Multimedia Tools and Applications
https://doi.org/10.1007/s11042-018-6789-4

Camera localization for a human-pose in 3D space using a single 2D human-pose image with landmarks: a multimedia social network emerging demand

Mo'taz Al-Hami[1] · Rolf Lakaemper[2] · Majdi Rawashdeh[3] · M. Shamim Hossain[4]

Received: 21 May 2018 / Revised: 31 August 2018 / Accepted: 17 October 2018 /
Published online: 13 November 2018

Abstract
Recovering a 3D human-pose in the form of an abstracted skeleton from a 2D image suffers from loss of depth information. Assuming the projected human-pose is represented by a set of 2D landmarks capturing the human-pose limbs, recovering back the original 3D locations is an ill posed problem. To recover a 3D configuration, camera localization in 3D space plays a major role, an inaccurate camera localization might mislead the recovery process. In this paper, we propose a 3D camera localization model using only human-pose appearance in a 2D image (i.e., the set of 2D landmarks). We apply a supervised multi-class logistic regression to assign the camera location in 3D space. In the learning process, we assume a set of predefined labeled camera locations. The features we train consist of relative length of limbs and 2D shape context. The goal is to build a relation between these projected landmarks and the camera location in 3D space. This kind of analysis allows us to reconstruct 3D human-poses based on the 2D projection only without any predefined camera parameters. Also, makes real-time multimedia exchange more reliable specially for human-pose related tasks. We test our model on a set of real images showing a variety of camera locations.

Keywords Human-pose · Projection · Camera localization · Multimedia · Logistic regression · 2D shape context · 3D reconstruction · Rotation matrix · Translation · Extrinsic camera · Intrinsic camera · Principal component analysis · Features · Projection error

1 Introduction

Nowadays, there are many multimedia sensors capable with a communication technology. These sensors serve as the eyes and ears of our life aspects, resulting in a huge source of collected data. Such a big multimedia data requires extensive processing and analyzing to reveal the intended knowledge and information. Microsoft Kinect is one of the most

✉ Mo'taz Al-Hami
motaz@hu.edu.jo

Extended author information available on the last page of the article.

popular sensors that has thrived in many multimedia computing applications [35]. With Kinect sensor a new generation of games started to appear and spread rapidly. The Kinect technology power comes through its ability to allow people to play games while using their body as a mean of interaction with the game itself. The Kinect sensor utilizes the depth information alongside with the RGB data input to detect human body joints (i.e., 3D human-pose) and ultimately decides how to interpret the body movements. Character control and navigation in a virtual environment in real-time allows the user to have a full freedom to interact with the virtual environment [21, 22]. Figure 1 shows several data format we can get from Kinect sensor.

When dealing with monocular RGB images the situation for reconstructing 3D human-pose is completely different, the depth information is completely lost. The reconstruction process in this scenario depends on the 2D human-pose landmarks that appear in the monocular RGB image. Assuming a human-pose is identified by a set of landmarks identifying the human-pose joints locations, any constructed 3D model depends directly on these landmarks. Many applications focus on this type of research like video browsing and indexing [36]. Ability to characterize and recognize human activities is an important step towards smart multimedia. Such enhancement leads to an automatic human-pose recognition and modeling in 3D space which makes classification and recognition process more attractive. In order to step forward towards such goal, the relation between 2D human-pose and its related 3D human-pose needs to be resolved logically and explanatory. The core demand is to reconstruct 3D human-poses from 2D human-poses correctly.

The projection of objects from 3D space into 2D is strongly affected by the camera location (i.e., view angle). For example, in Fig. 2 there is a rigid shape in 3D space. The shape has been projected onto different 2D planes using different camera locations. Placing the camera on the x, y, or z axis and projecting along these axes respectively leads to significantly different images, here ‘M’, ‘H’, ‘T’. From this example, if we assume an object can be represented by a set of 3D locations, then we can conclude that there are three main elements in the projection process: (1) the spatial location of the landmarks in the 3D space. (2) the spatial location of the projected landmarks on the 2D surface, and (3) the camera parameters (i.e., camera localization in the 3D space).

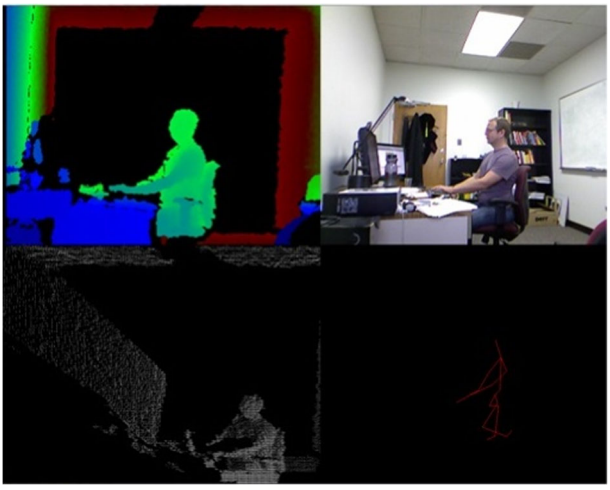


Fig. 1 **Up:(left)** Kinect depth map, **(right)** RGB image. **Down: (left)** depth point-cloud, **(right)** skeleton [17]

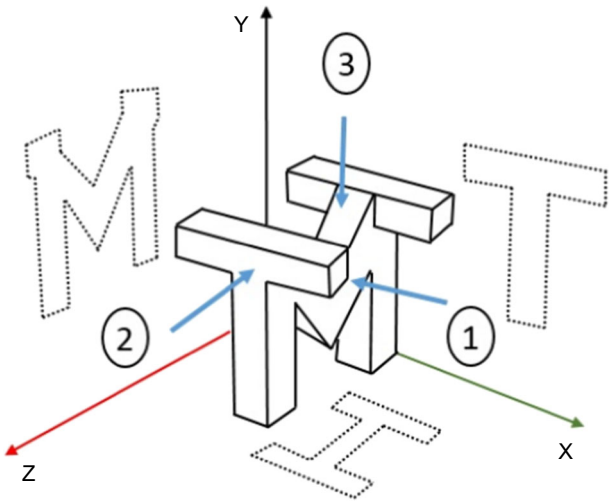


Fig. 2 3D object projection process using different viewing angles

Usually, the projection process is a straight forward process, since the object in 3D space is well recognized, specially in terms of depth recognition, and the ability to decide the camera parameter as needed. The problem becomes much more difficult when trying the reverse the process (i.e., reconstruct a 3D shape from a projected points on a 2D surface). In the reconstruction process, the depth information and the camera parameters are completely lost. Ability to reconstruct the original 3D object under such conditions is impossible, however the competition is to recover a representative object which can describe the original one as much as possible if additional constraints are known.

If the 3D object that we want to reconstruct is a deformable object, like the human skeleton, then the problem becomes even more difficult. The projected human-pose might satisfy different conformation in 3D space (i.e., the locations of these 3D conformations can project to the same projected human-pose). In addition to that, the camera localization also affects the projections based on its location in the 3D space. Localizing the camera accurately can enhance the 3D reconstruction, which means a better 3D model could be achieved. Such enhancement would improve the 3D human-pose reconstruction in terms of plausible human-pose. The importance of human-pose reconstruction problem is its ability to enhance the 3D modeling recognition and awareness using 2D images. Such awareness has a strong relationship to humanoid robot [5]. Humanoid robots are increasingly adapting to physically mimic human actions and poses, which require an understanding of human-poses and how they can be captured with relation to the robot's joint movements. A reliable 3D human-pose reconstruction can be admitted as a major source of understanding the human-pose.

Nowadays, the main approach for camera parameter estimation in the field of human-pose reconstruction is to use the Orthogonal Procrustes Transformation (OPT) [28]. In OPT, the goal is to find the orthogonal transformation which maps the limbs locations in 3D space to their corresponding landmarks in the projected plane. In this paper, we propose a new approach, which uses only the human-pose appearance in the projected image (i.e., the projected landmarks). We develop a new feature that represents a human-pose using the projected limbs relative lengths mixed with 2D shape context for that human-pose. We apply a multi-class logistic regression approach to localize the camera in the 3D space. Assuming

we have predefined camera locations, the goal is to learn these locations based only on the created feature structure.

Our contribution in this paper can be summarized as follows: First, we propose a new camera localization approach, which only requires projected landmarks. Second, we show how this approach can improve the 3D reconstruction process by applying it on the work in [24]. The paper is organized as follows. Related work is described in Section 2. Problem definition is discussed in Section 3. Section 4 describes camera localization. In Section 5, we describe the 2D human-pose feature descriptor. Dataset preparation is discussed in Section 6. Learning process in the multi-class logistic regression is reviewed in Section 7, and the evaluation of used approach is discussed in Section 8. Finally, conclusion remarks appear in Section 9.

2 Related works

2.1 Smart multimedia

Multimedia with artificial intelligence support play a core role in many recent applications including surveillance of human-pose and conversational groups [32], virtual reality [29], video indexing [9], and protein 3D shape prediction [2]. The ability to extract useful knowledge form videos and images and at the same time take the right decision at the right time added a lot of benefits. The Kinect sensor added a new dimension to the captured data which is the depth dimension. With this enhancement we are able to mix the RGB input video frames with their related depth information. KinectTCP [17] is a TCP/IP server that offers all video, depth and skeleton services of Microsoft's Win7 Kinect SDK, independent from specific programming languages. The KinectServer offers raw RGB video data in all available resolutions, depth and player ID data in all resolutions, and the entire set of skeleton data (joints, floor plane etc.) and audio data. For convenience, the server can additionally send depth data as XYZ point cloud, i.e. depth data as point cloud in a 3D volume, as well as depth data (both, depth and depth-XYZ) with corresponding RGB color information. Video streaming over wireless LAN with different resolution [4] allows different clients with different channel diversity to receive different video resolutions. In this work we focus on monocular RGB images having human-poses inside.

2.2 2D human-pose estimation

There are many studies dealing directly with human-pose. One direction of these studies focuses on the 2D human-pose estimation or action recognition [11, 13, 23, 25]. The goal of these studies is to extract the human-pose from images and apply further analysis, like identifying the performed actions in these poses. Deformable models have been utilized extensively in estimating human-pose. Many works like [13, 34] apply deformable models for the human-pose estimation. Such approach focuses on the human-pose parts appearance and localization in an image. Using a training set, the deformable model builds a learning system which is invariant to a human-pose location or appearance. For unseen images, the deformable model localizes poses without prior knowledge about these images and their backgrounds as well.

The work [27], focuses on a multimodal approach with different modes capturing the human-pose parts. The approach uses these modes as a rich structure discriminative using linear classifiers. The strength of this approach is its simplicity compared to the deformable

model. The work in [14] applies shape matching to detect human-poses in the real-time. Using a predefined database of shapes, the approach computes a distance measure to find the closest shape to the target one. A hidden Markov model, mixed with pictorial structure has been applied in [18] to track a human-pose model in videos. In [19] the work benefits from the existing symmetry in the human-pose (i.e., left/right arms, and left/right legs) and fits a graph model to represent the human-pose.

2.3 3D human-pose reconstruction

The other direction goes one forward step by trying to build a 3D model for the human-pose based on 2D images [1, 6, 12, 24, 33]. The benefit of such 3D modeling could increase the interaction of humans and humanoid robots [16]. For example, performing human mimicking tasks, such as walking, grasping, standing and sitting on objects [5]. Works in [3, 7, 8] focus on building approximate 3D human-pose based on a set of related images. This approximate model gives a generic description of the approximate 3D human-pose.

Talking about 3D human-pose reconstruction from 2D images means there are many elements we have to take into account: First, the reconstructed human-pose, the projected human-pose in the 2D image, and the camera localization. Since the camera localization has a major effect on the reconstruction process, localizing camera correctly is a major task. The work in [28] has showed a generalized solution to the orthogonal Procrustes problem. In this solution, there is a solved estimate to find the orthogonal transformation matrix which maps between two matrices.

The work in [30] applies a scaled orthographic projection model. Assuming a fixed length limbs model, a 3D human-pose is constructed by accounting foreshortening the pose limbs in the 2D image. Applying some triangulation equations on the extended rays from the landmarks in the 2D image, the 3D coordinate can be estimated. The approach is straight forward, however there is a depth direction ambiguity, and some parameters are assumed to be fixed.

2.4 Camera localization in 3D space

Many works [1, 24, 33] have applied OPT to localize the camera. The work in [24] uses a projected matching pursuit algorithm. The goal is to minimize the error between landmarks in a projected 3D human-pose and the ones that appear in a 2D image. At the same time, the approach estimates the camera parameters that are required for projection process. In [33] the work performs similar approach however, it applies the Alternating Direction Method (ADM) to solve the optimization. A recent work in [1] re-applies a modified projected matching pursuit algorithm, where the new algorithm forces limbs rotation limits in the optimization equation. One common thing between these works is the camera localization approach which is orthogonal Procrustes [28]. Unlike these approaches, our work focuses on localizing camera based on the appearance of human-pose in a 2D image.

3 Problem definition

Let the human-pose model $\mathcal{M}$ in a 2D image consists of a finite collection set of landmarks $\mathcal{L} \in \mathbb{R}^{2(|\mathcal{L}|)\times 1} = [\hat{l}_1, \hat{l}_2, ..., \hat{l}_{N+1}]^T$, where $|\mathcal{L}| = N + 1$. Each $\hat{l}_i \in \mathcal{L}$ is represented by $(u, v)^T$. These landmarks are organized according to a kinematic hierarchy model that satisfies a human-pose model. Camera localization is identified by a rotation matrix $R \in \mathbb{R}^{3\times 3}$

capturing camera orientation with respect to a reference location, and translation vector $t \in \mathbb{R}^{2\times1}$. The translation vector t is the cartesian position of the camera relative to the reference coordinate. Assuming the camera cartesian position is located on a virtual sphere surface, where the original 3D human-pose (unknown at this point) is centralized in this sphere origin. Given a predefined set $C \in \mathbb{R}^{3\times3} = [c_1, c_2, ..., c_k]$ where $|C| = k$ of camera locations on a predefined virtual sphere. For a new 2D image having a human-pose with landmarks, the task is to find the best camera location among the set C which describes the camera location in the space. Mathematically, the problem can be treated as a multi-class classification problem, where the classes are the camera locations.

4 Camera localization

Many approaches treat the camera localization for the 3D human-pose reconstruction in different ways. The work in [30] uses a scaled orthographic projection model, where predefined relative lengths of limbs were used. Such an approach assumes that the camera is in a fixed location (i.e. facing the projected landmarks). The 3D human-pose is defined by the set $\mathcal{X} = [l_1, l_2, ..., l_{N+1}]^T$ of locations, where $|\mathcal{X}| = N + 1$. These locations identify the limbs in 3D space. Each location consists of the triple $(x, y, z)^T$. A 3D human-pose limb's landmark location l_i is just a factor of scale of the 2D projection such that:

$$\hat{l}_i = s \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} l_i \tag{1}$$

Where s is the scaling factor parameter. The reconstruction process seems to be simple (see Fig. 3), however the problem happens in estimating the reconstructed limb's depth direction, which produces many possible solutions. In addition, the scaled factor s value affects the reconstructed 3D human-pose plausibility. Recent approaches in [24, 33] use camera parameters in 3D human-pose reconstruction, and the approach for camera reconstruction utilizes two sets of data points. A human-pose $\mathcal{X}$ in 3D space is projected into a 2D image such that:

$$\mathcal{L} = (I_{|N+1|\times|N+1|} \otimes \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R)\mathcal{X} + t \otimes 1_{2(N+1)\times1} \tag{2}$$

Where I is the identity matrix, $\otimes$ is the Kronecker product. s_x and s_y are the scaling factors in x, y dimensions. The used approach for estimating the camera localization (i.e., rotation matrix) depends on a set of points in 3D space like $\mathcal{X}$ and their related projections in 2D space like $\mathcal{L}$. In Orthogonal Procrustes Transform (OPT) [28]:

$$\mathcal{T} = \mathcal{L}\mathcal{X}(\mathcal{X}\mathcal{X}^{\mathcal{T}})^{-1} \tag{3}$$

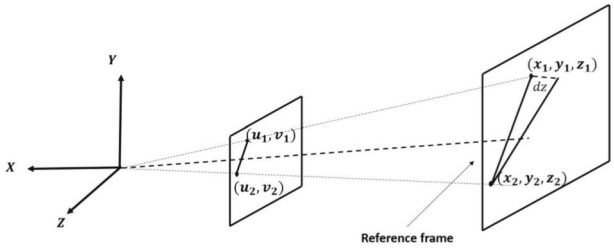


Fig. 3 3D reconstruction using scaled orthographic projection model [30]

Where $\mathcal{T}$ is the required transform to map the $\mathcal{X}$ points into the $\mathcal{L}$ points. The equation is solved using singular value decomposition approach $\mathbf{UDV}^T$, where the rotation matrix $R = \mathbf{UV}^T$.

In this paper we follow a completely different approach for estimating the camera localization (R). The approach focuses on learning the relation between the projected landmarks $\mathcal{L}$ and camera location among the set of cameras C. Thus, the direction of the camera is based only on the 2D human-pose projection in an image. We use a multi-class logistic regression to reconstruct the learning hypothesis for each camera location in the set C, and we apply the learned model to assign the camera location for new 2D projections.

To build a such model, we assume a 3D human-pose is localized at the center of a virtual sphere (see Fig. 4). The goal of this virtual sphere to place the cameras on, so we can link their parameters estimations with the virtual sphere characteristics. Precisely, the camera location can be determined using the triple (r, θ, ϕ). r is the sphere radius $r = \sqrt{(x^2 + y^2 + z^2)}$, θ is the outwarded angle from the optical axis $\theta = \cos^{-1}(\frac{r}{z})$, and ϕ is the rotation angle around the optical axis $\phi = \tan^{-1}(\frac{y}{x})$.

Given a point p on a sphere surface (p_x, p_y, p_z) which represents the camera location, and assuming this camera is directed towards sphere origin $(0, 0, 0)$. The rotation matrix can be reconstructed as follows: The unit vector of z direction $\mathbf{z}_{axis} = -(p^T)/\mathbf{norm}(p^T)$, where $\mathbf{norm}(.)$ is the vector Euclidian norm. Using the cross product $\times$, we can find the x direction such that $\mathbf{x}_{axis} = \mathbf{z}_{axis} \times [0\ 0\ 1]^T$ which makes the x direction located in the xy plane. Finally, the y direction is just applying the cross product between z direction and x

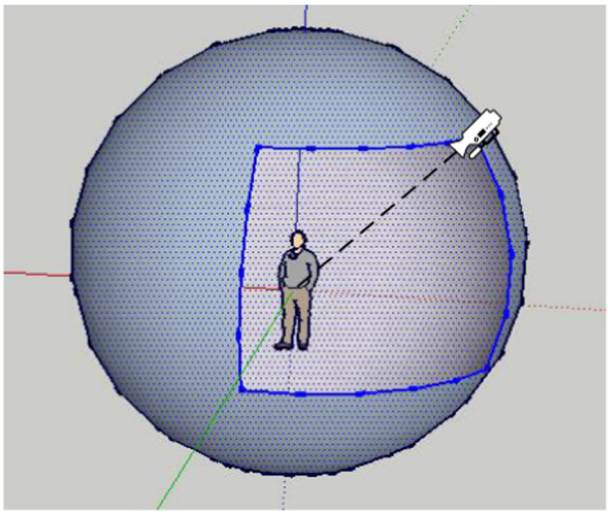


Fig. 4 Camera is located on a sphere and directed towards the sphere origin point. The 3D human-pose is centralized on the sphere origin

direction, so $\mathbf{y}_{axis} = \mathbf{z}_{axis} \times \mathbf{x}_{axis}$. The final extrinsic camera matrix $\mathcal{M}$ can be calculated such that:

$$\mathcal{M}^{-1} = \begin{bmatrix} \mathbf{x}_{axis} & \mathbf{y}_{axis} & \mathbf{z}_{axis} & p^T \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

Inverse is taken is to adjust the appearance of the projected image. The reconstructed rotation matrix R is the first 3×3 section of $\mathcal{M}$. Using such cameral localization system, the projected human-pose is affected by camera location which determines the projected appearance of a human-pose in an 2D image.

The work in [24] applies a projected matching pursuit algorithm to reconstruct a 3D human-pose using a 2D human-pose identified by a set of landmarks. A predefined database (i.e., basis poses) $\mathcal{B} = \{b_1, b_2, ..., b_t\}$, $|\mathcal{B}| = t$, of general 3D human-poses capturing various activities are utilized. The approach tries to approximate a 3D human-pose as a linear combination of a basis subset $\mathcal{B}^*$, selected from the database $\mathcal{B}$. Such approximation applies linear dimensionality reduction (Principal Component Analysis (PCA)) such that:

$$\bar{\mathcal{X}} = \mu + \sum_{i=1}^{|\mathcal{B}^*|} b_i \bar{\omega}_i \tag{5}$$

here $\bar{\mathcal{X}}$ is the reconstructed human-pose landmarks, μ is the mean human-pose in the database $\mathcal{B}$, and $\bar{\omega}$ vector is the associated weights for the basis poses. The chosen approximated human-pose is the one which minimizes the projection error such that:

$$\min \parallel \mathcal{L} - (I_{|N+1|\times|N+1|} \otimes \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R) \bar{\mathcal{X}} - t \otimes 1_{2(N+1)\times 1} \parallel_2 \tag{6}$$

5 2D human-pose feature descriptor

The desired feature descriptor for a 2D human-pose is the one which is able to describe a projected human-pose correctly. Thus, projected relative length of limbs along with 2D shape context have a strong relation to the camera localization in 3D space. Figure 5 shows a scenario of possible camera localization in 3D space. Figure 5a shows how the cameras are localized on a virtual sphere, where all cameras are directed to the same point (i.e., virtual sphere origin). A 3D human-pose is placed in the virtual sphere origin. The projected poses in Fig. 5b are the results from the cameras projections. The lower four poses in Fig. 5b are related to the lower four cameras in Fig. 5a (the same apply for the middle and the upper projections which both are related to the middle and the upper cameras). The limb's length has a direct relation to the camera location. For example, in general the projected lower limbs show larger length than upper limbs if the camera is placed at a lower location.

The human-pose descriptor utilizes relative lengths to partially identify the camera location. In addition to the relative length, 2D shape context is also incorporated into the feature descriptor.

In the relative length estimate, the torso limb is the normalized reference. Next, for the proposed descriptor estimation, we pass the relative length component to a gaussian function. The motivation of using such function is to smooth to relative length change when the

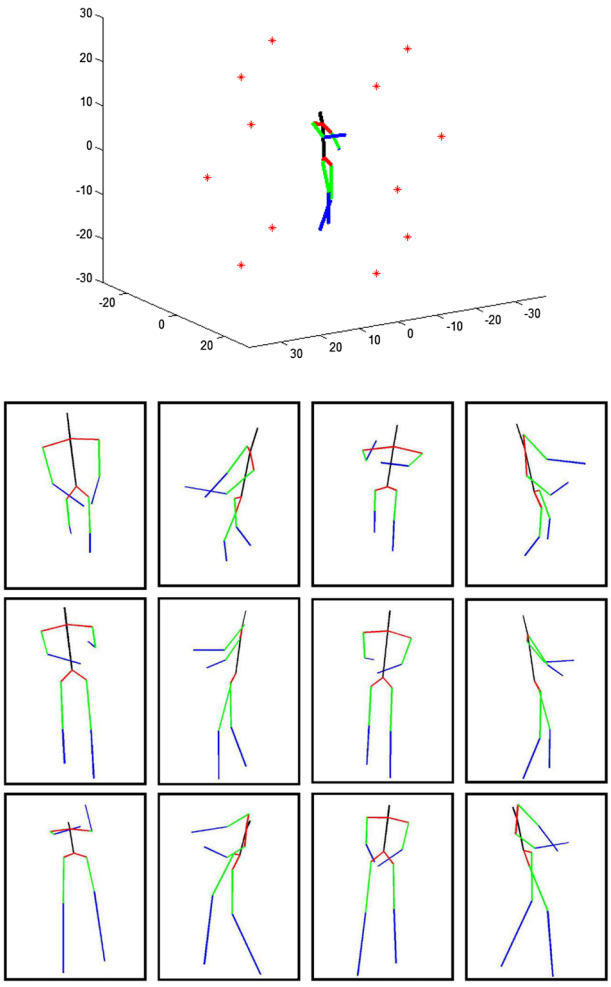


Fig. 5 Cameras localization on the virtual sphere surface. Each camera has different viewing angle, and produces a different projected human-pose

pose faces a rotation process. Figure 6 shows the effect of using direct relative length, and gaussian based relative length. The Gaussian smoothing introduces a parameter σ which is determined by experiment.

Beside relative length of limbs, which is intended to capture camera height, we need also to capture the camera direction. 2D shape context [10] is a well-known approach for shape matching. In 2D shape context a given shape is represented by a set of key points $\mathtt{P} = \{\mathtt{p}_1, \mathtt{p}_2, ..., \mathtt{p}_n\}$ where $\mathtt{p}_i \in \mathbb{R}^2$ describing that shape contour. Each key point $\mathtt{p}_i$ in the set has $n - 1$ descriptor vectors describing that shape (i.e., describing the relation between the point $\mathtt{p}_i$ and the rest of the points). The 2D shape context uses a binning disk with log polar histogram bins to compute the shape context (see Fig. 7b). The built histogram is the

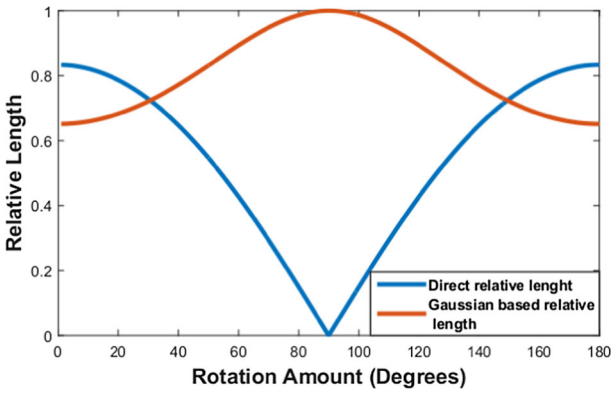
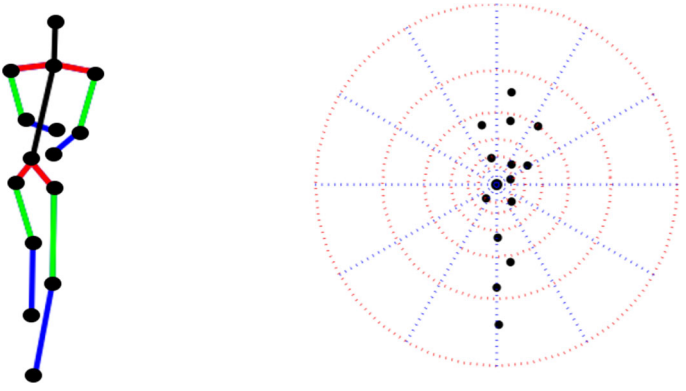


Fig. 6 The effect of using direct relative length and Gaussian based relative length. We place one limb in front of the camera and rotate it gradually to 180 degrees. Assuming the original limb length as the reference length, we estimate the relative length at each rotation process. The Gaussian approach shows a smooth transition when rotation is applied

shape context for the key point $\mathtt{p_i}$. In the human-pose case, the key points are the landmarks $\mathcal{L}$ (Fig. 7a). We estimated the shape context for each pose, using the given landmarks and assuming the torso key point is located at the center of the binning disk. Figure 7b shows how the landmarks are distributed on the binning disk.

Algorithm 1 describes the steps required to create the feature descriptor for a given 2D human-pose. It receives the set of projected 2D human-pose as an input at line 1. In line 2, the limbs lengths are estimated using Euclidean distance, and this is done for each 2D pose separately. 2D shape context is estimated for each pose in line 4. The next step in lines (6 - 7) is to estimate the limbs relative length for each pose, and this done based on the torso limb length for each pose separately. Instead of using relative lengths directly in the feature, we smooth it using a Gaussian function in line 8. Finally, the output feature descriptor for each



(a) Arbitrary Pose (b) Binning disk

Fig. 7 **Left a:** shows 2D projected human-pose. **Right b:** shows the binning disk with the key points (landmarks) distributed on the binning disk

pose includes the smoothed relative length information followed by the 2D shape context for that pose.

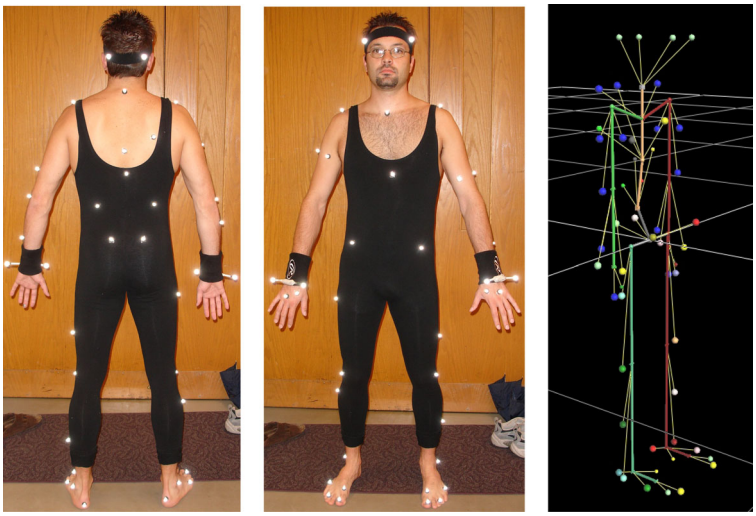
Algorithm 1 2D human-pose feature descriptor creation

1: **Input:** projected human-pose landmarks matrix $w \in \mathbb{R}^{2(|\mathcal{L}|)\times 1}$.
2: calculate the limbs lengths (i.e. Euclidian distance between a limb end points) for all limbs in the human-pose.
3: torso limb is the reference limb.
4: SC = shapeContext(pose).
5: feature = [ ]
6: **for all** limbs ∈ human-pose
7: calculate the relative length **rl** (i.e. limb's length relative to the torso length).
8: $\mathbf{n}_{rl} = \mathbf{e}^{\frac{-\mathrm{rl}^2}{2\sigma^2}}$
9: feature = [feature $\mathbf{n}_{\mathrm{rl}}$]
10: **end for**
11: **Output:** [feature SC]

6 Dataset preparation

In this section, we focus on preparing the training dataset. The goal of this dataset is to describe the relation between the projected 2D human-poses (represented by their related features descriptors) and their associated cameras viewing points (i.e., rotation matrices). In this work, we assume the camera might be located in one of 12 distinct viewing points as shown in Fig. 5. The rotation matrix for each viewing point is described in Table 1. Precisely, if we have a camera capturing a photo for a 3D human-pose then the result is a 2D image having a projected human-pose. The goal from the whole learning process is to understand the hidden relation between a camera viewing angle (orientation) and its resulted projected human-pose. To summarize the required steps for preparing the training dataset: First, we need a set of 3D human-poses. Second, a camera which is located in one of the predefined 12 distinct locations. Third, using the camera and the 3D human-pose we produce a projection of 2D human-pose in an image. The 3D human-poses in the dataset are all placed in the same direction, otherwise the camera location is meaningless.

The 3D human-poses are extracted from the CMU motion capture dataset [15]. It covers many activities including jumping, walking, bending and others. For each activity, a set of markers were placed on a human body identifying the accurate locations of the limbs landmarks (see Fig. 8). 12 calibrated Vicon infrared MX-40 cameras were utilized to capture the 3D motion of the human-body. We extracted only the motion where the pose moves in a fixed direction as we mentioned earlier. In this work, we focus only on landmarks identifying specific limbs. The number of the landmarks is 15 identifying torso, hip, femur, tibia, lower neck, head, clavicle, humerus, and wrist locations. Given the projected landmarks matrix $\mathcal{L} = [\hat{l}_1, \hat{l}_2, ..., \hat{l}_{N+1}]^T \in \mathbb{R}^{2\times|N+1|}$ and assuming we are using a specific number of cameras (12 cameras in this case) in the camera set $C = [c_1, c_2, ..., c_m]$ where $|C| = m$. Each projected human-pose matrix $\mathcal{L}$ is assigned to one camera location from the camera set C. The total number of the original 3D human-poses in the dataset is 3599, and after applying 12 cameras on the virtual sphere surface, we were able to generate 43188



(a) Back View (b) Front View (c) The Vicon Skeleton Template

Fig. 8 Markers set which is used to capture human-pose accurately. In the skeleton template, balls represent markers and the colored segments represent limbs [31]

different 2D projections. These projections are further processes to generate their related features as mentioned earlier in Section 5. The final dataset has the features describing the 2D projections and their related labels (i.e., the camera parameters).

7 Learning process

We use the regularized multi-class logistic regression classifier as the classification model. Given the training set of labeled projected human-pose descriptors $\{\hat{x}_i, \hat{y}_i\}_{i=1}^{K}$ where $\hat{x}_i$ is the feature descriptor for its related projected human-pose as defined in Section 5, and $\hat{y}_i \in \{1, 2, ..., |C|\}$. We train the probabilistic multi-class logistic classifier on each class label identified by camera localization in the set $\{C\}$. Precisely, each projected pose descriptor in the training set is trained using all the camera localization classes labels in order to build the hypothesis:

$$\begin{aligned} \mathrm{h}_{\theta}^{(j)}(\hat{x}_i) &= \mathrm{P}(\hat{y}_i = c_j \backslash \hat{x}_i, \theta_j) \\ \forall i &= (1, 2, ..., K) \\ \forall j &= (1, 2, ..., |C|) \end{aligned} \tag{7}$$

Such training is intended to tune the parameters θ_j which is related to class j where $j \in \{1, 2, ..., |C|\}$, which produces hypothesis parameters able to assign each human-pose descriptor to the right camera class. On a new human-pose descriptor input $\bar{x}_i$, the probability of

having a class label $y = c_j$, given the input $\bar{x}_i$, is modeled according the multi-class logistic regression by:

$$\mathrm{P}(y_i = c_j \backslash \bar{x}_i) = \frac{\exp(\bar{x}_i^T \theta_j)}{\sum_{j=1}^{|C|} \exp(\bar{x}_i^T \theta_j)} \tag{8}$$

The log likelihood $\mathscr{L}$ under this model is:

$$\mathscr{L}(\theta) = \sum_i [\hat{y}_i \log \mathrm{P}(\bar{x}_i) + (1 - \hat{y}_i) \log(1 - \mathrm{P}(\bar{x}_i))] \tag{9}$$

Where θ is the whole model hypothesis. The goal is to maximize the l_2-norm log likelihood function. For the purpose of generalization, the regularized log likelihood would be:

$$\mathscr{L}^{\lambda}(\theta) = \mathscr{L}(\theta) - \lambda \theta^T \theta \tag{10}$$

Where λ is the ridge parameter that controls the shrinkage of θ. The (10) can be reformulated as a minimization problem like:

$$\theta^{\star} = \underset{\theta}{\operatorname{argmin}} - \mathscr{L}(\theta) + \lambda \|\theta\|^2 \tag{11}$$

The optimization problem in (11) is convex [26]. To perform training and solve the equation, we use the Newton method [20].

8 Evaluation

In the evaluation, we show the effect of camera localization parameters on the 3D pose reconstruction process. We focus on showing the performance of our model when localizing camera using only 2D images with landmarks. Also, we show how such model can be integrated with 3D reconstruction model in [24] to reconstruct 3D poses.

8.1 Camera localization

For camera localization (i.e., defining the class of rotation matrix among several classes as shown in Table 1). 12 cameras cover the main directions at three different heights as follows : (1) upper-level where the cameras are in top-down view direction, (2) middle level (i.e., standard), where the cameras face the pose from different directions, and (3) lower-level, where the cameras are closer to bottom-up direction. To cancel the mirror effect, we assume the projected human-poses are all frontal poses. Based on that, any classified camera location in the back side is projected directly to its equivalent location in the frontal side. For example, if the classified camera location is at position with label '07' (i.e., middle-level with back location), then the chosen class label is '05' which is the middle-level frontal direction.

In Fig. 9, the classified camera location is at position at position '04' which means the camera is at the lower-level with the right side. The lower limbs lengths in the Fig. 9 is relatively larger than the upper limbs lengths, and this makes the learned model to use the lower-level camera height. Also, the human-pose is in the side position, and the added 2D

Table 1 Specifications of the used cameras system (i.e., camera localization) in the experiments. 12 cameras are placed on the virtual sphere surface, and they are organized at three different heights. Each camera is identified by a fixed rotation matrix and a fixed class label

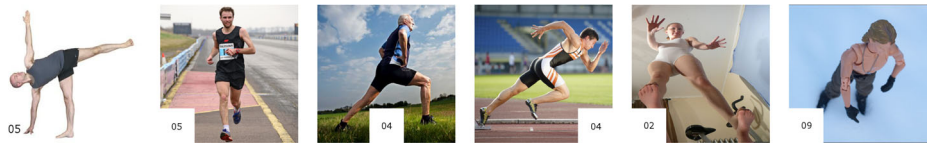
		Class Label and camera parameters	
	Camera	Label	Rotation matrix
Upper Level	Front	09	[0 -1 0 ; -0.7071 0 -0.7071 ; 0.7071 0 -0.7071]
	Left	10	[1 0 0 ; 0 -0.7071 -0.7071 ; 0 0.7071 -0.7071]
	Back	11	[0 1 0 ; 0.7071 0 -0.7071 ; -0.7071 0 -0.7071]
	Right	12	[-1 0 0 ; 0 0.7071 -0.7071 ; 0 -0.7071 -0.7071]
Middle Level	Front	05	[0 -1 0 ; 0 0 -1 ; 1 0 0]
	Left	06	[1 0 0 ; 0 0 -1 ; 0 1 0]
	Back	07	[0 1 0 ; 0 0 -1 ; -1 0 0]
	Right	08	[-1 0 0 ; 0 0 -1 ; 0 -1 0]
Lower Level	Front	01	[0 -1 0 ; 0.7071 0 -0.7071 ; 0.7071 0 0.7071]
	Left	02	[1 0 0 ; 0 0.7071 -0.7071 ; 0 0.7071 0.7071]
	Back	03	[0 1 0 ; -0.7071 0 -0.7071 ; -0.7071 0 0.7071]
	Right	04	[-1 0 0 ; 0 -0.7071 -0.7071 ; 0 -0.7071 0.7071]

shape context to the used feature allowed the learned model to decide to choose the side position.

Figure 10 shows the classified camera location for a set of newly unseen 2D images with labeled landmarks. The resulted labels are able to capture the direction of the camera successfully. In some cases, like Fig. 10k, the label is '02' but it seems it closer to label 03. Such little mistakes can be reduced by enhancing the training set to cover more activities. The model is also able to detect the side views in Figs. 10c, j, i. Figure 10i shows a close look, and the differences between limbs lengths appear clearly, which means camera focal length adjustment can enhance the performance of the localization model.

Fig. 9 The learned hypothesis during the training phase classified the camera location at position '04' which means the camera is at the lower-level with the right side

(a) Label: 02 (b) Label: 05 (c) Label: 02 (d) Label: 05 (e) Label: 06 (f) Label: 05



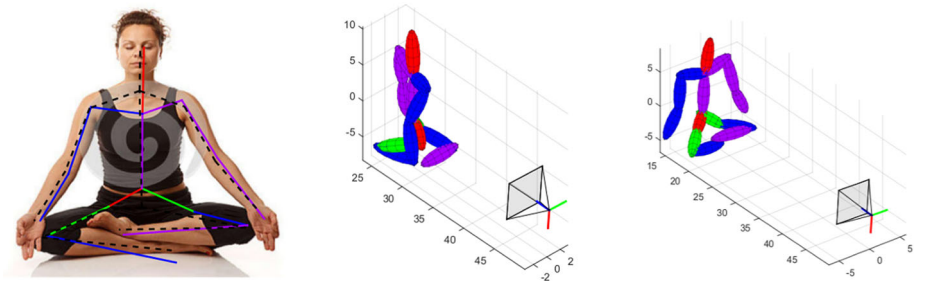
(g) Label: 05 (h) Label: 05 (i) Label: 04 (j) Label: 04 (k) Label: 02 (l) Label: 09

Fig. 10 Different real images having different camera directions. The classified camera location class is shown on each image. The details about each label are mentioned in Table 1

8.2 3D human-pose reconstruction

The second goal of this evaluation is to show how this model can enhance the 3D pose reconstruction. For this purpose, we modified the reconstruction approach in [24] to use a fix camera location as mentioned earlier rather than camera rotation from OPT. Throughout this evaluation, we want to emphasize the importance of reconstructing plausible poses more than minimizing the projection error which is described in (6).

Figure 11 shows reconstructing 3D human-pose using the approach in [24], and with same approach but with our camera localization method. The first image (Fig. 11a) is the original input 2D images with landmarks (i.e., black dashed line), and colored lines (i.e., resulted from projecting back the reconstructed model) all are produced from [24]. The second graph shows the reconstructed 3D human pose using [24]. The reconstructed pose is complicated and difficult to understand and imagine. The problem is the misleading camera

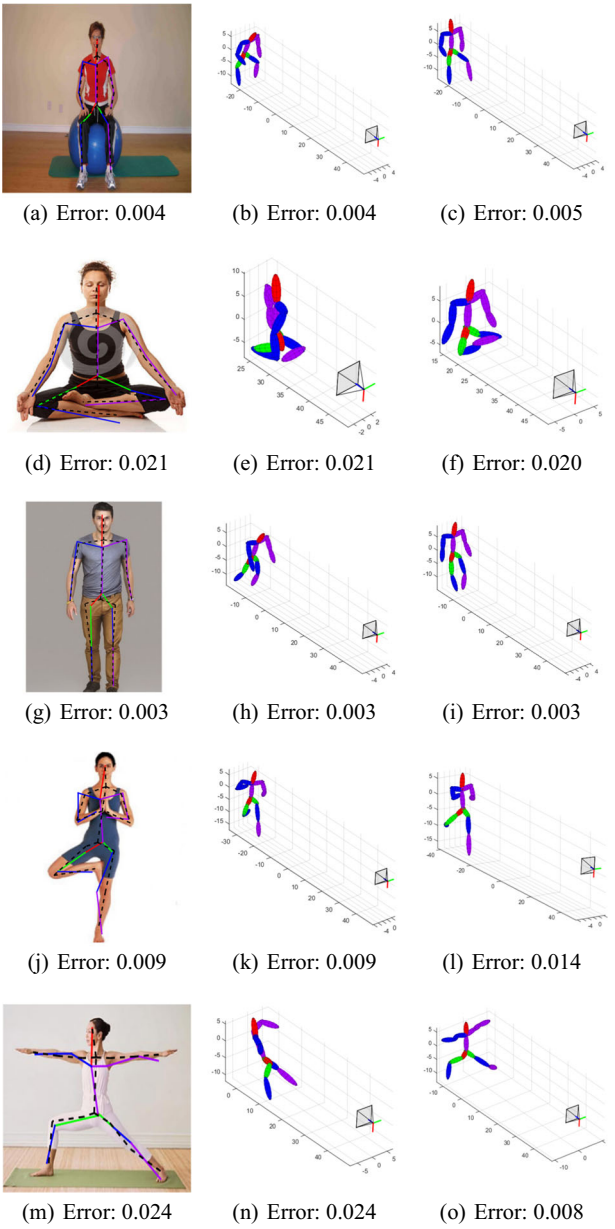


(a) 2D projected images (b) Reconstructed 3D pose using [8] (c) Reconstructed 3D pose using [8] with our camera localization method

Fig. 11 3D human-pose reconstruction using [24] and our enhanced camera localization approach

localization which localized the camera in the side direction. This localization made the reconstruction process restricted and conditioned to a wrong camera direction.

In Fig. 11c, we forced the camera to be placed at specific location (i.e., specific rotation matrix). The chosen camera location is based on applying the learned model in Section 7



(a) Error: 0.004 (b) Error: 0.004 (c) Error: 0.005

(d) Error: 0.021 (e) Error: 0.021 (f) Error: 0.020

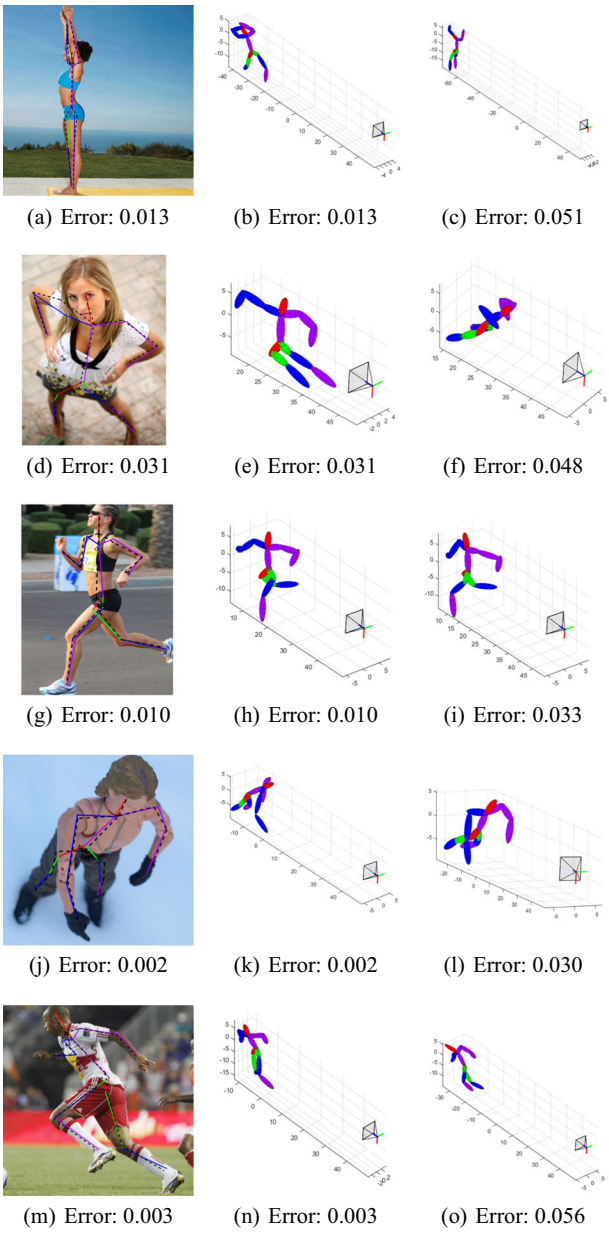
(g) Error: 0.003 (h) Error: 0.003 (i) Error: 0.003

(j) Error: 0.009 (k) Error: 0.009 (l) Error: 0.014

(m) Error: 0.024 (n) Error: 0.024 (o) Error: 0.008

Fig. 12 **First Column:** 2D poses with landmarks. **Second Column:** reconstructed poses using the approach in [24]. **Third Column:** readjusting the approach to use our model of camera localization makes the reconstructed pose much better

to find the class of the camera location. The reconstructed 3D human-pose is plausible and easy to imagine and build. The frontal camera location enhanced the reconstruction process dramatically.



(a) Error: 0.013 (b) Error: 0.013 (c) Error: 0.051

(d) Error: 0.031 (e) Error: 0.031 (f) Error: 0.048

(g) Error: 0.010 (h) Error: 0.010 (i) Error: 0.033

(j) Error: 0.002 (k) Error: 0.002 (l) Error: 0.030

(m) Error: 0.003 (n) Error: 0.003 (o) Error: 0.056

Fig. 13 **First Column:** 2D poses with landmarks. **Second Column:** 3D human-pose reconstruction using the approach in [24], and **Third Column:** readjusting the approach to use our model of camera localization makes the reconstructed pose much better. Dashed lines are the original selected limbs, while the colored lines are the projected ones

The proposed optimization problem in [24] is non-linear and non-convex, and the used approach might fall in a local minima. In addition, wrong camera localization might mislead the pose reconstruction. In this evaluation, we focus on the camera localization importance. Figure 12 shows how our camera localization approach can enhance pose reconstruction for standard viewpoints. The second column shows the reconstructed 3D poses with cameral localization as described in [24]. The third column shows how our model can enhance the reconstruction process. Our camera localization approach is able to reduce the projection error like poses in Figs. 12f, o. On the other hand, some reconstructed poses like in Fig. 12c appears to be more plausible even though it has a higher projection error. For evaluation, we estimated the normalized projection error relative to the input image dimensions (width (w) $\times$ height (h)) as follows:

$$\frac{\min \parallel \mathcal{L} - (I_{|N+1|\times|N+1|} \otimes \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} R)\bar{\mathcal{X}} - t \otimes 1_{2(N+1)\times 1} \parallel_2}{w \times h} \tag{12}$$

Figure 13 shows more additional examples where some of them are using non-standard viewpoints (i.e., the camera location might be in the upper-level or the lower-level). For non-standard viewpoints the reconstruction process suffers from perspective effect. This effect affects the assumed weak perspective projection model, however we show how the typical reconstruction and the updated one using our camera localization model behave in this case. The poses in Figs. 13f, l show better enhancement when the cameras are localized correctly. The pose Fig. 13f has a large difference with our approach. The new camera location enhanced the reconstructed pose compared to the one in Fig. 13e, even though the projection error has increased around 1.5 times.

From the aforementioned discussion, we can summarize many key points about camera localization as follows: First, localizing camera correctly in the 3D space has a strong influence on the reconstructed pose. Second, OPT approach does guarantee the optimal transformation between the 3D model and the 2D landmarks, however that does not mean camera is localized correctly. Third, the approach in [24] focuses on minimizing the projection error, however the reconstructed poses may not be correctly localized.

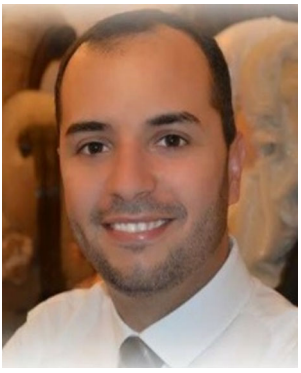
9 Conclusion

In this paper, we presented a camera localization approach that uses projected pose descriptor consists of the relative limbs lengths as well as 2D shape context. This descriptor constructs a projected human-pose related feature. To establish a connection between the constructed features and the camera localization in 3D space, we proposed a virtual sphere scenario where the camera resides on the surface of the virtual sphere. Using probabilistic multi-class logistic regression, we trained a labeled dataset which consists of a set of features, and their related labels identified by camera location index. The widely used orthogonal Procrustes approach is feasible when the used 3D model is strongly related to the projected one. If the 3D model is missed, or even approximated model, the orthogonal Procrustes approach has a high chance to localize the camera at a wrong position. The importance of this work is its ability to classify camera location without using any related 3D model, which gets rid of the 3D model dependency. The proposed camera localization approach can enhance 3D human-pose reconstruction process, and produces more plausible poses.

References

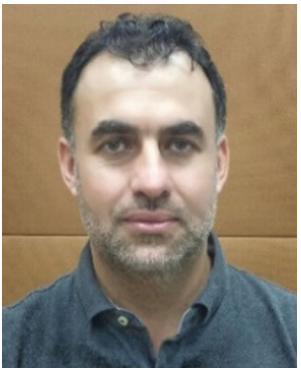
1. Akhter I, Black MJ (2015) Pose-conditioned joint angle limits for 3D human pose reconstruction. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 1446–1455
2. Al-Badarneh A, Khalil M, Al-Hami M (2008) Improving protein 3D structure prediction accuracy using dense regions areas of secondary structures in the contact map. Am J Biochem Biotechnol 4(4):375–384
3. Al-Hami M (2016) Towards a better pose understanding for humanoid robots. PhD thesis, Temple University Libraries
4. Al-Hami M, Khreishah A, Wu J (2013) Video streaming over wireless lan with network coding
5. Al-Hami M, Lakaemper R (2014) Sitting pose generation using genetic algorithm for nao humanoid robots. In: 2014 IEEE workshop on Advanced robotics and its social impacts (ARSO), IEEE, pp 137-142
6. Al-Hami M, Lakaemper R (2015) Towards human pose semantic synthesis in 3D based on query keywords. In: Scitepress
7. Al-Hami M, Lakaemper R (2015) Towards human pose semantic synthesis in 3D based on query keywords. In: VISAPP (3), pp 420–427
8. Al-Hami M, Lakaemper R (2017) Reconstructing 3D human poses from keyword based image database query. In: 2017 International Conference on 3D vision (3DV), IEEE, pp 440–448
9. Awad G, Le DD, Ngo CW, Nguyen VT, Quénot G, Snoek C, Satoh S (2017) Video indexing, search, detection, and description with focus on trecvid. In: Proceedings of the 2017 ACM on international conference on multimedia retrieval, ACM, pp 3–4
10. Belongie S, Malik J, Puzicha J (2002) Shape matching and object recognition using shape contexts. IEEE Trans Pattern Anal Mach Intell 24(4):509–522
11. Carreira J, Agrawal P, Fragkiadaki K, Malik J (2016) Human pose estimation with iterative error feedback. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4733–4742
12. Chen CH, Ramanan D (2017) 3D human pose estimation= 2D pose estimation+ matching. In: CVPR. Volume 2, p 6
13. Ferrari V, Marin-Jimenez M, Zisserman A (2008) Progressive search space reduction for human pose estimation. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2008, IEEE, pp 1–8
14. Gavrila D (2000) Pedestrian detection from a moving vehicle. In: Computer Vision ECCV 2000. Springer, pp 37–49
15. Gross R, Shi J (2001) The cmu motion of body (mobo) database
16. Jokinen K, Wilcock G (2014) Multimodal open-domain conversations with the nao robot. In: Natural interaction with Robots, Knowbots and Smartphones. Springer, pp 213–224
17. Lakaemper R KinectTCP documentation. https://sites.google.com/a/temple.edu/kinecttcp/ Accessed: 2018-08-8
18. Lan X, Huttenlocher DP (2004) A unified spatio-temporal articulated model for tracking. In: IEEE computer society conference on Computer Vision and Pattern Recognition (CVPR), 2004. Volume 1, IEEE, pp I–722
19. Lan X, Huttenlocher DP (2005) Beyond trees: Common-factor models for 2D human pose recovery. In: Tenth IEEE international Conference on Computer Vision (ICCV), 2005. Volume 1, IEEE, pp 470–477
20. Lin CJ, Weng RC, Keerthi SS (2008) Trust region newton method for logistic regression. J Mach Learn Res 9:627–650
21. Mehta D, Sridhar S, Sotnychenko O, Rhodin H, Shafiei M, Seidel HP, Xu W, Casas D, Theobalt C (2017) Vnect: Real-time 3D human pose estimation with a single rgb camera. ACM Transactions on Graphics (TOG) 36(4):44
22. Mousas C, Anagnostopoulos CN (2017) Performance-driven hybrid full-body character control for navigation and interaction in virtual environments. 3D Res 8(2):18
23. Newell A, Yang K, Deng J (2016) Stacked hourglass networks for human pose estimation. In: European conference on computer vision, Springer, pp 483–499
24. Ramakrishna V, Kanade T, Sheikh Y (2012) Reconstructing 3D human pose from 2D image landmarks, pp 573–586
25. Ramanan D (2006) Learning to parse images of articulated bodies. In: Advances in neural information processing systems, pp 1129–1136

26. Rennie JD (2005) Regularized logistic regression is strictly convex. Unpublished manuscript. people.csail.mit.edu/jrennie/writing/convexLR.pdf
27. Sapp B, Taskar B (2013) Modec: Multimodal decomposable models for human pose estimation. In: IEEE Conference onComputer Vision and Pattern Recognition (CVPR), 2013, IEEE, pp 3674–3681
28. Schönemann P (1966) A generalized solution of the orthogonal procrustes problem. Psychometrika 31(1):1–10
29. Sharma D, Lakhmi J, Favorskaya M, Howlett RJ (2015) Fusion of smart, multimedia and computer gaming technologies. Volume 1. Springer, Berlin
30. Taylor CJ (2000) Reconstruction of articulated objects from point correspondences in a single uncalibrated image. In: IEEE conference on Computer Vision and Pattern Recognition (CVPR), 2000. Volume 1, IEEE, pp 677–684
31. The vicon skeleton template. http://mocap.cs.cmu.edu/info.php Accessed: 2016-1-15
32. Varadarajan J, Subramanian R, Bulò SR, Ahuja N, Lanz O, Ricci E (2018) Joint estimation of human pose and conversational groups from social scenes. Int J Comput Vis 126(2-4):410–429
33. Wang C, Wang Y, Lin Z, Yuille AL, Gao W (2014) Robust estimation of 3D human poses from a single image. In: 2014 IEEE conference on Computer vision and pattern recognition (CVPR), IEEE, pp 2369-2376
34. Yang W, Ouyang W, Li H, Wang X (2016) End-to-end learning of deformable mixture of parts and deep convolutional neural networks for human pose estimation, pp 3073–3082
35. Zhang Z (2012) Microsoft kinect sensor and its effect. IEEE Multimedia 19(2):4–10
36. Zhou X, Zhu M, Leonardos S, Derpanis KG, Daniilidis K (2016) Sparseness meets deepness: 3D human pose estimation from monocular video. In: Proceedings of the IEEE conference on computer vision and pattern recognition, pp 4966–4975



Mo'taz Al-Hami is an Assistant Professor for Computer Science at Hashemite University, Jordan. He earned his PhD in computer and information sciences from Temple University, USA. He joined Cadence Design Systems Inc. at Silicon Valley as a Computer Vision engineer/ Deep learning. His research interest focuses on computer vision, Robotics, and Deep learning. Dr. Al-Hami is an IEEE, and ACM member.

Rolf Lakaemper is an Associate Professor (tenured) for Computer Science at Temple University, USA. He has more than 20 years of professional experience in technical fields & education. Also, he has 9 years of experience as game developer/programmer & technical director. Dr. Lakaemper has more than 70 publications in peer reviewed international conferences / journals / books.



Majdi Rawashdeh received his Ph.D. degree in Computer Science from the University of Ottawa, Canada. He is currently an Assistant Professor at Princess Sumaya University for Technology (PSUT), Jordan. His research interests include social media, recommender systems, smart cities, and big data. He has served as a member of the organizing and technical committees of several international conferences and workshops.

M. Shamim Hossain is a Professor at the Department of Software Engineering, College of Computer and Information Sciences, King Saud University, Riyadh, Saudi Arabia. Prof. Shamim is also an Adjunct Professor, School of Electrical Engineering and Computer Science (EECS), University of Ottawa, Canada. Prof. Shamim received his Ph.D. in Electrical and Computer Engineering from the University of Ottawa, Canada. His research interests include Cloud networking, smart environment (smart city, smart health), social media, IoT, edge computing and multimedia for healthcare, deep learning approach for multimedia processing, and multimedia big data. He has authored and coauthored approximately 175 publications including refereed IEEE/ACM/Springer/Elsevier journals, conference papers, books, and book chapters. Recently, his publication has been recognized as the ESI Highly Cited Paper. He has served as a member of the organizing and technical committees of several international conferences and workshops. He has served as co-chair, general chair, workshop chair, publication chair, and TPC for over 12 IEEE and ACM conferences and workshops. Currently, he is the co-chair of the 1st IEEE ICME workshop on Multimedia Services and Tools for smart-health (MUST-SH 2018). He is a recipient of a number of awards, including, the Best Conference Paper Award, the 2016 ACM Transactions on Multimedia Computing, Communications and Applications (TOMM) Nicolas D. Georganas Best Paper Award, and the Research in Excellence Award from the College of Computer and Information Sciences (CCIS), King Saud University (3 times in a row). He is on the editorial board of IEEE Network, IEEE Multimedia, IEEE Access, Journal of Network and Computer Applications (Elsevier), Computers and Electrical Engineering (Elsevier), Human-centric Computing and Information Sciences (Springer), Games for Health Journal, and International Journal of Multimedia Tools and Applications (Springer). Currently, he serves as a lead guest editor of IEEE Communication Magazine, Future Generation Computer Systems (Elsevier), IEEE Network Magazine, and IEEE Access. Previously, he served as a guest editor of IEEE Transactions on Information Technology in Biomedicine (currently JBHI), IEEE Transactions on Cloud Computing, International Journal of Multimedia Tools and Applications (Springer), Cluster Computing (Springer), Future Generation Computer Systems (Elsevier), Computers and Electrical Engineering (Elsevier), Sensors (MDPI), and International Journal of Distributed Sensor Networks. Prof. Shamim is a Senior Member of IEEE, a member of ACM and ACM SIGMM.

Affiliations

Mo'taz Al-Hami[1] · Rolf Lakaemper[2] · Majdi Rawashdeh[3] · M. Shamim Hossain[4]

Rolf Lakaemper
lakamper@temple.edu

Majdi Rawashdeh
m.rawashdeh@psut.edu.jo

M. Shamim Hossain
mshossain@ksu.edu.sa

1 Department of Computer Information System, The Hashemite University, Zarqa, 13115, Jordan

2 Department of Computer & Information Sciences, Temple University, Philadelphia, PA 19122, USA

3 Department of Business Information Technology, Princess Sumaya University for Technology, Amman, 11941, Jordan

4 Department of Software Engineering, King Saud University, Riyadh, Saudi Arabia