3D Human Pose Estimation in Vietnamese Traditional Martial Art Videos

Tuong Thanh Nguyen 1,2, Van-Hung Le∗,3, Duy-Long Duong1, Thanh-Cong Pham1, Dung Le1

1Hanoi University of Science and Technology
2Quy Nhon University
3Tan 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 recent 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 finally 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 im-
ages, 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,

In particular, Microsoft (MS) Kinect sensor
version 1 (v1) is a common and cheap sensor
that can collect information such as color im-
ages, depth images, skeleton and acceleration

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 cap-
tured MS Kinect sensor v1; (3) We propose mea-
sures to evaluate and publish the dataset of Viet-
namese'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 liter-
ature 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. ).

2. Related Works

3-D human pose estimation is often using most
color computer vision techniques. These studies
can be based on a single image or a sequence of
images. The human poses and actions estima-
tion 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 estima-
tion 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) (stud-
ies 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.

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

<table>
<thead>
<tr>
<th>Method</th>
<th>Results of Mean Per Joint Position Error (MPJPE) (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pavllo et al. [12]</td>
<td>Protocol 1: 51.8</td>
</tr>
<tr>
<td></td>
<td>Protocol 2: 40.0</td>
</tr>
<tr>
<td>Liu et al. [26]</td>
<td>61.1</td>
</tr>
<tr>
<td>Nibali et al. [18]</td>
<td>57.0</td>
</tr>
<tr>
<td>Veges et al. [27]</td>
<td>Protocol #1: 61.1</td>
</tr>
<tr>
<td>Wang et al. [28]</td>
<td>Protocol #1: 63.6</td>
</tr>
<tr>
<td>Martinez et al. [29]</td>
<td>protocol #1: 45.5</td>
</tr>
<tr>
<td>Pavlakos et al. [30]</td>
<td>51.9</td>
</tr>
<tr>
<td>Wang et al. [13]</td>
<td>Protocol #1: 40.8</td>
</tr>
<tr>
<td>Hossain et al. [15]</td>
<td>Protocol #1: 39.2</td>
</tr>
<tr>
<td>Li et al. [31]</td>
<td>Protocol #1: 52.7</td>
</tr>
<tr>
<td></td>
<td>Protocol #2: 42.6</td>
</tr>
<tr>
<td>Karim et al. [14]</td>
<td>Protocol 1: 49.9</td>
</tr>
<tr>
<td>Fang et al. [32]</td>
<td>Protocol #1: 60.4</td>
</tr>
<tr>
<td></td>
<td>Protocol #2: 45.7</td>
</tr>
<tr>
<td></td>
<td>Protocol #3: 72.8</td>
</tr>
<tr>
<td>Tekin et al. [33]</td>
<td>50.12</td>
</tr>
<tr>
<td>Omran et al. [34]</td>
<td>59.9</td>
</tr>
<tr>
<td>Pavllo et al. [35]</td>
<td>36</td>
</tr>
<tr>
<td>Bastian et al. [17]</td>
<td>Protocol #1: 50.9</td>
</tr>
<tr>
<td>Kocabas et al. [16]</td>
<td>51.83</td>
</tr>
<tr>
<td>Rhodin et al. [7]</td>
<td>131.7</td>
</tr>
<tr>
<td>Mehta et al. [36]</td>
<td>ResNet 100: 82.5</td>
</tr>
<tr>
<td></td>
<td>ResNet 50: 80.5</td>
</tr>
<tr>
<td>Tome et al. [37]</td>
<td>Protocol #1: 88.39</td>
</tr>
<tr>
<td></td>
<td>Protocol #2: 70.4</td>
</tr>
<tr>
<td></td>
<td>Protocol #3: 79.6</td>
</tr>
</tbody>
</table>
Tab. 2: Survey: 3-D human pose estimation from a single image.

<table>
<thead>
<tr>
<th>Year</th>
<th>Main Author/Reference</th>
<th>2-D pose library</th>
<th>Method Highlights</th>
<th>Evaluation dataset</th>
<th>Evaluation metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>2010</td>
<td>Pavllo et al. [14]</td>
<td>Yes</td>
<td>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 Adam and train for 80 epochs in Human3.6m dataset.</td>
<td>Human3.6m</td>
<td>HumanEval-I</td>
</tr>
<tr>
<td>2019</td>
<td>Liu et al. [20]</td>
<td>No</td>
<td>The feature matching network, the stacked-hourglass CNN is adopted to learn the convolutional features for the RGB image. The feature maps to perceive the geometrical long short-term dependency among different hand (or body) parts using the designed Graphical ConvLSTM. 3D human pose estimation: the 2-D heatmap first as an intermediate representation for inferring the final 3-D pose.</td>
<td>Human3.6m</td>
<td>MPI-INF-3DHP</td>
</tr>
<tr>
<td>2019</td>
<td>Nibali et al. [18]</td>
<td>No</td>
<td>To 3D human pose estimation, coordinates predicted by the model are in the same xy coordinate space as the input, making it straightforward to construct a single fully convolutional network which takes RGB inputs to xy heatmaps. 3D coordinate prediction which avoids the aforementioned undesirable tradeoffs by predicting 2D marginal heatmaps under a segmented softmax scheme.</td>
<td>MPIII</td>
<td>PK</td>
</tr>
<tr>
<td>2019</td>
<td>Wang et al. [26]</td>
<td>Yes</td>
<td>For 2D pose estimation, authors employ a stacked hourglass network designed to simultaneously exploit 2D joint location confidence maps and 3D image cues for 3D human pose estimation.</td>
<td>Human3.6m</td>
<td>HumanEval-I</td>
</tr>
<tr>
<td>2019</td>
<td>Voges et al. [30]</td>
<td>No</td>
<td>The authors adopt the state-of-the-art multi-person pose detector OpenPose on the depth image; the 2D-to-3D component is called 3D PoseNet.</td>
<td>MPJII</td>
<td>L5P</td>
</tr>
<tr>
<td>2019</td>
<td>Wang et al. [13]</td>
<td>Yes</td>
<td>The authors adopt the state-of-the-art stacked hourglass network as the 2D joint estimation; Propose a novel approach to generate multiple feasible hypotheses of the 3D pose from 2D joints.</td>
<td>Human3.6M</td>
<td>MPIII</td>
</tr>
<tr>
<td>2019</td>
<td>Li et al. [34]</td>
<td>No</td>
<td>2D pose is determined with an off-the-shelf component and then the 3D position is predicted from the 2D skeleton. 3D pose estimation: using the Adam optimizer with a learning rate of 0.001 and an exponential decay with a rate of 0.95. The batch size was set to 256. The training ran for 100 epochs.</td>
<td>Human3.6m</td>
<td>HumanEval-I</td>
</tr>
<tr>
<td>2018</td>
<td>Voges et al. [27]</td>
<td>Yes</td>
<td>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.</td>
<td>AIChallenger</td>
<td>UP-2D</td>
</tr>
<tr>
<td>2018</td>
<td>Sun et al. [40]</td>
<td>Yes</td>
<td>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.</td>
<td>COCO</td>
<td>MPIII</td>
</tr>
<tr>
<td>2018</td>
<td>Pang et al. [32]</td>
<td>Yes</td>
<td>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 captures both powerful encoding capabilities of deep neural networks and high-level dependencies and relations of human body.</td>
<td>Human3.6m</td>
<td>HumanEval-I</td>
</tr>
<tr>
<td>2018</td>
<td>Omran et al. [54]</td>
<td>No</td>
<td>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.</td>
<td>Human3.6m</td>
<td>HumanEval-I</td>
</tr>
<tr>
<td>2018</td>
<td>Pavllo et al. [30]</td>
<td>No</td>
<td>The authors propose a viewpoint invariant metrics for 3-D human pose estimation from a single depth image. To achieve this, our discriminative model embeds local features into a learned viewpoint invariant feature space.</td>
<td>MUPO-T3-3D</td>
<td>MPIII</td>
</tr>
</tbody>
</table>

© 2019 Journal of Advanced Engineering and Computation (JAEC)
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 $L_c$ 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 $S_j \in \mathbb{R}^{w \times h}; j \in \{1, 2, ..., J\}$. 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" [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 convolutional 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

<table>
<thead>
<tr>
<th>Year</th>
<th>Author/Reference</th>
<th>Method Highlights</th>
<th>Evaluation dataset</th>
<th>Evaluation matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>2019</td>
<td>Karim et al. [14]</td>
<td>No</td>
<td>Human3.6m CMU Panoptic</td>
<td>MPJPE</td>
</tr>
<tr>
<td>2019</td>
<td>Bastian et al. [17]</td>
<td>Yes</td>
<td>Human3.6m MPI-INF-3DHP LSP</td>
<td>MPJPE</td>
</tr>
<tr>
<td>2019</td>
<td>Ko cabas et al. [16]</td>
<td>Yes</td>
<td>Human3.6m MPI-INF-3DHP</td>
<td>MPJPE</td>
</tr>
<tr>
<td>2018</td>
<td>Rhodin et al. [7]</td>
<td>Yes</td>
<td>Human3.6m</td>
<td>MPJPE</td>
</tr>
<tr>
<td>2018</td>
<td>Hossain et al. [15]</td>
<td>Yes</td>
<td>Human3.6m</td>
<td>MPJPE</td>
</tr>
</tbody>
</table>
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:

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

- The second method is called "\texttt{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 "\texttt{3-D\_VNECT\_Method}": 2-D, 3-D human pose estimation using the VNect in study of Mehta et al. [36].

The method of Tomè 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 \(\mu\), a set of \(J\) orthonormal basis matrices \(e\) and noise variance \(\sigma\), along with each per sample rotations \(R_i\) and basis coefficients \(a_i\).
to minimize the following estimate as Eq. 1.

\[
\arg \min_{R_i, \mu, a, e, \sigma} \sum_{i=1}^{N} \left( \| P_i - R_i(\mu + a_i e) \|_2^2 + \sum_{j=1}^{J} (a_{i,j} \sigma_j)^2 + \ln \sum_{j=1}^{J} \sigma_j^2 \right)
\]

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 Thompson 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
3-D Comparative Study

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:

\[ H_m = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \]  

(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:

\[ H_m = \begin{bmatrix} 594.214 & 0 & 339.307 \\ 0 & 591.040 & 242.739 \\ 0 & 0 & 1 \end{bmatrix} \]  

(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.

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 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 × height: 368 × channel: 3); \( batch\text{Size} = 16; stacks = 4; \) 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 trained a 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.

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

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

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_Metho d") 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_Metho d") is 21 key points.

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

\[
\begin{pmatrix}
x'
y'
z'
1
\end{pmatrix} =
\begin{pmatrix}
R_{11} & R_{12} & R_{13} & T_1 \\
R_{21} & R_{22} & R_{23} & T_2 \\
R_{31} & R_{32} & R_{33} & T_3 \\
0 & 0 & 0 & 1
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z \\
1
\end{pmatrix}
\]

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).

\[
\begin{align*}
x' &= R_{11}x + R_{12}y + R_{13}z + T_1 \\
y' &= R_{21}x + R_{22}y + R_{23}z + T_2 \\
z' &= R_{31}x + R_{32}y + R_{33}z + T_3
\end{align*}
\]

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).
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).

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

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

$$
\begin{align*}
X' &= \begin{bmatrix}
x'_1 \\
x'_2 \\
. \\
x'_n
\end{bmatrix} \\
Y' &= \begin{bmatrix}
y'_1 \\
y'_2 \\
. \\
y'_n
\end{bmatrix} \\
Z' &= \begin{bmatrix}
z'_1 \\
z'_2 \\
. \\
z'_n
\end{bmatrix}
\end{align*}
$$

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).

$$
\begin{align*}
X' &= M\theta_1 \\
Y' &= M\theta_2 \\
Z' &= M\theta_3
\end{align*}
$$

The entire source of the rotation and translation is stored in the path: [https://drive.google.com/file/d/1dIHgal63Tcg0-6_hnTJ1sEdFh8qukNOsE/view?usp=sharing](https://drive.google.com/file/d/1dIHgal63Tcg0-6_hnTJ1sEdFh8qukNOsE/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.

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.

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

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 individuals’ 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 Human3.6M 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


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

<table>
<thead>
<tr>
<th>#Video</th>
<th>3-D_COCO_Method</th>
<th>MPJPE (mm)</th>
<th>3-D_HUMAN3.6_Method</th>
<th>3-D_VNECT_Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>114.0716</td>
<td>114.0716</td>
<td>228.8319</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>107.5917</td>
<td>111.025</td>
<td>332.8037</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>88.5689</td>
<td>91.536</td>
<td>245.1891</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>78.6414</td>
<td>79.9366</td>
<td>239.818</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>99.0704</td>
<td>101.6098</td>
<td>282.843</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>111.0964</td>
<td>112.0768</td>
<td>292.2822</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>114.7642</td>
<td>118.3664</td>
<td>309.3528</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>285.0776</td>
<td>292.9947</td>
<td>318.6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>90.0766</td>
<td>92.9212</td>
<td>253.3029</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>280.8594</td>
<td>284.8666</td>
<td>294.9349</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>91.2715</td>
<td>91.2715</td>
<td>249.4076</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>219.4037</td>
<td>219.4037</td>
<td>242.6467</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>89.3462</td>
<td>89.3462</td>
<td>267.3336</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>264.4088</td>
<td>262.0707</td>
<td>271.0392</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>85.9806</td>
<td>87.3728</td>
<td>254.4252</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>318.4422</td>
<td>318.4422</td>
<td>343.7987</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>99.5296</td>
<td>101.7892</td>
<td>271.0186</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>308.1409</td>
<td>310.7236</td>
<td>331.4765</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>110.9321</td>
<td>110.9321</td>
<td>320.2984</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>239.3639</td>
<td>241.5342</td>
<td>271.7371</td>
<td></td>
</tr>
<tr>
<td>21</td>
<td>81.9572</td>
<td>81.9572</td>
<td>206.8996</td>
<td></td>
</tr>
<tr>
<td>22</td>
<td>103.5087</td>
<td>105.8891</td>
<td>280.5987</td>
<td></td>
</tr>
<tr>
<td>23</td>
<td>267.6513</td>
<td>292.217</td>
<td>282.1385</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>170.866</td>
<td>173.7285</td>
<td>279.4472</td>
<td></td>
</tr>
</tbody>
</table>


In: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), vol. 11214 LNCS. 765–782.


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.
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.


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


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 lecturer of Tan Trao University. His research interests include Computer vision, RANSAC and RANSAC variation and 3-D object detection, recognition; machine leaning, deep learning.

Duy-Long DUONG 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]]
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_lifting" folder.

The results of "3-D_VNECT_Method" method on the VNMA dataset are shown in the "Result_ourdataset_VNe ct" 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.