End-to-end Dynamic Matching Network for Multi-view Multi-person 3d Pose Estimation

Congzhentao Huang, Shuai Jiang*, Yang Li, Ziyue Zhang, Jason Traish, Chen Deng, Sam Ferguson, and Richard Yi Da Xu

University of Technology Sydney, Sydney, Australia

Abstract. As an important computer vision task, 3d human pose estimation in a multi-camera, multi-person setting has received widespread attention and many interesting applications have been derived from it. Traditional approaches use a 3d pictorial structure model to handle this task. However, these models suffer from high computation costs and result in low accuracy in joint detection. Recently, especially since the introduction of Deep Neural Networks, one popular approach is to build a pipeline that involves three separate steps: (1) 2d skeleton detection in each camera view, (2) identification of matched 2d skeletons and (3) estimation of the 3d poses. Many existing works operate by feeding the 2d images and camera parameters through the three modules in a cascade fashion. However, all three operations can be highly correlated. For example, the 3d generation results may affect the results of detection in step 1, as does the matching algorithm in step 2. To address this phenomenon, we propose a novel end-to-end training scheme that brings the three separate modules into a single model. However, one outstanding problem of doing so is that the matching algorithm in step 2 appears to disjoint the pipeline. Therefore, we take our inspiration from the recent success in Capsule Networks, in which its Dynamic Routing step is also disjointed, but plays a crucial role in deciding how gradients are flowed from the upper to the lower layers. Similarly, a dynamic matching module in our work also decides the paths in which gradients flow from step 3 to step 1. Furthermore, as a large number of cameras are present, the existing matching algorithm either fails to deliver a robust performance or can be very inefficient. Thus, we additionally propose a novel matching algorithm that can match 2d poses from multiple views efficiently. The algorithm is robust and able to deal with situations of incomplete and false 2d detection as well.

Keywords: 3d human pose estimation, end-to-end, multi-view multi-person, dynamic matching

1 Introduction

3d human pose estimation is a fundamental problem in computer vision. It can be applied to various applications such as human-computer interactions, augmented reality and video surveillance. Due to the availability of increasingly

sophisticated datasets, and more and more powerful deep learning models, researchers have made significant progress in this area using deep convolutional neural networks (CNNs). While 3d pose estimation research into a single human under monocular or multi-camera settings has made remarkable advances, fewer works have studied 3d pose estimation of multiple humans, which is a significantly more challenging problem to address. This is primarily due to the occurrences of frequent and sometimes severe occlusions when multiple people are involved. These difficulties have been further exacerbated by the lack of labeling for identifying corresponding people under a multi-view setting.

Despite these difficulties, there are two main reasons why multi-view multi-person 3d pose estimations will become mainstream research. First, models involving multiple people are more generic in many real-world applications compared to those for a single human, such as in supermarkets and factories. Secondly, using multi-cameras, the pose estimation can be made more robust than using a monocular camera due to the multiplied information available from different views, such as when dealing with occlusions.

The methodology for multi-view multi-person 3d pose estimation in many existing studies includes two steps. The first is to predict 2d poses in each view individually using off-the-shelf 2d models [6, 22, 9]. The second is to aggregate these 2d poses and generate their 3d counterpart. One typical idea is to use the so-called 3d Pictorial Structures model (3DPS), which directly generates 3d human poses by exploring an ample state space of all possible human key points or human body parts in 3d space [20, 3]. However, this method lacks efficiency due to the enormous state space needed for exploration.

In contrast to the above two-step models, a recent direction is to use a matching algorithm that identifies matched 2d skeletons from multiple views before the estimation for 3d poses [10]. If the matching algorithm is perfect, the subsequent 3d pose estimation for multiple people can be regarded as multiple 3d pose estimation for a single person. Thus the accuracy will be significantly improved. However, the matching algorithm may make mistakes or even fail. Once a reliable skeleton matching is established, we can then build an effective model in which its pipeline consists of three separate steps: (1) detect 2d skeletons in each camera view, (2) identify matched skeletons and (3) estimate the 3d pose.

An intuitive approach is, of course, to train each of these steps/modules independently. During testing, we can feed the 2d images and camera parameters through these trained modules one by one. However, all of the three operations are highly correlated in both directions of the pipeline. How individual poses are extracted in step 1 will undoubtedly influence the 3d pose estimation result in step 3. The reverse is also true: any adjustments that occur in the 3d estimation in step 3 will ultimately affect the way in which the detection should be carried out in step 1. Therefore, it is essential that the information can be back-propagated in reverse order through step 3 to step 1.

At the same time, when the parameters of the detection module in step 1 are not trained properly, especially during the early stage of the training, the matching algorithm in step 2 may fail to identify the matched skeletons and

catastrophically impact the 3d estimation result in step 3. The traditional one-directional pipeline approach will not improve the parameters of step 1 as each module works independently while having an end-to-end training mechanism allows the model to keep improving the parameters of each step as a result.

However, there is still one bottleneck when we carry out this design. The matching algorithm in step 2 makes the pipeline discontinuous, i.e., it is not a smooth function in which we can back-propagate the changes in parameters freely. However, we can reconcile this with inspiration from Capsule Networks [14, 26]. In CapsNet, the Dynamic Routing step decides how lower layer capsules are fed to their immediate upper layer, either by agreement or expectation-maximization (EM) clustering. In our work, the matching algorithm in step 2 acts in a very similar fashion to the Dynamic Routing. It also decides the feed-forward paths in which information flows from step 1 to step 3, i.e., we apply our matching algorithm to dynamically route/match the poses. This justification and analogy makes our end-to-end approach highly appropriate and is the central theme of our paper.

As one may appreciate, in this end-to-end training mechanism, the dynamic matching step plays a pivotal role. Hence it is vital that we also improve upon the existing works in this area. To this end, we additionally propose a novel matching algorithm which can match multiple 2d poses from multiple views efficiently. The algorithm is robust and can handle situations where there is incomplete and false 2d detection.

In summary, the main contributions of our work are stated below:

- We propose a novel end-to-end training scheme for multi-view multi-person 3d pose estimation. Different from training independent modules separately, our model back-propagates the gradients from the last 3d estimation step to the first 2d detection step, so as to significantly improve the efficiency, robustness and accuracy on 3d pose estimation.
- We propose a multi-view 2d human pose dynamic matching algorithm. This could dynamically match the corresponding 2d poses detected in multiple views for each person involved. The approach does not require the exact number of people in the scene and can handle cases where false detection and severe occlusions exist.
- Experiments on the Shelf and Campus datasets demonstrate that our proposed model outperforms the state-of-the-art methods with respect to both efficiency and accuracy.

2 Related work

In this section, we review the literature related to the techniques of this paper.

2.1 Single-view 2d pose estimation

Single person pose estimation predicts 2d keypoints of the human body in one RGB image. Many existing deep learning-based methods have achieved amazing

results [22, 15, 7] since DeepPose [28] was proposed, which was the first method to use deep neural networks for pose estimation.

For multi-person 2d pose estimation, current state-of-the-art solutions can be divided into two categories. The first category is called the "top-down methods" [9, 12, 17]. It uses an object detection method to detect all the people in the image and sends them separately to a single 2d pose detector to obtain their corresponding 2d poses. In [17], the authors constructed a fully connected graph from a set of detected joint candidates of each person in an image and resolved the joint-to-person association and outlier detection by using integer linear programming. [12] proposed a framework with three components for pose estimation which can extract a high-quality single person region from an inaccurate bounding box. In [9], a two-part network structure was proposed where GlobalNet localizes the "simple" keypoints and the RefineNet deals with the "hard" keypoints. The second category, "bottom-up methods", jointly labels part detection candidates and associates them with individuals by a matching algorithm [23, 6, 16]. The authors in [6] mapped the relationship between keypoints into part affinity fields (PAFs), then clustered detected keypoints into different 3d human poses. [23] interpreted the problem of distinguishing different people in an image as an Integer Linear Programming problem and partitioned part detection candidates into identity clusters. On the basis of [23], the authors in [16] used a stronger part detectors based on ResNet [13] and image-dependent pairwise scores, vastly improving the run time by using an incremental optimization approach.

In our work, we choose the "top-down methods" for their higher accuracy. We adopt the Cascaded Pyramid Network (CPN) [9] as the 2d pose estimator backbone.

2.2 Multi-view 3d pose estimation

Instead of estimating with a single image, multi-view 3d pose estimation methods require image inputs from multiple views, which are believed to obtain better 3d pose estimation than using a monocular camera. Most previous efforts had focused on single person estimation [19, 27]. Traditional methods [1, 4, 5] used 2d pose estimation captured by calibrated cameras to predict 3d poses by point triangulation or 3DPS. Recent works have begun to adopt deep neural networks in this area and have delivered significant achievements. For example, in [18], a volumetric triangulation approach was proposed to project the feature maps produced by 2d pose estimators into 3d volumes, which were then used to predict 3d poses. There are also self-supervised approaches that predict 3d poses separately in different camera views and minimize the distance between pairwise 3d poses after rotating to the same view [21, 25, 8].

As for multi-view multi-person 3d pose estimation, 3DPS is the most widely used approach [2, 3, 20]. It predicts 3d keypoints or 3d body parts by exploring an ample state space and the candidates in the state space are generated by the grid sampling. With the 2d priors given by the 2d detector, the 3d pose can be generated through the maximum likelihood estimation. Recent work [10] has

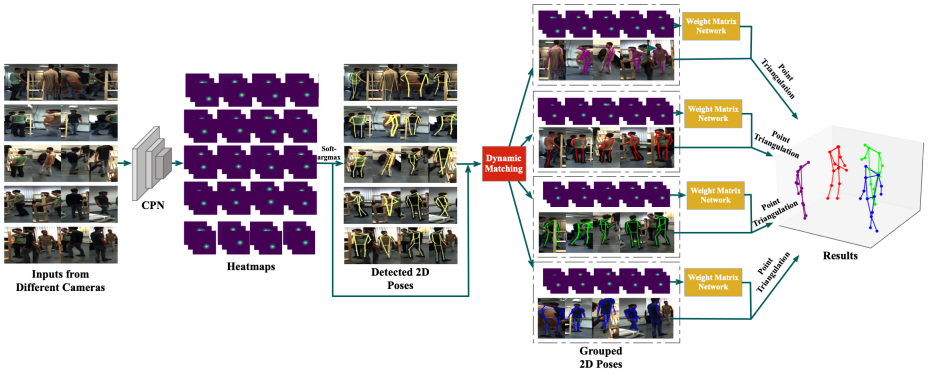

Fig. 1. The framework of our proposed model. First, the images I are input into the 2d human keypoints detector backbone, which is based on CPN [9], to get the heatmaps h. Next, we apply soft-argmax on h to get the corresponding 2d human poses y. Then, we feed both h and y into the dynamic matching module which groups them by identities and automatically determines the number of groups. After that, the heatmaps are sent into a network to get the weight matrices. Last, each cluster is sent to a weight-sharing 3d pose estimator to get the final results Y.

proposed a model to combine person re-identification (re-id) [29, 30] and epipolar geometry to match the pose, followed by the prediction of 3d poses using 3DPS. The shortcoming of this approach is that the speed of the person re-id model is relatively slow, which causes efficiency problems. On the contrary, our approach is efficient on multi-view multi-person 3d pose estimation, which benefits from our novel matching algorithm.

3 Method

In this section, we demonstrate our proposed end-to-end 3d pose estimation model in detail. The scenario assumes there are synchronized video streams from multiple cameras with known parameters, and all cameras capture the same scene with one or more people in it from different views. The goal is to estimate the 3d positions of the keypoints of these people. Note that the exact number of people in the scene is not required.

The inputs of the model are cropped 2d human images from all cameras in the same frame. The images, denoted by I, are cropped by using bounding boxes from either available off-the-shelf 2d human bounding box detectors or ground truths. $I = \{I_n^c | c = 1, 2, \ldots, C, n = 1, 2, \ldots, N_c\}$ where I_n^c is the nth image in the cth view, C is the number of views and N_c is the number of detected bounding boxes in the cth view. The outputs, denoted by Y, are the 3d keypoints of all detected people in the scene. The overview architecture of our model is illustrated in Fig. 1.

In the following text, we will demonstrate the 2d pose estimator backbone, dynamic matching algorithm and 3d pose estimation module respectively.

3.1 2d pose estimator backbone

The 2d pose estimator backbone f_p with trainable weights θ_p consists of GlobalNet and RefineNet. The GlobalNet predicts all keypoints while the RefineNet justifies the "hard" keypoints. The backbone outputs the heatmaps:

$$h_n^c = f_p\left(I_n^c; \theta_p\right), c = 1, 2, \ldots, C, n = 1, 2, \ldots, N_c. \tag{1}$$

The next step is to estimate the 2d positions. To keep the gradient flow, we use soft-argmax instead of argmax to the heatmaps across spatial axes:

$$g_{n,j}^c = e^{h_{n,j}^c} / \left(\int_{q\in\Omega} e^{h_{n,j}^c(q)}\right), \tag{2}$$

where $h_{n,j}^c$ denotes the heatmap of the jth keypoint of the nth detected person in the cth view and Ω denotes the domain of the heatmap. Then the 2d coordinates of the estimated joint $y_{n,j}^c$ is the integration of all locations q in the domain, weighted by their corresponding probabilities (we use y_n^c to denote the 2d coordinates of all keypoints of the nth detected person in the cth view):

$$y_{n,j}^c = \int_{q\in\Omega} q * g_{n,j}^c(q). \tag{3}$$

3.2 Dynamic matching

A matching algorithm is to group 2d poses from different views with people's identities so as to connect the 2d pose detection and 3d pose estimation. It is a challenging task due to several reasons. First of all, there are sizable errors in the estimated 2d poses which can significantly influence the matching accuracy. The second reason is that the number of people in the scene is unknown, which means one cannot cluster these 2d poses to centers like what k-means does. Furthermore, the matching itself is hard to be cycle-consistent. For example, 2d poses y_1^1 and y_1^2 are matched, so do y_1^1 and y_1^3, but y_1^2 and y_1^3 are not matched.

Different from previous methods which compute the matching score for 2d poses, we propose a new matching algorithm that creates a 3d pose subspace first and recursively finds matched 3d poses in this subspace. It resolves both the efficiency and cycle-consistent problems simultaneously. This newly proposed matching algorithm is illustrated in Fig. 2.

3d pose subspace construction To construct the 3d pose subspace, we first enumerate all possible pairs of 2d poses from different views. For each pair of 2d poses, we apply the traditional point triangulation to generate the corresponding 3d pose. All generated 3d poses compose a 3d pose subspace containing a small quantity of correct 3d poses (i.e., matched 2d poses) and a large quantity of incorrect 3d poses. For each pair of 2d keypoints $y_{n,j}^c$ and $y_{m,j}^d$, $c \neq d$, we can get the coefficient matrices for their corresponding homogeneous 3d vectors:

$$A_{n,j}^c = \begin{bmatrix} y_{n,j}^c \\ 1 \end{bmatrix} \times P_c, \qquad A_{m,j}^d = \begin{bmatrix} y_{m,j}^d \\ 1 \end{bmatrix} \times P_d, \tag{4}$$

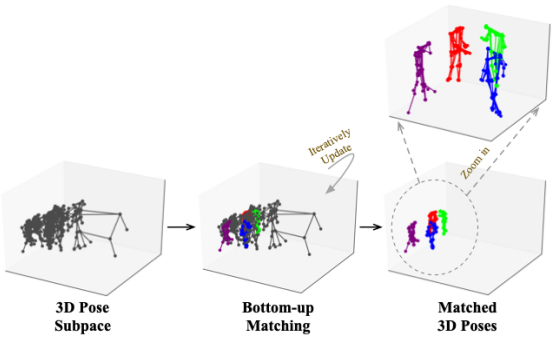

Fig. 2. Overview of the our matching algorithm

where P_c and P_d are the projection matrices of cameras c and d respectively. Thus, the 3d point $\tilde{Y}_{(c_n,d_m),j}$ can be obtained by solving the following linear system:

$$\begin{bmatrix} A^c_{n,j} \\ A^d_{m,j} \end{bmatrix} \cdot \begin{bmatrix} \tilde{Y}_{(c_n,d_m),j} \\ 1 \end{bmatrix} = 0. \tag{5}$$

We use $\tilde{Y}_{(c_n,d_m)}$ to denote the calculated 3d pose given 2d poses y^c_n and y^d_m. The number of 3d poses constructed is

$$T = \sum_{c=1}^{C} N_c \sum_{d=c+1}^{C} N_d. \tag{6}$$

Bottom-up matching After the construction of 3d pose subspace, we now need to pick out the correct 3d poses. The idea we distinguish the correct 3d poses with incorrect ones is that, the correct 3d poses are almost always calculated by 2d poses belonging to the same person. For example, if a person is captured by four cameras, we will detect four 2d poses which are used to construct six 3d poses, and these 3d poses are almost always very similar to each other, i.e. their distances are very small. Therefore, if the distance between a pair of 3d poses is sufficiently small, their corresponding 2d poses are regarded as a match.

We use the euclidean distance as the measurement between pairwise 3d poses $\tilde{Y}_{(c_n,d_m)}$ and $\tilde{Y}_{(c'_p,d'_q)}$:

$$E(\tilde{Y}_{(c_n,d_m)}, \tilde{Y}_{(c'_p,d'_q)}) = \|\tilde{Y}_{(c_n,d_m)} - \tilde{Y}_{(c'_p,d'_q)}\|_F, \tag{7}$$

where $\|\cdot\|$ is the Frobenius norm. Since we do not need to calculate the distance between 3d poses coming from the same views (i.e. $c = c'$ and $d = d'$), the number of distances calculated is

$$|D| = \sum_{c=1}^{C} \sum_{d=c+1}^{C} (T - N_c N_d) \cdot N_c N_d / 2. \tag{8}$$

where D denotes the set of distances between all possible pairwise 3d poses and $|\cdot|$ here is the cardinality.

In order to efficiently obtain all matches, we propose a bottom-up matching algorithm. Suppose the matching result is stored in a set $S = \{s_k | k = 1, 2, \dots\}$ where s_k is a subset which contains the indices of 2d poses belonging to the same person. We initialize S as an empty set and update it by iterations. In each iteration, we first find the minimal distance in D, denoted by $D_{\min}$ which relates to two 3d poses generated by four 2d poses (three if one of them is shared by both pairs), say $y_{n_1}^{c_1}$, $y_{n_2}^{c_2}$, $y_{m_1}^{d_1}$ and $y_{m_2}^{d_2}$, and their corresponding indices can be denoted by a set of view-image pairs $V = \{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\}$. Next, we find a subset s_k^* in S which contains any of the indices in V. If no subset is found, we add an empty set $s_k^* = \{\}$ into S. This finding process is referred as $F(S, V)$. Then we update s_k^* by $s_k^* = s_k^* \cup V$. Note that an index will be dropped if s_k^* has already contained another index from the same view. After the update, $D_{\min}$ will be removed from D. We repeat the above steps until $D_{\min} > \rho$ where ρ is a predefined threshold. The complete bottom-up matching algorithm is presented in Algorithm 1.

Algorithm 1 Bottom-up matching algorithm

Input: D, ρ
Output: S

InitializeS $\leftarrow \emptyset$
$D_{\min} \leftarrow \min(D)$
while $D_{\min} < \rho$ **do**
  $\{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\} \leftarrow D_{\min}$
  $V \leftarrow \{(c_1, n_1), (c_2, n_2), (d_1, m_1), (d_2, m_2)\}$
  $s_k^* \leftarrow F(S, V) \cup V$
  $D \leftarrow D \setminus D_{\min}$
  $D_{\min} \leftarrow \min(D)$

Through the matching algorithm we can get the resultant $S = \{s_1, s_2, \dots, s_K\}$ where K is the estimated number of people in the scene. It is determined automatically by the algorithm. According to the indices in s_k we can select the 2d poses and heatmaps of the kth person and group them together:

$$y^{(k)}, h^{(k)} = G\left(y, h, s_k\right), k \in [1, K], \tag{9}$$

where y and h are the 2d poses and heatmaps for all people from all views, and function $G(\cdot)$ does the operations of both selection and grouping. Each group of 2d poses and heatmaps will be sent to the subsequent module for 3d pose estimation.

This dynamic matching module plays a similar role as the dynamic routing (especially the EM routing) in CapsNet. The difference between them is that the dynamic routing integrates the features from lower capsules by using weighted

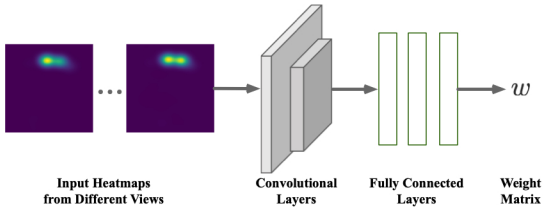

Fig. 3. The structure of the weight matrix network

summation, while our dynamic matching clusters the 2d poses and corresponding heatmaps without any value changes.

Note that the proposed dynamic matching requires at least three views of the scene, which can be inferred form Eq. (8). When there are only two views, $|D|$ in Eq. (8) becomes 0, which invalidates the whole matching algorithm. Therefore, for this special case of two views, we use auxiliary approaches such as the above mentioned person re-id and epipolar geometry.

3.3 3d pose estimation

Given the grouped 2d poses and heatmaps of each person, we can reconstruct their 3d poses in several ways. The point triangulation described previously is one of them. However, we are now using the 2d keypoints from all views instead of a pair of views, and the corresponding linear system becomes:

$$A_j^{(k)} \cdot \begin{bmatrix} Y_j^{(k)} \\ 1 \end{bmatrix} = 0, \tag{10}$$

where $A_j^{(k)}$ is a matrix concatenating the homogeneous 3d vectors of all views for the $j$the keypoint of the kth person.

The point triangulation is an efficient 3d pose estimation algorithm with strong theoretical supports but often produces imprecise 3d poses if there are erroneous detection of 2d poses. The reason is that the coordinates of different keypoints are computed separately. This phenomenon can occur quite frequently at the beginning of training when the 2d pose detection module has not been trained well enough, which in turn affects the improvements of the 2d detection.

To deal with the inaccuracy, inspired by [18], we add a learnable module f_w illustrated in Fig. 3 before the point triangulation, which accepts the heatmaps as inputs:

$$w_j^{(k)} = f_w\left(h_j^{(k)}; \theta_w\right). \tag{11}$$

The output $w_j^{(k)}$ is a weight matrix which is in the same size of $A_j^{(k)}$. We add it to Eq. (10) and have

$$\left(w_j^{(k)} \circ A_j^{(k)}\right) \cdot \begin{bmatrix} Y_j^{(k)} \\ 1 \end{bmatrix} = 0, \tag{12}$$

The original module in [18] predicts a scalar weight for each view denoting how important the keypoints of a view will be. However, scalar weights cannot reflect the details of importance. For example, if a detected keypoint is inaccurate on the horizontal axis but very accurate on the vertical axis, scalar weights have to balance their importance and there will be no difference of importance if we switch the accuracy for both axes. Therefore, we propose to use a weight matrix instead of a scalar weight to better learn the importance so that the accuracy of point triangulation can be further improved.

3.4 Loss function

Our loss function contains two parts, the 2d reprojection loss and the 3d mean square error (MSE) loss. The reason we add the 2d reprojection loss is that, if we only use the 3d MSE loss, there would be infinite points that have the same loss value but target at the 3d ground truth in different directions. The 2d reprojection loss can indicate the correct direction by constraining projected 2d poses from different views.

The 3d MSE loss between the estimated 3d pose and 3d ground truth is defined as:

$$L_{\text{mse}}^{\text{3d}} = \sum_{k=1}^{K} \frac{1}{|Y^{(k)}|} \|Y^{(k)} - Y_{gt}^{(k)}\|_F^2. \tag{13}$$

The 2d reprojection loss between the reprojected 2d pose from the computed 3d pose and the detected 2d pose from backbone is defined as:

$$L_{\text{repj}}^{\text{2d}} = \sum_{k=1}^{K} \sum_{c=1}^{C} \frac{1}{|y_c^{(k)}|} \|\tilde{y}_c^{(k)} - y_c^{(k)}\|_F^2, \tag{14}$$

where

$$\tilde{y}_c^{(k)} = \left[p_1 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} / p_3 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix}, p_2 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} / p_3 \cdot \begin{bmatrix} Y_k \\ 1 \end{bmatrix} \right], \tag{15}$$

and

$$P_c = \begin{bmatrix} p_1 \ p_2 \ p_3 \end{bmatrix}^T. \tag{16}$$

Thus, the total loss of our model is defined as:

$$L = L_{\text{mse}}^{\text{3d}} + \alpha L_{\text{repj}}^{\text{2d}}, \tag{17}$$

where α is a weight coefficient.

4 Experiments

4.1 Datasets

We conduct experiments on two standard datasets for multi-view multi-person 3d human pose estimation.

Shelf [2]: The Shelf dataset is one of the public 3d multi-person human pose datasets in multi-view setting. It consists of 3200 frames from 5 synchronized cameras along with the 2d pose annotations and 3d pose ground truth derived by pose triangulation. There are 4 human subjects interacting with each other in a small room. All 3200 frames are split into an evaluation set (frame 300-600) and a training set (other frames).

Campus [2]: The Campus dataset contains three human subjects interacting with each other in an outdoor environment. The scene is captured by three calibrated cameras. The dataset consists of 2000 frames and is divided into an evaluation set (frame 350-470, frame 650-750) and a training set (other frames).

For the evaluation protocol, we use the percentage of correctly estimated parts (PCP@0.5) to measure the model performance, which is the most commonly adopted in this area [2, 10].

4.2 Implementation details

As for the data preprocessing, we crop the images with bounding boxes estimated by an off-the-shelf 2d human detector, Yolo [24]. The 2d pose detection backbone is the same as [9] with pretrained weights, which outputs heatmaps and connects to a soft-argmax function to obtained the 2d poses. The dynamic matching module is implemented according to Algorithm 1. The 3d pose estimator consists of two convolutional layers and three fully-connected layers. The weight coefficient α in the loss function is set to 2. We choose the Adam optimizer with a learning rate of 10^{-6} which reduces by a decay factor of 10 in each epoch. The training set and evaluation set are kept the same as described in the datasets.

4.3 Ablation study

Our first experiment is to verify the effectiveness of different settings for our model through the ablation study on the Shelf dataset.

End-to-end vs multi-step architecture Our model is end-to-end and can predict the 3d poses from 2d human images as a whole. An alternative is to divide the model into three consecutive steps which deal with the 2d pose detection, matching and 3d pose estimation separately. We compare these two architectures and the results are presented in Table 1.

From the table, we can see that the performance of our end-to-end model is better than the multi-step model for all three people in the scene. The average improvement is 0.72. This demonstrates that the end-to-end model is more capable of learning the features of human poses which refines the 2d pose detection with gradients flowing back from the overall loss function.

Table 1. The PCP@0.5 performance of the alternative multi-step model and our end-to-end model on the Shelf dataset. They are using the same 2d pose detection backbone, matching algorithm, 3d pose estimator and loss function.

	Actor 1	Actor 2	Actor 3	Average
Multi-step	98.12	95.16	96.77	96.67
End-to-end (ours)	**98.75**	**96.22**	**97.20**	**97.39**

Matching method Given the 2d poses obtained from the 2d detection module, we propose a novel matching algorithm to group the 2d poses and heatmaps by identities. There are two existing matching methods in the literature, the person re-id and epipolar geometry. The former finds matches by using the re-id appearance matrix as confidence scores, while the latter uses epipolar geometry affinity matrix as the confidence scores. The comparison between these three matching methods is shown in Table 2.

Table 2. Comparison of matching methods including the person re-id, epipolar geometry and our algorithm on the Shelf dataset over the PCP@0.5 and time cost. All three methods use the same 2d pose detector and 3d pose estimator.

	Actor 1	Actor 2	Actor 3	Average	Time (s)
Person re-id	97.62	93.72	95.69	95.68	6.73
Epipolar geometry	97.28	91.76	91.27	93.44	**0.64**
Our method	**98.75**	**96.22**	**97.20**	**97.39**	0.96

The results show that our matching method achieves the best performance among the three, with average improvements of 1.71 and 3.95. The time cost of person re-id is the highest while that of epipolar geometry is the lowest. Our matching method is slightly slower than epipolar geometry, but still much faster than person re-id. This experiment demonstrates that our matching algorithm is robust and efficient. The reason is that both person re-id and epipolar geometry use 2d information, thus there may be cases where the poses of different people result in a larger confidence score than those of the same person because of the angle of camera views or imprecise 2d detection. On the contrary, our method finds the matches in the 3d pose subspace directly, which leverages the information inequality between the 2d and 3d spaces and makes our method more robust and insensitive to imprecise or even incorrect 2d poses.

3d pose estimation method As described in the method section, we use the point triangulation with a learnable weight matrix to estimate 3d poses. Alternatives include the sole point triangulation or the original learnable triangulation network [18]. We compare these two methods with ours and the result is presented in Table 3.

Table 3. Performance of our 3d pose reconstruction method compared with the point triangulation and learnable triangulation on the Shelf dataset. They are implemented with the same 2d pose detection backbone and dynamic matching.

	Actor 1	Actor 2	Actor 3	Average
Point triangulation	98.05	91.17	92.78	94.00
Learnable triangulation	98.64	95.83	96.91	97.13
Our method	**98.75**	**96.22**	**97.20**	**97.39**

Table 4. Comparison of multi-view multi-person 3d pose estimation models on the Shelf and Campus datasets under PCP@0.5. All results are obtained from the original papers except for the (*) which only provides the average performance (in the parentheses) and its results on body parts presented here are from our own experiments using the authors' published code.

Shelf dataset		Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts	Average
Belagiannis et al. [2]	Actor 1	89.30	90.20	72.16	60.59	37.12	70.61	66.05	
	Actor 2	72.10	92.80	80.11	44.20	46.30	71.80	64.97	71.39
	Actor 3	94.66	96.35	91.00	89.00	45.80	94.50	83.16	
Belagiannis et al. [3]	Actor 1	96.29	100.00	82.24	66.67	43.17	86.07	75.26	
	Actor 2	78.95	100.00	82.58	47.37	50.00	78.95	69.67	77.51
	Actor 3	98.00	100.00	93.15	92.30	56.50	97.00	87.59	
Ershadi-Nasab et al. [11]	Actor 1	98.27	97.34	92.57	83.33	95.94	96.83	93.29	
	Actor 2	63.05	94.61	78.33	33.38	95.30	93.45	75.85	87.99
	Actor 3	98.15	94.12	94.43	89.82	97.41	96.34	94.83	
Dong et al. [10]*	Actor 1	88.17	100.00	99.82	99.28	99.82	100.00	98.60	96.76 (96.90)
	Actor 2	97.30	100.00	98.65	71.62	100.00	100.00	93.78	
	Actor 3	94.41	100.00	95.96	96.27	100.00	100.00	**97.89**	
Our model	Actor 1	88.89	100.00	99.82	99.46	100.00	100.00	**98.75**	
	Actor 2	100.00	100.00	100.00	81.08	100.00	100.00	**96.22**	**97.39**
	Actor 3	90.06	100.00	95.65	95.96	95.96	99.38	97.20	
Campus dataset		Head	Torso	Upper Arms	Lower Arms	Upper Legs	Lower Legs	All parts	Average
Belagiannis et al. [2]	Actor 1	93.62	49.94	82.85	77.80	86.23	91.39	82.01	
	Actor 2	97.40	41.13	90.36	39.65	73.87	89.02	72.43	75.79
	Actor 3	81.26	69.67	77.58	61.84	83.44	70.27	73.72	
Belagiannis et al. [3]	Actor 1	96.55	93.10	96.55	86.21	93.10	96.55	93.45	
	Actor 2	98.24	48.82	97.35	42.94	75.00	89.41	75.65	84.49
	Actor 3	93.20	85.44	89.81	74.76	91.75	76.21	84.37	
Ershadi-Nasab et al. [11]	Actor 1	97.31	94.16	96.83	87.48	93.67	97.27	94.18	
	Actor 2	98.73	95.41	94.12	78.98	98.94	95.34	92.89	90.56
	Actor 3	95.36	84.37	93.16	70.34	88.36	81.38	84.62	
Dong et al. [10]*	Actor 1	100.00	100.00	97.96	89.80	100.00	100.00	97.55	95.85 (96.30)
	Actor 2	97.88	100.00	100.00	67.72	100.00	100.00	93.33	
	Actor 3	99.28	99.28	98.91	89.86	97.46	97.83	96.67	
Our model	Actor 1	100.00	100.00	98.98	90.82	100.00	100.00	**97.96**	
	Actor 2	99.47	100.00	100.00	74.34	100.00	100.00	**94.81**	**96.71**
	Actor 3	100.00	100.00	99.64	90.58	97.10	97.46	**97.39**	

We can see from the table that our method outperforms the other two methods by 3.39 and 0.26 respectively in average. This demonstrates that (1) the 3d poses estimated by point triangulation is not accurate enough, (2) adding learnable scalar weights can significant improve the performance and (3) using

a learnable weight matrix instead of the scalar weights can further improve the model's robustness.

4.4 Comparison with previous works

We compare our model with existing state-of-the-art models for multi-view multi-person 3d pose estimation on both datasets. The models compared are:

- Belagiannis et al. [2], the first one applying the 3DPS to 3d pose estimation for multiple humans.
- Belagiannis et al. [3], an improved version of their previous work.
- Ershadi-Nasab et al. [11], an extension of the 3DPS.
- Dong et al. [10], which uses person re-id and geometry methods to match 2d poses.

For the Campus dataset, since the number of views is insufficient to generate enough 3d pose candidates, we use person re-id and epipolar geometry as auxiliaries in our matching algorithm. The comparison results are shown in Table 4.

On both datasets our model surpasses the state-of-the-art methods in almost all cases. The average performance of our model is 97.39 and 96.71 respectively with improvements of 0.63 and 0.86 comparing with the second best model (0.49 and 0.41 improvements if compared with the results from their paper). It is noteworthy that, the performance of existing models on the lower arms of Actor 2 in Shelf dataset is quite low, while ours achieves 81.08 with a huge improvement of 9.46. We notice that there exists a large quantity of occlusions in this case, which means our model can better handle occlusions than others in a multi-person setting.

5 Conclusion

In this paper, we have proposed a novel end-to-end dynamic matching network for multi-view multi-person 3d pose estimation. Different from previous studies, the end-to-end scheme of our work enables the gradients to flow back from the 3d pose estimation module to the 2d pose detection backbone. A bottom-up dynamic matching algorithm is proposed to group the 2d poses and heatmaps by identities so as to connect the 2d pose detector and the 3d pose estimator. The algorithm is efficient and robust and able to automatically determine the number of people in the scene. The ablation study verified the effectiveness of each part of our model and the experimental results on the Shelf and Campus datasets demonstrate that our proposed model is superior to the state-of-the-art models with respect to accuracy, robustness and efficiency.

References

1. Amin, S., Andriluka, M., Rohrbach, M., Schiele, B.: Multi-view pictorial structures for 3d human pose estimation. In: Bmvc. vol. 2, p. 7. Citeseer (2013)
2. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3d pictorial structures for multiple human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1669–1676 (2014)
3. Belagiannis, V., Amin, S., Andriluka, M., Schiele, B., Navab, N., Ilic, S.: 3d pictorial structures revisited: Multiple human pose estimation. IEEE transactions on pattern analysis and machine intelligence **38**(10), 1929–1942 (2015)
4. Bergtholdt, M., Kappes, J., Schmidt, S., Schnörr, C.: A study of parts-based object class detection using complete graphs. International journal of computer vision **87**(1-2), 93 (2010)
5. Burenius, M., Sullivan, J., Carlsson, S.: 3d pictorial structures for multiple view articulated pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 3618–3625 (2013)
6. Cao, Z., Simon, T., Wei, S.E., Sheikh, Y.: Realtime multi-person 2d pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7291–7299 (2017)
7. Carreira, J., Agrawal, P., Fragkiadaki, K., Malik, J.: Human pose estimation with iterative error feedback. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 4733–4742 (2016)
8. Chen, X., Lin, K.Y., Liu, W., Qian, C., Lin, L.: Weakly-supervised discovery of geometry-aware representation for 3d human pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 10895–10904 (2019)
9. Chen, Y., Wang, Z., Peng, Y., Zhang, Z., Yu, G., Sun, J.: Cascaded pyramid network for multi-person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7103–7112 (2018)
10. Dong, J., Jiang, W., Huang, Q., Bao, H., Zhou, X.: Fast and robust multi-person 3d pose estimation from multiple views. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 7792–7801 (2019)
11. Ershadi-Nasab, S., Noury, E., Kasaei, S., Sanaei, E.: Multiple human 3d pose estimation from multiview images. Multimedia Tools and Applications **77**(12), 15573–15601 (2018)
12. Fang, H.S., Xie, S., Tai, Y.W., Lu, C.: Rmpe: Regional multi-person pose estimation. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 2334–2343 (2017)
13. He, K., Zhang, X., Ren, S., Sun, J.: Deep residual learning for image recognition. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 770–778 (2016)
14. Hinton, G.E., Sabour, S., Frosst, N.: Matrix capsules with em routing. In: International conference on learning representations (2018)
15. Huang, S., Gong, M., Tao, D.: A coarse-fine network for keypoint localization. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3028–3037 (2017)
16. Insafutdinov, E., Pishchulin, L., Andres, B., Andriluka, M., Schiele, B.: Deepercut: A deeper, stronger, and faster multi-person pose estimation model. In: European Conference on Computer Vision. pp. 34–50. Springer (2016)

17. Iqbal, U., Gall, J.: Multi-person pose estimation with local joint-to-person associations. In: European Conference on Computer Vision. pp. 627–642. Springer (2016)
18. Iskakov, K., Burkov, E., Lempitsky, V., Malkov, Y.: Learnable triangulation of human pose. arXiv preprint arXiv:1905.05754 (2019)
19. Joo, H., Liu, H., Tan, L., Gui, L., Nabbe, B., Matthews, I., Kanade, T., Nobuhara, S., Sheikh, Y.: Panoptic studio: A massively multiview system for social motion capture. In: Proceedings of the IEEE International Conference on Computer Vision. pp. 3334–3342 (2015)
20. Joo, H., Simon, T., Li, X., Liu, H., Tan, L., Gui, L., Banerjee, S., Godisart, T., Nabbe, B., Matthews, I., et al.: Panoptic studio: A massively multiview system for social interaction capture. IEEE transactions on pattern analysis and machine intelligence **41**(1), 190–204 (2017)
21. Kocabas, M., Karagoz, S., Akbas, E.: Self-supervised learning of 3d human pose using multi-view geometry. arXiv preprint arXiv:1903.02330 (2019)
22. Newell, A., Yang, K., Deng, J.: Stacked hourglass networks for human pose estimation. In: European conference on computer vision. pp. 483–499. Springer (2016)
23. Pishchulin, L., Insafutdinov, E., Tang, S., Andres, B., Andriluka, M., Gehler, P.V., Schiele, B.: Deepcut: Joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 4929–4937 (2016)
24. Redmon, J., Farhadi, A.: Yolov3: An incremental improvement. arXiv preprint arXiv:1804.02767 (2018)
25. Rhodin, H., Spörri, J., Katircioglu, I., Constantin, V., Meyer, F., Müller, E., Salzmann, M., Fua, P.: Learning monocular 3d human pose estimation from multi-view images. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 8437–8446 (2018)
26. Sabour, S., Frosst, N., Hinton, G.E.: Dynamic routing between capsules. In: Advances in neural information processing systems. pp. 3856–3866 (2017)
27. Song, Y., Morency, L.P., Davis, R.: Multimodal human behavior analysis: learning correlation and interaction across modalities. In: Proceedings of the 14th ACM international conference on Multimodal interaction. pp. 27–30. ACM (2012)
28. Toshev, A., Szegedy, C.: Deeppose: Human pose estimation via deep neural networks. In: Proceedings of the IEEE conference on computer vision and pattern recognition. pp. 1653–1660 (2014)
29. Zhong, Z., Zheng, L., Cao, D., Li, S.: Re-ranking person re-identification with k-reciprocal encoding. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 1318–1327 (2017)
30. Zhong, Z., Zheng, L., Zheng, Z., Li, S., Yang, Y.: Camera style adaptation for person re-identification. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. pp. 5157–5166 (2018)