTS-3D [25], AIChallenger [13].

3-D human pose estimation result was based on $MPJPE$ measurement, as shown in Tab. 1.

Tab. 1: Statistics of the result of studies based on the MPJPE(mm) measurement on the Human3.6m dataset [2] for 3-D human pose estimation.

Method	Results of Mean Per Joint Position Error (MPJPE) (mm)
Pavllo et al. [12]	Protocol 1: 51.8 Protocol 2: 40.0
Liu et al. [26]	61.1
Nibali et al. [18]	57.0
Veges et al. [27]	Protocol #1: 61.1
Wang et al. [28]	Protocol #1: 63.67
Martinez et al. [29]	protocol #1: 45.5
Pavlakos et al. [30]	51.9
Wang et al. [13]	Protocol#1: 40.8
Hossain et al. [15]	Protocol #1: 39.2
Li et al. [31]	Protocol #1: 52.7 Protocol #2: 42.6
Karim et al. [14]	Protocol 1: 49.9
Fang et al. [32]	Protocol #1: 60.4 Protocol #2: 45.7 Protocol #3: 72.8
Tekin et al. [33]	50.12
Omran et al. [34]	59.9
Pavllo et al. [35]	36
Bastian et al. [17]	Protocol #1: 50.9
Kocabas et al. [16]	51.83
Rhodin et al. [7]	131.7
Mehta et al. [36]	ResNet 100: 82.5 ResNet 50: 80.5
Tome et al. [37]	Protocol #1: 88.39 Protocol #2: 70.4 Protocol #3: 79.6

2.1. 3-D human pose estimation from a single image

As reported in the survey of Sarafianos et al. [6], 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.

2.2. 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.

2.3. Traditional martial arts and datasets

In Vietnam [42], [43] as well as many countries in the world like China [44], Japan, Thailand, there are many martial arts postures or martial arts that need to be preserved and passed down to posterity. Conservation and storage in the era of technology can be done in many different ways. An intuitive approach is to save the bone joints in the skeleton model of martial arts instructor. Data obtained from MS Kinect sensor v1 usually contains a lot of noise and is lost when obscured, especially skeleton data of people. The skeleton data is important and presents human pose in video action.

Recently, Zhang et al. [45] published the benchmark dataset 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

Tab. 2: 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. [12]	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	Liu et al. [26]	No	The feature boosting network, the state-of-the-art Hourglass CNN is adopted to learn the convolutional features for the RGB image. The feature maps to perceive the graphical long short-term dependency among different hand (or body) parts using the designed Graphical ConvLSTM. 3D human pose estimation: the 2-D heatmaps first as an intermediate representation for inferring the final 3-D pose.	Human3.6m MPI-INF-3DHP	MPJPE
2019	Nibali et al. [18]	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. [28]	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. [39]	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. [13]	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. [31]	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 3DHP	MPJPE
2018	Veges et al. [27]	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. [40]	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. [32]	Yes	For 2D pose estimation, existing large-scale pose estimation datasets (Andriluka et al. 2014; Charles et al. 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. [34]	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. [35]	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. [29]	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. [30]	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. [33]	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. [41]	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

Tab. 3: Survey: 3-D human pose estimation from a sequence of images.

Year	Main Author/ reference	Using 3-D pose library	Method Highlights	Evaluation dataset	Evaluation matrix
2019	Karim et al. [14]	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
2019	Bastian et al. [17]	Yes	One part of the proposed reprojection network (RepNet) learns a mapping from a distribution of 2D poses to a distribution of 3D poses using an adversarial training approach.	Human3.6m MPI-INF-3DHP LSP	MPJPE
2019	Kocabas et al. [16]	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
2018	Rhodin et al. [7]	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
2018	Hossain et al. [15]	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

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 [21] 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.

3. 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 $\mathbf{L_c}$ vector in 2-D space (image space) in a set of vectors on human body $\mathbf{S}$, where the set of vectors $\mathbf{L} = \{\mathbf{L_1}, \mathbf{L_2}, ..., \mathbf{L_C}\}$, has $\mathbf{C}$ vectors on human body $\mathbf{S}$. The body of $\mathbf{S}$ is represented by the key points $\mathbf{J}$, $\mathbf{S} = \{\mathbf{S_1}, \mathbf{S_2}, ..., \mathbf{S_J}\}$. For an input image of size $(\mathbf{w} \times \mathbf{h})$ pixels, the position of the key points can be $\mathbf{S_j} \in \mathbf{R^{w \times h}}, \mathbf{j} \in \{\mathbf{1}, \mathbf{2}, ..., \mathbf{J}\}$. 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 $\mathbf{F}$ 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 $\mathbf{F}$ 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" [46] to estimate 2-D human pose.

The detailed model of training and predicting (Figure 3) of Zhe's study [47] 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 $9 \times 9, 2\times, 5\times, ...$ 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

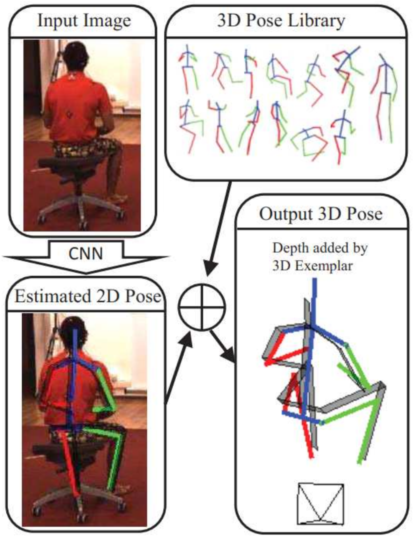

Fig. 2: Illustration of method for 3-D human pose estimation [38]: the input is a RGB image, the first estimate a 2-D pose and then estimate its depth by matching to a library of 3-D poses. The final prediction is given by the colored skeleton based on the 3-D poses library, while the ground-truth is shown in gray.

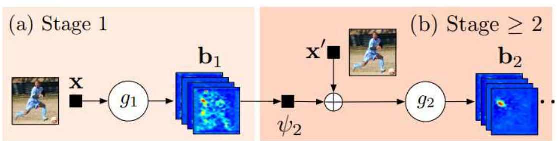

Fig. 3: Illustration of the detail model to predict the heatmaps [48].

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.

In this paper, we conduct a comparative study of 3-D human pose estimation, as is shown in Fig. 6. In which the methods are presented as follows:

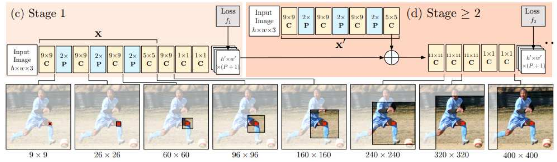

Fig. 4: Illustration of the detail model to extract the feature for training model and to predict the heatmaps at each stage [48].

- The first method is called "**3-D_COCO_Method**": 2-D human pose estimation by using CPM that was trained on the MSCOCO Key points Challeng [1] dataset + mapping to 3-D space by 3-D pose library of Human 3.6m dataset [37].

- 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 [2] + mapping to 3-D space by 3-D pose library of Human 3.6m dataset [37].

- 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. [36].

The method of Tome et al. [37] implemented the process of 3-D human pose estimation based on mapping the 2-D 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 e and noise variance σ, along with each per sample rotations R_i and basis coefficients a_i

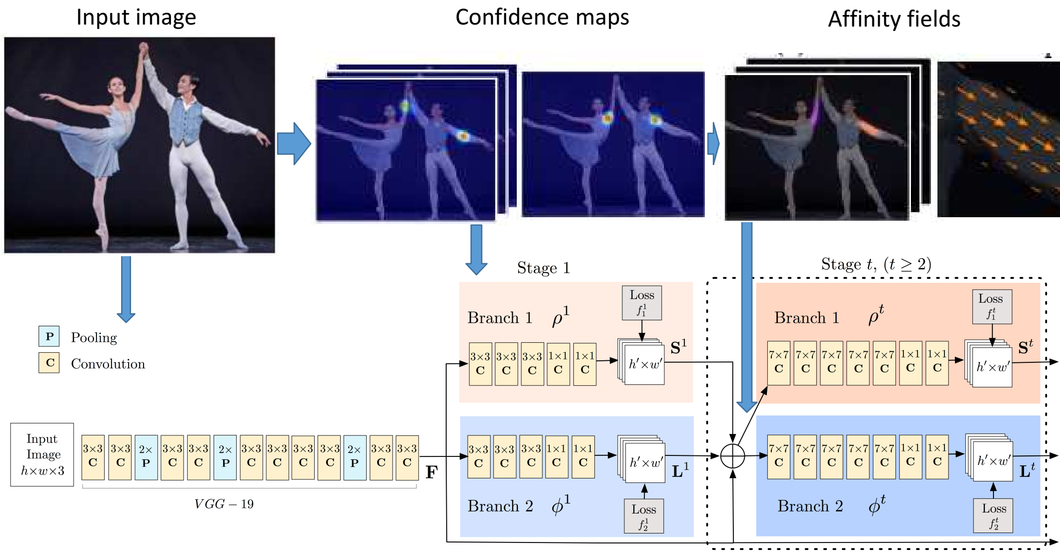

Fig. 5: The architecture of the two-branch multi-stage CNN for training the model estimation [47]

to minimize the following estimate as Eq. 1.

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

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. [36], The authors use the regression CNN model to predict the heatmaps by method of Tompson et al. [49] Especially the training of features for learning and predicting the map highlights is based on ResNet (Deep Residual Networks) network [50], which provides a breakthrough idea for building Characteristic and training. The ResNet in [50] is built on the platform of Tensorflow library of [51]. The model in this network uses the MPII dataset [3], LSP [4], [5] 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. [52] with the use of Human3.6m dataset [2] and MPI-INF-3DHP [53] for projecting 2-D human pose estimation to 3-D space.

4. Experimental Results

4.1. Data collection and evaluation

Traditional martial arts, a very important sport, help people exercise and protect themselves. In many countries around the world, especially in Asia, there are many traditional martial arts handed down from generation to generation. With the development of technology, it is important to maintain, preserve and teach such martial arts [54], [55]. There are also many different types of image sensors that can collect information about martial arts teaching and learning of the schools of martial arts. The MS Kinect sensor v1 is the cheapest sensor. This type of sensor can collect a lot of information such as color images, depth images, skeleton, acceleration vectors, sounds, etc. From the collected data, it is possible to recreate the environment in 3-D space about teaching martial arts in the schools of martial arts. However, in this paper, based on

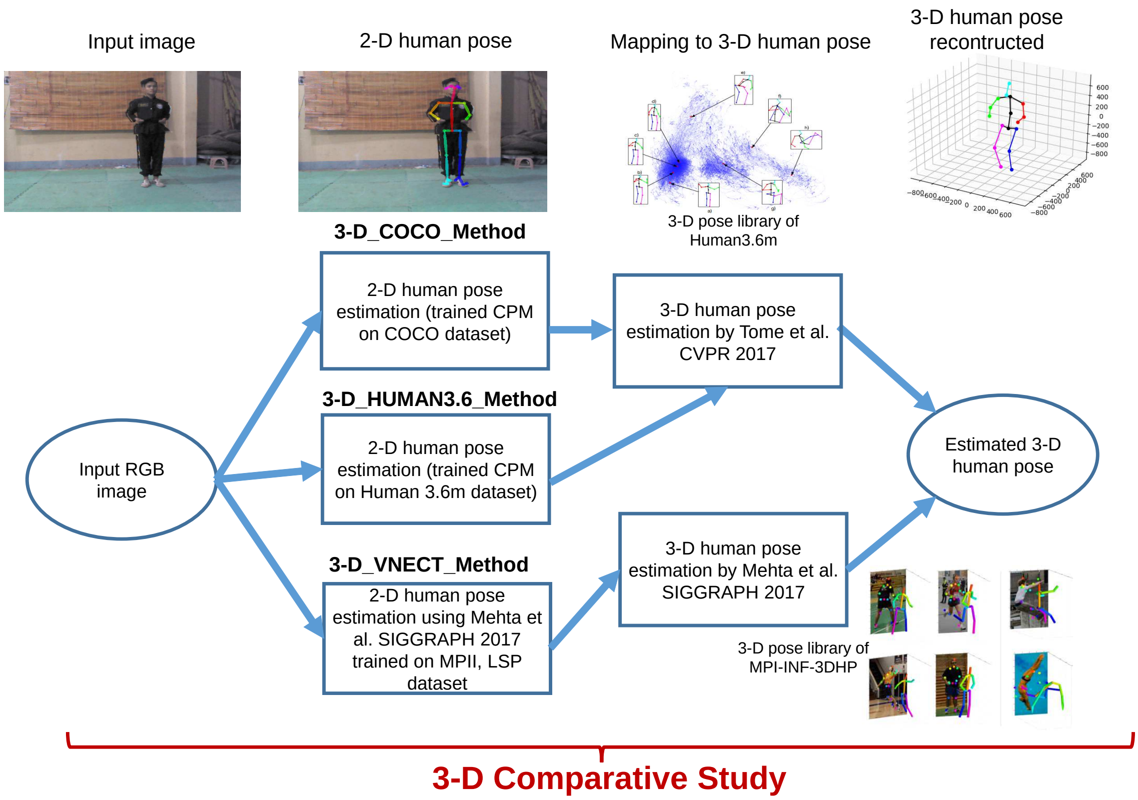

Fig. 6: Comparative study for evaluating 2-D human pose estimation in the 3-D space.

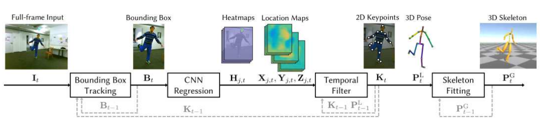

Fig. 7: Illustration of VNect network [36].

the information collected from the MS Kinect sensor, we only use color images for the construction of this study. To obtain data from the sensor environment, the MS Kinect SDK 1.8 is used to connect computers and sensors [56]. To perform data collection on computers, we use a data collection program developed at MICA Institute [57] with the support of the OpenCV 3.4 libraries [58], C++ programming language. Between the sensors of color images, depth images, and the skeleton. Therefore, it is recommended to make a calibration to take the data on color images and depth images; particularly, we apply the data calibration of Zhou et al. [59] and Jean et al. [60]. In these two calibration tools, the calibration matrix is used as follows:

$$H_m = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \tag{2}$$

where (c_x, c_y) is the principle point (usually the image center), f_x and f_y are the focal lengths. The matrix H_m (in Nicolas et al. [61]) is calculated as follows:

$$H_m = \begin{bmatrix} 594.214 & 0 & 339.307 \\ 0 & 591.040 & 242.739 \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

In this dataset we also provided the 3-D pose annotation. The ground truth data of key points is marked on data in 3-D space. To do this, we showed 3-D data (point cloud data) of the scene on the visualization window of a program that we developed based on the Visual Studio programming environment and the support of the PCL library [62] with c ++ programming language. Figure 8 illustrates the 3-D human pose data. We marked 17 key points on the human body. In some cases when the limbs are obscured, we assume that the person's hands or feet, are often close to the human body and they are chosen as in the case of hand or foot data being seen. Currently marking points in 3-D space is manually done, only considering the data of one side of the MS Kinect sensor. This study has not looked into cases when the data is obscured and when the actions of people are complicated. In order to mark data in 3-D space when obscured, which is often used MOCAP system [63] for calculating the actual coordinates of human hands and feet.

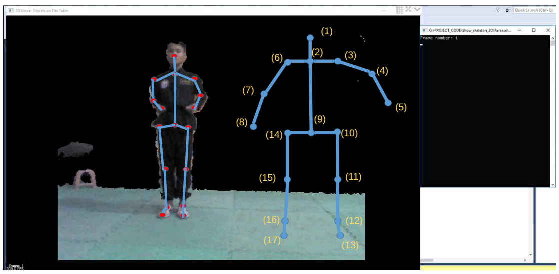

Fig. 8: Illustration marking of 3-D human pose annotation, in which the order of marking of key points is as follows: (1) Head, (2) Neck, (3) Right Shoulder, (4) Right Elbow, (5) Right Wrist, (6) Left Shoulder, (7) Left Elbow, (8) Left Wrist, (9) Center Hip, (10) Right Hip, (11) Right Knee, (12) Right Ankle, (13) Right Big Toe, (14) Left Hip, (15) Left Knee, (16) Left Ankle, (17) Left Big Toe.

The dataset is collected from a MS Kinect sensor v1, it can collect data at a rate of about 10 frames/s on a low-configuration Laptop. MS Kinect sensor v1 is mounted on a fixed rack; martial arts instructor represents a space of about 3×3m as Fig. 9 and calls "VNMA - VietNam Martial Arts".

The obtained images (color images, depth images) are 640×480 pixels. The obtained data set consists of 24 videos of different postures

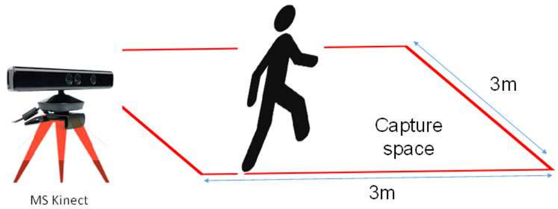

Fig. 9: Illustration of MS Kinect sensor v1 settings and data collection space.

with 24 subjects (12 males and 12 females), with the number of frames listed in Tab. 4. This dataset was collected at a martial arts school in Binh Dinh Province, Vietnam. In this dataset, we also provided point cloud data of each scene corresponding to each frame obtained. The entire dataset can be downloaded at this link: https://drive.google.com/file/d/1dIHgal63TcGn0-6_hnTJsEDfh8qkNOsE/view?usp=sharing

In this paper, we use a trained model on the 2016 MSCOCO Key points Challenge dataset [1] for 2-D human pose estimation of the first method "3-D_HUMAN3.6_Method". The training models are based on the published OpenPose [64]. The parameters of training the whole CNN network are as follows: the size of the input image is (width: $368\times$ height: $368\times$ channel: 3); $batchSize = 16$; $stacks = 4$; the number of stages is 6 for pooling; 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. We also a trained model on the Human 3.6m dataset [2] for 2-D human pose estimation of the second method "3-D_HUMAN3.6_Method". The parameters of training the whole CPM are provided 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 are shown in the link: https://github.com/DenisTome/Lifting-from-the-Deep-release/

Tab. 4: Number of frames in martial arts postures of VNMA database.

Video number	1	2	3	4	5	6	7	8	9	10	11	12
Number of frames	50	89	71	77	98	109	87	79	89	76	79	95
Video	**13**	**14**	**15**	**16**	**17**	**18**	**19**	**20**	**21**	**22**	**23**	**24**
The number of frames	131	71	95	101	108	117	109	112	80	110	96	105

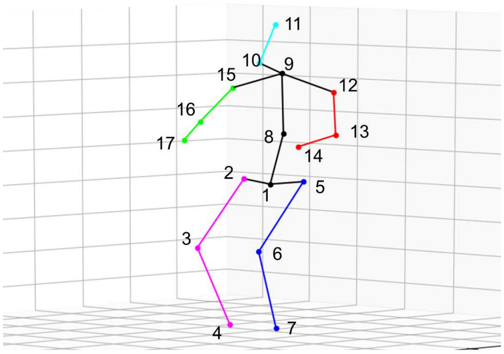

Fig. 10: The output of 3-D human pose estimation based on the method of Tome et al. [37]

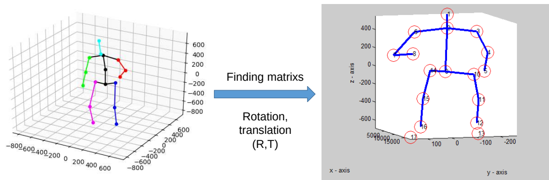

Fig. 11: Illustration of finding the rotation, translation matrix in the 3-D space.

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 [65] as Eq. 4

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 3-D human pose estimation based on the method of Tome et al. [37] (the methods: "3-D_COCO_Method", "3-D_HUMAN3.6_Method") is 17 key points, as shown in Fig. 10. The output of 3-D human pose estimation based on the method of Mehta et al. [36] (the methods: "3-D_VNECT_Method") is 21 key points.

There is the fact that the 3-D ground truth data follows the MS Kinect coordinate system, 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 dataset or MPI-INF-3DHP dataset [53] . These two types of data are not in the same coordinate system, and thus take steps to the synchronized coordinate system.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} R_{11} & R_{12} & R_{13} & T_1 \\ R_{21} & R_{22} & R_{23} & T_2 \\ R_{31} & R_{32} & R_{33} & T_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \tag{4}$$

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 as in Eq. (5).

$$\left\{ \begin{array}{lcl} x' & = & R_{11}x + R_{12}y + R_{13}z + T_1 \\ y' & = & R_{21}x + R_{22}y + R_{23}z + T_2 \\ z' & = & R_{31}x + R_{32}y + R_{33}z + T_3 \end{array} \right\} \tag{5}$$

From the coordinates of the key points in the 3-D human pose of the dataset, we define the coordinates of a 3-D pose including n points as in Eq. (6).

$$\begin{bmatrix} 1 & z_1 & y_1 & x_1 \\ 1 & z2 & y_2 & x_2 \\ . & . & . & . \\ . & . & . & . \\ . & . & . & . \\ 1 & z_n & y_n & x_n \end{bmatrix} \tag{6}$$

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. (7).

$$\theta_1 = \begin{bmatrix} T_1 \\ R_{13} \\ R_{12} \\ R_{11} \end{bmatrix} \theta_2 = \begin{bmatrix} T_2 \\ R_{23} \\ R_{22} \\ R_{21} \end{bmatrix} \theta_3 = \begin{bmatrix} T_3 \\ R_{33} \\ R_{32} \\ R_{31} \end{bmatrix} \tag{7}$$

The results of rotation and translation are shown in the vector X', Y', Z' as in the Eq. (8).

$$X' = \begin{bmatrix} x'_1 \\ x'_2 \\ . \\ . \\ . \\ x'_n \end{bmatrix} Y' = \begin{bmatrix} y'_1 \\ y'_2 \\ . \\ . \\ . \\ y'_n \end{bmatrix} Z' = \begin{bmatrix} z'_1 \\ z'_2 \\ . \\ . \\ . \\ z'_n \end{bmatrix} \tag{8}$$

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.

From this, we have a system of linear equations presented in the Eq. (9).

$$\begin{aligned} X' &= M\theta_1 \\ Y' &= M\theta_2 \\ Z' &= M\theta_3 \end{aligned} \tag{9}$$

In which the estimation θ_i is the using the Least Squares method (LS) [66], [67] as in Eq. (10).

$$\begin{aligned} \theta_1 &= (M^T M)^{-1} M^T X' \\ \theta_2 &= (M^T M)^{-1} M^T Y' \\ \theta_3 &= (M^T M)^{-1} M^T Z' \end{aligned} \tag{10}$$

The entire source of the rotation and translation is stored in the path: https://drive.google.com/file/d/1dIHgal63TcGn0-6_hnTJsEDfh8qkNOsE/view?usp=sharing and explained in detail in appendix A and appendix B. 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. [37] method and the output key points of Mehta et al. [36] method as in Fig. 12.

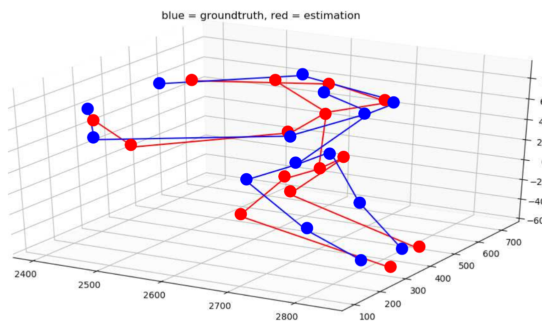

Fig. 12: Illustrating 3-D human pose for evaluating 3-D human pose estimation. The blue key points are ground truth data, the red key points are the estimated data which transformed the same coordinate system.

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. 11.

$$D(p_g, p_e) = \sqrt{(x_g - x_e)^2 + (y_g - y_e)^2 + (z_g - z_e)^2} \tag{11}$$

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.

4.2. Results of estimation and discussion

The results of 3-D human pose estimation on VNMA database are provided in Tab. 5.

Figure 13 shows the error distance distribution when estimating 3-D human pose on the VNMA database with 15 key points.

Table 5 and Figure 13 reveal that the first method "3-D_COCO_Method" has the best estimation results (the average of MPJPE is 170.866 mm). These error values are high because the 3-D ground truth data is manually analyzed; therefore it is not as accurate as the 3-D ground truth data calculated from the MOCAP system. The third method "3-D_VNECT_Method" has the lowest estimation results (the average of MPJPE is 279.4472 mm). During the testing process, we found that the 2-D human pose estimation result of "3-D_VNECT_Method" method is much wrong as in Fig. 14.

Figure 15 shows several 3-D human pose estimation results on the VNMA dataset with 17 key points.

In particular, 3-D human pose estimation based on the proposed comparative study, has solved the cases when the parts are obscured, 3-D human skeleton is fully restored as in Fig. 16.

5. 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 inviduals' self-defense. 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. Many studies in 3-D human pose estimation used the Human 3.6m dataset for training the models estimation and based on MPJPE measurement for evaluating the errors estimation. Studies from 2016 to 2018 have a tolerance of about 80-150 mm, and use a GPU that can be done. However, studies from 2019 have errors smaller than 80 mm, but the number of GPUs required for training and testing is greater than 1.

We proposed a dataset by the Vietnam martial arts called "VNMA" and proposed a comparative study based on the methods which used the CNN model for estimating 3-D human pose. In particular, studies of 3-D human pose estimation restored the full skeleton even when the joints are obscured.

In the future, we will substantially build this body the in 3-D space with mesh technique. From this, we will build the 3-D videos on Vietnamese traditional martial arts, served for storing, preserving, and teaching martial arts.

References

[1] COCO (2019). Observations on the calculations of COCO metrics. https://github.com/cocodataset/cocoapi/issues/56. [Accessed 24 April 2019].

[2] Ionescu, C., Papava, D., Olaru, V., & Sminchisescu, C. (2014). Human3.6M: Large Scale Datasets and Predictive Methods for 3D Human Sensing in Natural Environments. IEEE Transactions on Pattern Analysis and Machine Intelligence, 36(7), 1325–1339.

[3] Andriluka, M., Pishchulin, L., Gehler, P., & Schiele, B. (2014). 2D Human Pose Estimation New Benchmark and State of the Art Analysis. In: IEEE Conference on Computer Vision and Pattern Recognition.

[4] Johnson, S., & Everingham, M. (2010). Clustered Pose and Nonlinear Appearance

Tab. 5: The results of 3-D human pose estimation on the VNMA dataset with 15 key points.

#Video	MPJPE (mm)		
	3-D_COCO_ Method	3-D_HUMAN3.6_ Method	3-D_VNECT_ Method
1	114.0716	114.0716	228.8319
2	107.5917	111.025	332.8037
3	88.5689	91.536	245.1891
4	78.6414	79.9366	239.818
5	99.0704	101.6908	282.843
6	111.0964	112.0768	292.2822
7	114.7642	118.3664	309.3528
8	285.0776	292.9947	318.6
9	90.6766	92.9212	253.3029
10	280.8594	284.8666	294.9349
11	91.2715	91.2715	249.4076
12	219.4037	219.4037	242.6467
13	89.3462	89.3462	267.3336
14	264.4068	262.0707	271.0392
15	85.9806	87.3728	254.4252
16	318.4422	318.4422	343.7987
17	99.5296	101.7892	271.0186
18	308.1409	310.7236	331.4765
19	110.9321	110.9321	320.2984
20	239.3639	241.5342	271.7371
21	81.9572	81.9572	206.8996
23	103.5087	105.8891	280.5987
24	267.6513	292.217	282.1385
Average	**170.866**	173.7285	279.4472

Models for Human Pose Estimation. In: Proceedings of the British Machine Vision Conference. BMVA Press, 12.1–12.11. Doi:10.5244/C.24.12.

[5] Johnson, S., & Everingham, M. (2011). Learning Effective Human Pose Estimation from Inaccurate Annotation. In: IEEE Proc. CVPR.

[6] Sarafianos, N., Boteanu, B., Ionescu, B., & Kakadiaris, I. A. (2016). 3D Human Pose Estimation : A Review of the Literature and Analysis of Covariates. Computer Vision and Image Understanding, Volume 152(vii), Pages 1–20.

[7] Rhodin, H., Salzmann, M., & Fua, P. (2018). Unsupervised geometry-aware representation for 3D human pose estimation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11214 LNCS. 765–782.

[8] Gong, W., Zhang, X., Gonzàlez, J., Sobral, A., Bouwmans, T., Tu, C., & Zahzah, E. H. (2016). Human Pose Estimation from Monocular Images: A Comprehensive Survey. Sensors (Basel, Switzerland), 16(12), 1–39.

[9] Rantz, M., Banerjee, T., Cattoor, E., Scott, S., Skubic, M., & Popescu, M. (2014). Automated fall detection with quality improvement "rewind" to reduce falls in hospital rooms. J Gerontol Nurs, 40(1), 13–17.

[10] IgualCarlos, R., Carlos, M., & Plaza, I. (2013). Challenges, Issues and Trends in

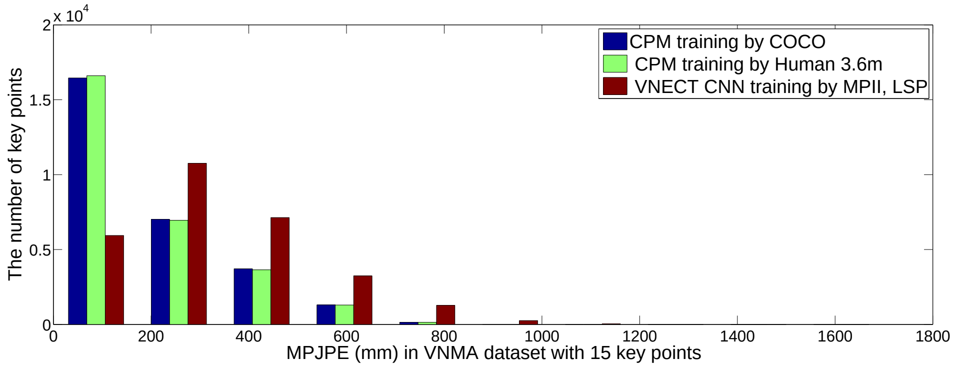

Fig. 13: Error distance distribution between key points on the 3-D ground truth data and the estimated 3-D pose data on the VNMA dataset. where: "CMP training by COCO" is "3-D_COCO_Method" , "CMP training by Human 3.6m" is "3-D_HUMAN3.6_Method", "VNECT CNN training by MPII, LSP" is "3-D_VNECT_Method".

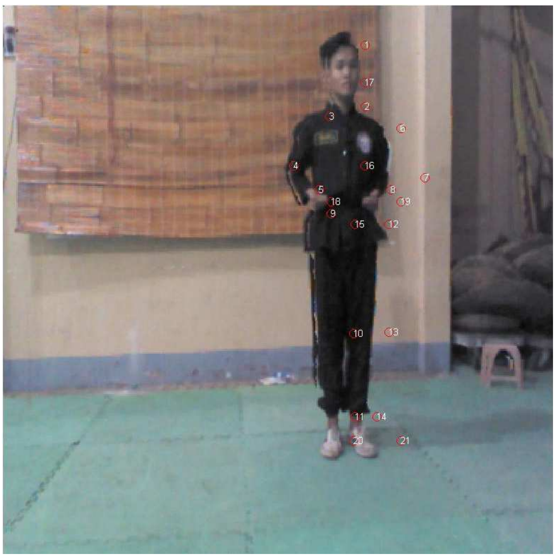

Fig. 14: The result of 2-D human pose estimation based on the method of Mehta et al. [36], 21 key points are predicted.

Fall Detection Systems. BioMedical Engineering OnLine, 12(1), 147–158.

[11] Kramer, J., Parker, M., Castro, D., Burrus, N., & Echtler, F. (2012). Hacking the Kinect. Apress.

[12] Pavllo, D., Feichtenhofer, C., Grangier, D., & Auli, M. (2019). 3D human pose estimation in video with temporal convolutions and semi-supervised training. In: Conference on Computer Vision and Pattern Recognition (CVPR).

[13] Wang, L., Chen, Y., Guo, Z., Qian, K., Lin, M., Li, H., & Ren, J. S. (2019). Generalizing Monocular 3D Human Pose Estimation in the Wild. arXiv preprint arXiv:1904.05512.

[14] Iskakov, K., Burkov, E., Lempitsky, V. S., & Malkov, Y. (2019). Learnable Triangulation of Human Pose. CoRR, abs/1905.05754.

[15] Hossain, M. R. I., & Little, J. J. (2018). Exploiting temporal information for 3D human pose estimation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11214 LNCS. 69–86.

[16] Kocabas, M., Karagoz, S., & Akbas, E. (2019). Self-Supervised Learning of 3D Human Pose using Multi-view Geometry. In: IEEE Computer Vision and Pattern Recognition.

[17] Wandt, B., & Rosenhahn, B. (2019). RepNet: Weakly Supervised Training of an Adversarial Reprojection Network for 3D Human Pose Estimation. CoRR, abs/1902.09868.

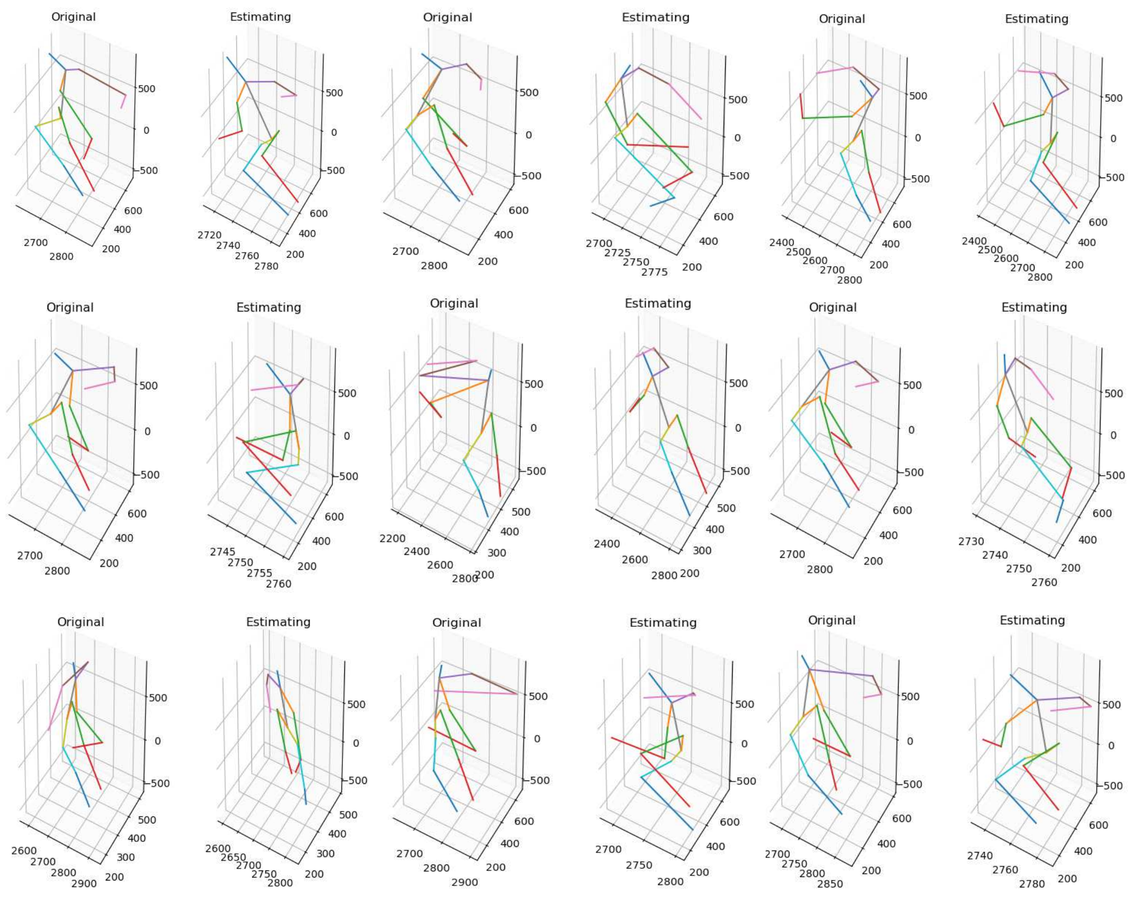

Fig. 15: The results of 3-D human pose estimation. Each block is a pair of correspondences between the 3-D pose of the ground truth data (ground truth - original) and the estimated 3-D human pose (estimating). Each pair of frames in a block has been synchronized to the coordinate system.

[18] Nibali, A., He, Z., Morgan, S., & Prendergast, L. (2019). 3D human pose estimation with 2D marginal heatmaps. In: Proceedings - 2019 IEEE Winter Conference on Applications of Computer Vision, WACV 2019, Figure 1. 1477–1485.

[19] Johnson, S., & Everingham, M. (2010). Clustered Pose and Nonlinear Appearance Models for Human Pose Estimation. In: Proc. BMVC. 12.1–11. Doi:10.5244/C.24.12.

[20] Mehta, D., Rhodin, H., Casas, D., Fua, P., Sotnychenko, O., Xu, W., & Theobalt, C. (2017). Monocular 3D Human Pose Estimation In The Wild Using Improved CNN Supervision. In: 3D Vision (3DV), 2017 Fifth International Conference on.

[21] Sigal, L., Balan, A. O., & Black, M. J. (2010). HUMANEVA: Synchronized Video and Motion Capture Dataset and Baseline Algorithm for Evaluation of Articulated Human Motion. International Journal of Computer Vision, Volume 87(1).

[22] Burenius, M., Sullivan, J., & Carlsson, S. (2013). 3D Pictorial Structures for Multiple View Articulated Pose Estimation. In: 2013 IEEE Conference on Computer Vision and Pattern Recognition.

[23] Plagemann, C. (2010). Real Time Motion Capture Using a Single Time-Of-Flight

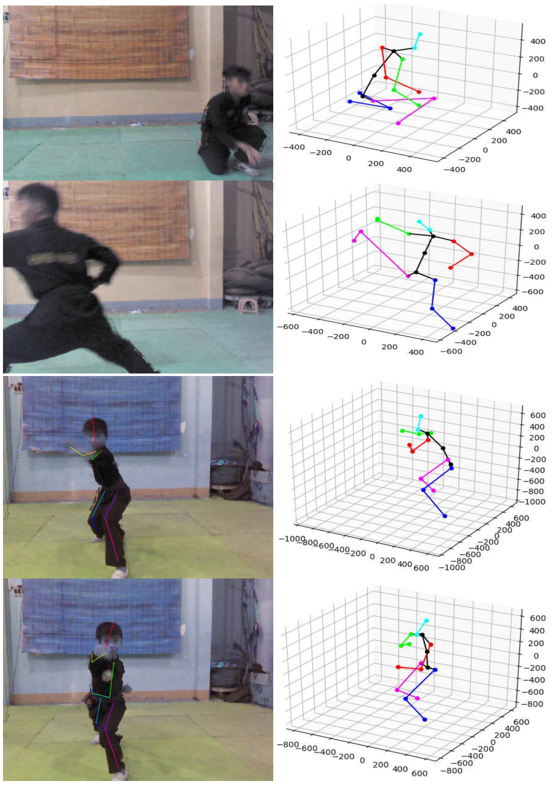

Fig. 16: The results of 3-D human pose estimation, when some parts are obscured.

Camera. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition.

[24] Varun Ganapathi, D. K. S. T., Christian Plagemann (2012). Real-Time Human Pose Tracking from Range Data. In: ECCV.

[25] Mehta, D., Sotnychenko, O., Mueller, F., Xu, W., Sridhar, S., Pons-Moll, G., & Theobalt, C. (2018). Single-Shot Multi-Person 3D Pose Estimation From Monocular RGB. In: 3D Vision (3DV), 2018 Sixth International Conference on. IEEE.

[26] Liu, J., Ding, H., Shahroudy, A., Duan, L.-y., Jiang, X., Wang, G., & Kot, A. C. (2019). Feature Boosting Network For 3D Pose Estimation. IEEE Transactions on Pattern Analysis and Machine Intelligence, January 2019.

[27] Véges, M., Varga, V., & Lőrincz, A. (2018). 3D Human Pose Estimation with Siamese Equivariant Embedding. arXiv preprint arXiv:1809.07217.

[28] Wang, K., Lin, L., Jiang, C., Qian, C., & Wei, P. (2019). 3D Human Pose Machines with Self-supervised Learning. IEEE transactions on pattern analysis and machine intelligence.

[29] Martinez, J., Hossain, R., Romero, J., & Little, J. J. (2017). A Simple Yet Effective Baseline for 3d Human Pose Estimation. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob. 2659–2668.

[30] Pavlakos, G., Zhou, X., Derpanis, K. G., & Daniilidis, K. (2017). Coarse-to-fine volumetric prediction for single-image 3D human pose. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua. 1263–1272.

[31] Li, C., & Hee Lee, G. (2019). Generating Multiple Hypotheses for 3D Human Pose Estimation With Mixture Density Network. In: The IEEE Conference on Computer Vision and Pattern Recognition (CVPR).

[32] Fang, H.-s., Xu, Y., Wang, W., Liu, X., & Zhu, S.-c. (2018). Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation. In: Thirty-Second AAAI Conference on Artificial Intelligence.

[33] Tekin, B., Marquez-Neila, P., Salzmann, M., & Fua, P. (2017). Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In: Proceedings of the IEEE International Conference on Computer Vision, vol. 2017-Octob. 3961–3970.

[34] Omran, M., Lassner, C., Pons-Moll, G., Gehler, P., & Schiele, B. (2018). Neural body fitting: Unifying deep learning and model based human pose and shape estimation. In: Proceedings - 2018 International Conference on 3D Vision, 3DV 2018. 484–494.

[35] Pavllo, D., Grangier, D., & Auli, M. (2018). QuaterNet: A Quaternion-based Recurrent Model for Human Motion. In: British Machine Vision Conference (BMVC).

[36] Mehta, D., Sridhar, S., Sotnychenko, O., Rhodin, H., Shafiei, M., Seidel, H.-P., Xu, W., Casas, D., & Theobalt, C. (2017). VNect: Real-time 3D Human Pose Estimation with a Single RGB Camera. vol. 36.

[37] Tome, D., Russell, C., & Agapito, L. (2017). Lifting from the deep: Convolutional 3D pose estimation from a single image. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua. 5689–5698.

[38] Chen, C. H., & Ramanan, D. (2017). 3D human pose estimation = 2D pose estimation + matching. In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, vol. 2017-Janua. 5759–5767.

[39] Véges, M., & Lörincz, A. (2019). Absolute Human Pose Estimation with Depth Prediction Network. CoRR, abs/1904.05947.

[40] Sun, X., Li, C., & Lin, S. (2018). An Integral Pose Regression System for the ECCV2018 PoseTrack Challenge. In: ECCV. 1–5.

[41] Haque, A., Peng, B., Luo, Z., Alahi, A., Yeung, S., & Fei-Fei, L. (2016). Towards viewpoint invariant 3D human pose estimation. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9905 LNCS. 160–177.

[42] Dinh, T. B. (2017). Bao ton va phat huy vo co truyen Binh dinh: Tiep tuc ho tro cac vo duong tieu bieu. http://www.baobinhdinh.com.vn/viewer.aspx?macm=12&macmp=12&mabb=88043. [Accessed; April, 4 2019].

[43] Dinh, T. B. (2019). Ai ve Binh Dinh ma coi, Con gai Binh Dinh bo roi di quyen. http://www.seagullhotel.com.vn/du-lich-binh-dinh/vo-co-truyen-binh-dinh-5. [Accessed; April, 4 2019].

[44] Chinese (2019). Chinese Kung Fu (Martial Arts). https://www.travelchinaguide.com/intro/martial_arts/. [Accessed; April, 4 2019].

[45] Zhang, W., Liu, Z., Zhou, L., Leung, H., & Chan, A. B. (2017). Martial Arts, Dancing and Sports dataset: a Challenging Stereo and Multi-View Dataset for 3D Human Pose Estimation. Image and Vision Computing, Volume 61.

[46] Wei, S.-E., Ramakrishna, V., Kanade, T., & Sheikh, Y. Convolutional pose machines. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2016) year = 2016,.

[47] Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. (2017). Realtime Multi-Person 2D Pose Estimation using Part Affinity Field.

[48] Wei, S.-e., Ramakrishna, V., Kanade, T., & Sheikh, Y. (2016). Convolutional Pose Machines. In: CVPR.

[49] Tompson, J. J., Jain, A., LeCun, Y., & Bregler, C. (2014). Joint training of a convolutional network and a graphical model for human pose estimation. In: Advances in neural information processing systems. 1799–1807.

[50] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Identity mappings in deep residual networks. In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 9908 LNCS. 630–645.

[51] Kim, J. (2019). ResNet-Tensorflow. https://github.com/taki0112/ResNet-Tensorflow. [Accessed 18 April 2019].

[52] Ionescu, C., Carreira, J., & Sminchisescu, C. (2014). Iterated secondorder label sensitive pooling for 3d human pose estimation.

In: Proceedings - 30th IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2014.

[53] Mehta, D., Rhodin, H., Casas, D., Sotnychenko, O., Xu, W., & Theobalt, C. (2016). Monocular 3D Human Pose Estimation Using Transfer Learning and Improved CNN Supervision. CoRR, abs/1611.09813.

[54] Dinh, T. B. (2011). Preserving traditional martial arts). http://www.baobinhdinh.com.vn/culture-sport/2011/8/114489/. [Accessed 18 April 2019].

[55] Chinese (2012). traditional Chinese martial arts and the transmission of intangible cultural heritage). https://www.academia.edu/18641528/Fighting_modernity_traditional_Chinese_martial_arts_and_the_transmission_of_intangible_cultural_heritage. [Accessed 18 April 2019].

[56] Microsoft (2012). Kinect for Windows SDK v1.8. https://www.microsoft.com/en-us/download/details.aspx?id=40278. [Accessed 18 April 2019].

[57] MICA (2019). International Research Institute MICA. http://mica.edu.vn/. [Accessed 19 April 2019].

[58] Opencv (2018). Opencv library. https://opencv.org/. [Accessed 19 April 2019].

[59] X, Z. (2012). A Study of Microsoft Kinect Calibration. Technical report Dept. of Computer Science George Mason University.

[60] B., J.-Y. (2019). Camera calibration toolbox for matlab. http://www.vision.caltech.edu/bouguetj/calib_doc/. [Accessed 19 April 2019].

[61] Nicolas, B. (2018). Calibrating the depth and color camera. http://nicolas.burrus.name/index.php/Research/KinectCalibration. [Online; accessed 10-January-2018].

[62] PCL (2014). How to use Random Sample Consensus model. http://pointclouds.org/documentation/tutorials/random_sample_consensus.php.

[63] Rapp, I. (2019). Motion Capture Actors: Body Movement Tells The Story. https://www.nycastings.com/motion-capture-actors\-body-movement-tells-the-story/. [Accessed; June, 21 2019].

[64] Cao, Z., Simon, T., Wei, S.-E., & Sheikh, Y. Realtime Multi-Person Pose Estimation. https://github.com/ZheC/Realtime_Multi-Person_Pose_Estimation. [Accessed 23 April 2019].

[65] Geometric (2019). Geometric Transformations. https://pages.mtu.edu/~shene/COURSES/cs3621/NOTES/geometry/geo-tran.html. [Accessed; April, 4 2019].

[66] geeks forgeeks (2019). Linear Regression (Python Implementation). https://www.geeksforgeeks.org/linear-regression\-python-implementation/. [Accessed; April, 4 2019].

[67] Linear (2019). Linear Regression. https://machinelearningcoban.com/2016/12/28/linearregression/. [Accessed; April, 4 2019].

About Authors

TUONG-THANH NGUYEN received B.E. degree from Hanoi University Science and Technology in 2002 in Electronics and Telecommunications; He received M.E. degree in Electronic Engineering, University of Transport and Communications. He is now PhD student in Electronic Engineering, Hanoi University of Science and Technology. Currently, he is working at the Faculty of Engineering and Technology, Quy Nhon University. His research interests include computer vision; image processing 2-D, 3-D machine leaning, deep learning.

VAN-HUNG LE received M.Sc. degree at Faculty Information Technology- Hanoi National University of Education (2013). He received PhD degree at International Research Institute MICA HUSTCNRS/UMI - 2954 - INP Grenoble (2018). Currently, he is a lecture of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; machine leaning, deep learning.

DUONG DUY LONG is a fourth-year student of electrical engineering, School of Electrical Engineering, Hanoi University of Science and Technology. His research interests include computer vision; image processing, machine leaning, deep learning.

THANH-CONG PHAM received M.Sc. degree at Electronics and Telecommunications, Hanoi University of Science and Technology in 1998. He received PhD degree at Electronics and Telecommunications, Turin Polytechnic University, Italy in 2010. Currently, he is a lecturer of institute of Electronics and Telecommunications, Hanoi University of Science and Technology. His research interests include Super high frequency technology, Antennas, Telecommunication systems.

DUNG LE received M.Sc. degree at Electronics and Telecommunications, Hanoi University of Science and Technology in 1998. He received PhD degree at Electronics and Telecommunications, Shibaura Institute of Technology, Japan in 2009. Currently, he is a lecturer of institute of Electronics and Telecommunications, Hanoi University of Science and Technology. His research interests include 2D, 3D and video image processing, pattern recognition with neural network, Human-robot intelligent communication, Design on FPGA and DSP.

A Appendix: Code

The source codes of *"3-D_COCO_method"* and *"3-D_Human3.6_Method"* methods are presented in the folder *"Lifting-from-the-Deep-release-master"*. In this folder, to load the trained model, the defined parameters, we use the source code in the *"demo2.py"* file as follows:

```
SAVED_SESSIONS_DIR = PROJECT_PATH + '/data/saved_sessions'
SESSION_PATH = SAVED_SESSIONS_DIR + '/init_session/init'
PROB_MODEL_PATH = SAVED_SESSIONS_DIR + '/prob_model/prob_model_params.mat'
```

To load the images for 3-D human pose estimation, we assign the path as follows:

```
frame_dir= "/home/hunglv/Lifting-from-the-Deep-release-master/data/images/"
```

To select the format of the image files in the video and load the image files in a folder for estimating 3-D human pose in the 3-D space, we use the source code as follows:

```
frame_paths=glob.iglob(os.path.join(frame_dir, "*.png"))
while i <number_frame:
frame_path=frame_paths[i]
if not os.path.isfile(frame_path):
i=i+1
continue
frame, ext = os.path.splitext(os.path.basename(frame_path))
print ('Processing :d/:d :s...'.format(i,number_frame,frame))
img_path=frame_dir + frame +'.png'
print (img_path)
image = cv2.imread(img_path)
```

To estimate the 3D human pose of the person in the image, we use the source code in the "demo2.py" file as follows:

```
pose_estimator_convert3D = PoseEstimator_convert3D(image_size, SESSION_PATH, PROB_MODEL_PATH)
```

Therein, the function *"PoseEstimator_convert3D"* is presented in *"Load_convert_data.py"* at path *"Lifting-from-the-Deep-release-master/packages/lifting"*. To load 2-D human pose estimation results of using Open pose is the input of 3D human pose estimation as *"3-D_COCO_Method"* is presented in *"Load_convert_data.py"* file as follows:

```
estimated_2d_pose = self.read_openpose_2D(duongdan)
visibility=[[True,True,True,True,True,True,True,True,True,True,True,True,True,True]]
visibility=np.asarray(visibility)
```

With "3-D_Human3.6_Method" method using 2-D human pose estimation results with CPM trained on Human 3.6m dataset. This result includes 14 estimated key points. 2-D human pose estimation function is presented in "Load_convert_data.py" file as follows:

```
pred_2d_pose, pred_likelihood = sess.run([self.pred_2d_pose,self.likelihoods],feed_dict)
estimated_2d_pose, visibility = utils.detect_parts_from_likelihoods(pred_2d_pose,centers,pred_likelihood)
```

The function for drawing the estimated 3D skeleton is based on the "3-D_COCO_Method " and *"3-D_Human3.6_Method"* methods shown in the *"draw.py"* file in the *"Lifting-from-the-Deep-release-master/packages/lifting/utils"* path.

The source code for method *"3-D_ VNECT_ Method"* is shown in the *"VNect-tensorflow-master"* folder. The description of the entire source code for this method is shown in the *"README.md"* file.

The results of *"3-D_ COCO_ Method"* method on the VNMA dataset are shown in the *"Result_ outdata_ Human3._ Input_ COCO"* folder. The results of *"3-D_ Human 3.6_ Method"* method on the VNMA dataset are shown in the *"Result_ ourdata_ human3.6_ liffting"* folder.

The results of *"3-D_ VNECT_ Method"* method on the VNMA dataset are shown in the *"Result_ ourdataset_ VNect"* folder.

B appendix: Dataset

The VNMA dataset includes 24 videos and store in the "Data_24_video" folder, where each video includes the color images, depth images, point cloud data in the 3-D space of each frame.

In order to synchronize the coordinate system of the estimated 3D human pose and the ground truth data, we have built the source code to rotate and translate the estimated 3-D human pose data to the same coordinate system with the ground truth data by *"calculate_ coco.m"* and *"calculate_ matrix_ 14.m"* and *"estimateCoord_ 14.m"* files in the *"rotated_ translated_ 14_ points"* folder.

