Cross-View Person Identification Based on Confidence-Weighted Human Pose Matching

Guoqiang Liang, Xuguang Lan, *Member, IEEE*, Xingyu Chen, Kang Zheng, Song Wang, *Senior Member, IEEE*, and Nanning Zheng, *Fellow, IEEE*

***Abstract*— Cross-view person identification (CVPI) from *multiple temporally synchronized* videos taken by multiple wearable cameras from different, varying views is a very challenging but important problem, which has attracted more interest recently. Current state-of-the-art performance of CVPI is achieved by matching appearance and motion features across videos, while the matching of pose features does not work effectively given the high inaccuracy of the 3D pose estimation on videos/images collected in the wild. To address this problem, we first introduce a new metric of confidence to the estimated location of each human-body joint in 3D human pose estimation. Then, a mapping function, which can be hand-crafted or learned directly from the datasets, is proposed to combine the inaccurately estimated human pose and the inferred confidence metric to accomplish CVPI. Specifically, the joints with higher confidence are weighted more in the pose matching for CVPI. Finally, the estimated pose information is integrated into the appearance and motion features to boost the CVPI performance. In the experiments, we evaluate the proposed method on three wearable-camera video datasets and compare the performance against several other existing CVPI methods. The experimental results show the effectiveness of the proposed confidence metric, and the integration of pose, appearance, and motion produces a new state-of-the-art CVPI performance.**

***Index Terms*— Confidence metric, cross-view person identification, human pose matching.**

I. Introduction

VIDEO-BASED surveillance has been widely used in many security, civil, and military applications. Traditional surveillance videos are captured by multi-camera network, where all the cameras are installed at fixed locations. Since they cannot move freely, the cameras can only cover limited areas from prefixed view angles. In recent years, wearable cameras, like Google Glass and GoPro, have been introduced to many applications to expand the video coverage. Compared with fixed cameras, wearable cameras are mounted over the head of the wearers and can move with the wearers to better capture the scene of interest. For example, in a sport game or a protest event, multiple policemen can wear cameras to record videos at different locations and from different view angles, which can facilitate the detection of abnormal persons and activities.

One fundamental problem in analyzing multiple videos taken by multiple wearable cameras is **cross-view person identification (CVPI)** – identifying the same person from these multiple videos [1]. As in [1], we assume all the videos are **temporally synchronized**, which can be achieved by sharing a clock across all the cameras. Given the temporal synchronization, if the corresponding frames across multiple videos cover the same person, this person must bear a unique pose and motion in 3D space. As a result, we can estimate the 3D pose and motion of each video and match them across these videos to accomplish CVPI. As in many person re-identification methods, appearance feature matching can also be used for CVPI [1], although the extracted 2D appearance features may vary under different views.

Zheng *et al.* [1] have shown the effectiveness of using motion features for CVPI, especially when using the view-invariant motion features extracted by supervised deep learning. It also shows that the appearance features and motion features can complement each other to improve the CVPI performance. However, the use of pose features for CVPI [2] is not very successful due to the high inaccuracy of the 3D human pose estimation (HPE) on videos/images collected in the wild. For example, for the body joints in 3D HPE, the average Euclidean distance between the ground-truth 3D locations and the estimations by [3] is 71.90 mm, which is about 1/7 of the length of human torso in Human 3.6M dataset [4]. Even so, this dataset is collected in a highly-controlled lab environment with very simple background. The accuracy of estimated 3D location of joints will be much worse in outdoor environments with more camera motion and view-angle changes [5].

What's more, the localization accuracy of different joints is highly inconsistent in 3D HPE. As described in [3], the 3D localization error at wrist is much larger than that at hip – 101.48 mm versus 28.81mm. This inconsistency comes from

Manuscript received May 18, 2018; revised November 20, 2018 and January 31, 2019; accepted February 8, 2019. Date of publication February 15, 2019; date of current version June 13, 2019. This work was supported in part by the Key Project of Trico-Robot Plan of NSFC under Grant 91748208, in part by the National Science and Technology Major Project under Grant 2018ZX01028-101, in part by the Key Project of Shaanxi province under Grant 2018ZDCXLGY0607, in part by the NSFC under Grant 61573268, Grant 61672376, Grant U1803264, and in part by the NSF under Grant 1658987. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xudong Jiang. *(Corresponding authors: Xuguang Lan; Song Wang.)*

G. Liang, X. Lan, X. Chen, and N. Zheng are with the Institute of Artificial Intelligence and Robotics, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: gqliang@stu.xjtu.edu.cn; xglan@mail.xjtu.edu.cn; nnzheng@mail.xjtu.edu.cn; chenxingyu_1990@163.com).

K. Zheng is with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: zheng37@email.sc.edu).

S. Wang is with the School of Computer Science and Technology, Tianjin University, Tianjin 30072, China, and also with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC 29208 USA (e-mail: songwang@cec.sc.edu).

Digital Object Identifier 10.1109/TIP.2019.2899782

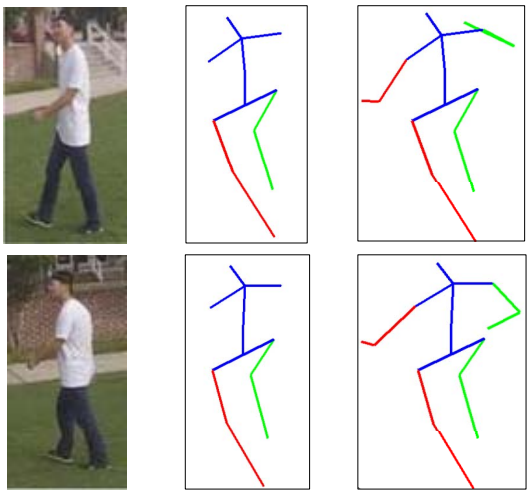


Fig. 1. An illustration of 3D human pose estimation with confidence. Left: Different views of the same person taken at the same time. Middle: (HPE-localized) joints with confidence larger than 0.7. Right: (HPE-localized) joints with confidence larger than 0.1.

possible occlusions and different degrees of freedom. Using poorly estimated joint locations may significantly reduce the CVPI accuracy based on matching human poses. An example is shown in Fig. 1, where two images of the same person are taken from different views at the same time. Due to self-occlusion, the estimated locations of the right arm by a 3D HPE algorithm are largely incorrect. Without considering the right arm, we can match the estimated 3D poses from these two images much better. Inspired by this, we propose to estimate the confidence of each localized joint, which is then combined with human pose to boost CVPI. Note that the goal of this paper is not to develop a new 3D HPE algorithm with higher accuracy. Instead, we simply select one existing HPE algorithm, derive confidence at each joint and then apply them for CVPI.

In this paper, we propose a confidence-weighted human pose matching method for cross-view person identification. First, we introduce a confidence metric for the estimated location of each joint in 3D HPE. Considering the steps of 3D HPE, we derive the confidence at each joint from three aspects: (1) 2D confidence, which is derived from the process of 2D HPE; (2) 3D confidence, which refers to the certainty of 3D HPE from the 2D heat-maps of joints generated by 2D HPE; (3) temporal confidence, which reflects the stability of joints' locations over time. Then, a mapping function is developed to combine the 3D poses and corresponding confidence metrics of two videos to compute their pose distance. Specifically, we employ two different ways to construct the mapping functions. One is based on direct multiplication while the other is based on deep learning. For a given video, the matched video is the one with the smallest matching pose distance in the gallery dataset. Finally, we integrate the pose matching with appearance and motion feature matching for CVPI. In experiments, we evaluate the proposed method on the three datasets SEQ1 [2], SEQ2 [2] and SYN [1]. All of them are human walking videos, which are taken by two GoPro cameras from different views at the same time. As a result, they are composed of temporally synchronized video pairs that capture the same walking subject from different views. The experimental results on these datasets show the effectiveness of the introduced confidence metric and the complementarity of poses, appearance and motion features in CVPI.

There are mainly three contributions: 1) We introduce a new metric of confidence, which represents the certainty of estimated joint's location in 3D HPE; 2) We propose to combine the inaccurately estimated human pose with inferred confidence metric to accomplish CVPI, which illustrates the effectiveness of confidence metric for pose-based CVPI; 3) On the SEQ1, SEQ2 and SYN datasets, the combination of pose, appearance and motion features achieves a new state-of-the-art CVPI performance, which shows that the use of the poses weighted by confidence can complement the appearance and motion features in CVPI.

A preliminary conference version of this paper was published in [6]. In this journal version, we propose to use fully-connected network to combine the confidence metrics and the estimated pose for video distance computation, which extends and outperforms the original multiplication-based method in [6]. We also include in this paper more related works on traditional person identification and confidence metrics and more experiments to show the influence of different configurations and parameters.

II. Related Work

In this section, we briefly review the related works on person identification, human pose estimation, and confidence metric.

A. Person Identification

The aim of CVPI is to associate person from temporally synchronized videos taken by wearable cameras, which is proposed by Zheng *et al.* [1], [2]. Compared with traditional person re-identification, the temporal synchronization assumption brings new characteristics to person identification: 1) 3D human pose of the same person is identical in the same frame across the videos; 2) human motion of the same person is also consistent in 3D space in different videos. Zheng *et al.* [2] adapt the method in [7] to estimate 3D human pose and use the pose distance as a matching metric, but resulting in unsatisfactory CVPI performance. Recently, Zheng *et al.* [1] train a network to learn view-invariant features from optical flow. Then the Euclidean distance between these features is regarded as a metric to select the matching video. The combination of appearance and motion feature matching method does lead to much better CVPI accuracy, but the cross-dataset result is still lower than that produced by unsupervised methods.

Also related to this paper is person re-identification, which aims to match persons captured by non-overlap cameras. The works on person re-identification can be roughly divided into two kinds. The first kind focuses on how to construct discriminative feature [8]–[17]. For example, Li *et al.* [8] model the interactions between persons in two different views by learning a cross-view projective dictionary. To further improve its representation power, Li *et al.* [9] introduce multi-level features

to this method. Due to its high effectiveness, it was applied to solve the divergence of image resolution caused by different camera positions [14]. Dai *et al.* [15] developed a feature transformation method based on unified matrix factorization to improve the representation ability of hand-crafted features. Chen *et al.* [16] proposed a fast re-identification method by jointly learning a subspace projection and a binary coding scheme. The other kind is focused on learning an effective distance metric [18]–[21]. Wang *et al.* [21] propose a data-driven metric, which re-exploits the relationship between a query-gallery pair in training data to adjust the learned general metric. Ye *et al.* [22] utilize the similarity and dissimilarity relationships to optimize the original ranking results. More recent works on person re-identification include the training of an end-to-end CNN model to learn feature and metric at the same time [23], [24] and the semi-supervised or unsupervised person re-identification [25]–[27] to reduce the amount of needed data labeling. An Adaptive Ranking Support Vector Machines method is developed to deal with person label deficiency under target cameras in [25]. Ye *et al.* [26] design a dynamic graph matching method to estimate cross-camera labels. This unsupervised method obtains competitive performance against the fully supervised baselines.

Compared with conventional person re-identification, the CVPI problem focuses on associating persons captured by synchronized wearable cameras. Due to continuous moving of wearable cameras, it is very difficult to calibrate them accurately, which may lead to the failure of methods based on prior camera calibration. We need to design new algorithm for this problem. On the other hand, the above methods try to learn better features or distance metrics from appearance cue, which can be adapted for appearance-based CVPI and further combined with our pose-based method.

B. Human Pose Estimation

Like many other tasks in computer vision, convolutional neural network (CNN) based HPE methods have achieved much better performance than traditional methods [28]. As a fundamental step, 2D HPE from one image has achieved high accuracy in various scenes [29]–[31]. For example, Chu *et al.* [31] achieve 91.5% score in PCKh on MPII dataset [29]. In contrast, the accuracy of 3D HPE is still far from satisfactory [32]. A simple idea is to train a CNN to regress joint locations directly [33]. After this, many improvements have been proposed, such as adding viewpoint prediction [34], enforcing structural constraints [35] and fusing 2D and 3D information [36], [37]. These algorithms are mainly developed for controlled lab environments [4], [38]. Their performances cannot be preserved when the images/videos are taken in outdoor environments or in the wild. To alleviate this problem, two-step approaches are proposed [5], [39], [40]. The first step is to estimate 2D joint heat-maps, which can benefit from existing 2D HPE methods [30], [41]. Then, 3D locations of joints are regressed from the estimated heat-maps or 2D locations, which can be trained by only using annotated images captured in the lab [42]. These methods achieve better performance in the wild environment, but the inconsistency of joints localization accuracy still exits, which will severely affect the CVPI results based on matching 3D human poses.

Our goal is to improve CVPI performance by introducing a confidence metric to each joint in the estimated 3D human pose instead of developing a new 3D HPE method. In this paper, we select one recent 3D HPE algorithm proposed by Zhou *et al.* [40] to derive the confidence on each joint for CVPI. However, the proposed approach can be easily applied to other 3D HPE methods.

C. Confidence Metric

Although significant advance has been achieved in these years, computer vision algorithms are still unreliable and even fail sometimes due to the large variation of environments [43]. Therefore, it is important to own the ability of self-evaluation or introspection [44]–[46]. Confidence metric, which reflects the certainty of an algorithm on its output, is a kind of self-evaluation ability. Confidence metric of methods is crucial if human/system plans to take action based on their output. If we know it will fail, we can alleviate the bad influence by taking action in advance. Generally, there exist two kinds of definitions for confidence: (1) confidence of a model [47]–[49], (2) confidence of a model's prediction on a single example [44], [50]. The former is a statistic analysis of a model's results on a sample set, while the latter is to predict the confidence of an individual instance. Our proposed confidence metric belongs to the latter kind. There have been existing works on confidence analysis for human pose estimation [50], [51], face recognition [52], and other related topics [53], [54]. In [50], a human pose evaluator is trained to predict whether the algorithm returns a correct result for a new test data. Besides, Gal and Ghahramani [55] interpret dropout using Bayesian approximation to model the uncertainty, which has been applied in active learning [56] and scene understanding [57].

Different from analyzing the confidence of the entire pose, we are deriving a confidence metric for every joint's location. A related work is [58], which considers reliability of joints for posture classification. However, our work focuses on the reliability of 3D pose estimation procedure from appearance instead of the pose obtained by Kinect. Moreover, we use the introduced confidence metric to compute confidence-weighted pose distance for CVPI.

III. Cross-View Person Identification

In this paper, we propose to accomplish CVPI by matching the 3D human poses extracted from different videos. Considering the high inaccuracy and inconsistency of 3D human pose estimation, we develop a confidence metric to 3D HPE. Specifically, the confidence metric measures the certainty of each joint's location predicted by a 3D HPE method. Then, this confidence metric is combined with the inaccurately estimated 3D human pose to compute the distance between a pair of videos. As shown in Fig. 2, our method consists of three components: 3D human pose estimation from videos, confidence metric computation for each joint, and video distance

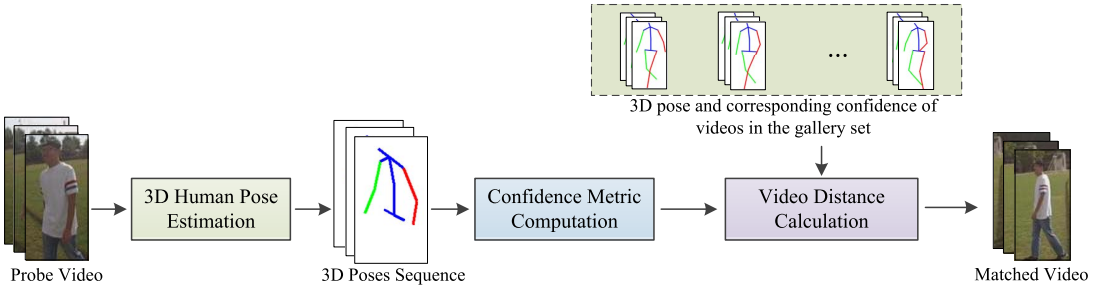


Fig. 2. An illustration of the proposed method for CVPI. First, we estimate 3D pose from the probe video. Then we compute its confidence metric. We apply the same algorithms to estimate the 3D pose and the confidence metric for each video in the gallery set. Finally, we calculate the confidence-weighted pose distance between the probe video and each gallery video. The video in the gallery set with the minimal distance is taken as the matched one.

calculation for CVPI. The first two parts aim to estimate 3D human pose and corresponding confidence metric from videos, which are then combined in the last component to calculate pose-based distance between two videos. For a probe video, the matched video is the one with the smallest pose distance in the gallery dataset. This section will briefly introduce the distance calculation component and the remaining parts will be described in the following two sections.

Before pose distance calculation, the pose should be normalized into the same view and scale. Our normalization includes three steps: rescale each limb′s length to the average of corresponding limb in the training set; translate the pelvis to the origin of axis; rotate the zenith and azimuthal of torso (the segment between pelvis and spine) to a constant angle since this body part can be assumed to be rigid. After the normalized 3D poses and their confidence metrics are obtained, the distance D_P between two videos is defined as

$$D_P = F(W^1, W^2, \boldsymbol{S}^1, \boldsymbol{S}^2) \tag{1}$$

where $\boldsymbol{S}^1(\boldsymbol{S}^2)$ represents the 3D pose of the first (second) video, whose corresponding confidence metric is $W^1(W^2)$. F is a mapping function which takes poses and confidence metrics as input and outputs the distance between two videos. F can be manually designed or directly learned from the training dataset, which will be detailed in Section VI.

Further, we can integrate the pose-based CVPI method with appearance- and motion-based CVPI methods by

$$D = D_M + \alpha D_A + \beta D_P \tag{2}$$

where D is the fused distance, D_M, D_A and D_P are the matching distances computed by motion, appearance and pose respectively. α, β are coefficients to balance the different value ranges of the three distances. The selection of their values will given in the experimental Section. Here, we use the method in [1] and [59] to compute the motion and appearance distances respectively. Specifically, for each video, two feature vectors are derived to represent the motion or appearance of the person respectively. Then, the Euclidean distance between these derived motion (appearance) feature vectors are taken as the matching distance $D_M(D_A)$ between two videos.

IV. 3D Human Pose Estimation

As stated above, we choose the 3D human pose estimation method in [40] to develop the proposed method of confidence estimation and pose-based CVPI. The selected pose estimation procedure contains two steps: 2D heat-maps estimation and 3D pose recovery from 2D heat-maps. The original method assumes all human joints are captured by the camera, which is not true in the videos captured by wearable cameras. An adaption is proposed to handle the case of varying number of captured joints. In the following, we first review the two steps in [40] and then introduce our adaption.

A. 2D Heat-Maps Estimation

Performed on each frame independently, the first step in [40] is to estimate 2D heat-maps of all joints of a video. On each frame, it outputs J heat-maps, each of which is a likelihood of the corresponding human joint over every image coordinate. We replace the original 2D heat-maps estimation component in [40] with the stacked hourglass network architecture [30] due to its great performance. This network is trained by minimizing the following Euclidean distance

$$L_{2D} = \frac{1}{J}\sum_{j=1}^{J} \|Y_j - Y'_j\| \tag{3}$$

where Y_j and Y'_j are the predicted heat-map and corresponding ground-truth for j-th joint respectively, and J is the number of human joints. For each joint, its 2D location is the coordinate of the peak value in its heat-map. Refer to [30] for more details. We directly adopt their released network model and parameters trained on MPII dataset [29], which contains more than 28 000 variable human poses in the wild environment. Due to this large-scale dataset and powerful network architecture, this model generates feasible 2D pose in the videos taken by wearable cameras.

B. 3D Human Pose Estimation Using 2D Heat-Maps

After getting 2D heat-maps of all the frames of a video, Zhou *et al.* [40] recover the 3D human poses via penalized maximum likelihood estimation (MLE). There exist two different cases: 1) 2D poses are provided and 2) 2D heat-maps instead of 2D poses are given. If 2D poses $\boldsymbol{P}$ of a sequence

are given, the MLE for recovering the 3D pose parameter θ is defined as following

$$\theta^* = \arg\max_{\theta} \ln \Pr(\boldsymbol{P}|\theta) - \mathcal{R}(\theta), \tag{4}$$

where $\boldsymbol{P} = \{\boldsymbol{P}_t\}$ is the set of 2D poses, $\Pr(\boldsymbol{P}|\theta)$ is the conditional distribution of 2D pose given 3D pose, $\mathcal{R}(\theta)$ is the prior, which will be detailed later.

Considering the specific form of the conditional distribution $\Pr(\boldsymbol{P}|\theta)$, Eq. (4) can be converted to minimize the following function

$$L(\theta; \boldsymbol{P}) = \frac{\nu}{2} \sum_{t=1}^{n} \|\boldsymbol{P}_t - \boldsymbol{R}_t \sum_{i=1}^{k} c_{it} \boldsymbol{B}_i - \boldsymbol{T}_t \boldsymbol{1}^T\|_F^2 + \mathcal{R}(\theta) \tag{5}$$

where $\boldsymbol{P}_t \in R^{2\times J}$ is the 2D locations of joints at frame t, c_{it} is coefficient of the i-th basis pose $\boldsymbol{B}_i \in \boldsymbol{B}$ at frame t. $\boldsymbol{B} = \{\boldsymbol{B}_1, \cdots \boldsymbol{B}_k\} \subset R^{3\times J}$ is the 3D pose dictionary which concisely summarizes the pose variation [60]. Each element $\boldsymbol{B}_i$ is a 3D pose which represents a typical configuration of the joints location. Currently, $\boldsymbol{B}$ is learned from 3D pose of walking videos in Human 3.6M dataset. $\boldsymbol{R}_t \in R^{2\times 3}$ and $\boldsymbol{T}_t \in R^2$ denote the camera rotation and translation respectively. $\|\cdot\|_F$ represents the Frobenius norm. Here, to satisfy the dimension requirement of mathematical operation, we use a row vector $\boldsymbol{1}^T$ to replicate $\boldsymbol{T}_t$ in column aspect. Its specific length is determined according to actual need. For notational convenience, we use $\boldsymbol{C} = \{c_{it}\}$, $\boldsymbol{R} = \{\boldsymbol{R}_t\}$ and $\boldsymbol{T} = \{\boldsymbol{T}_t\}$ to represent the set of parameters in all frames of a video. Finally, all these parameters are denoted as $\theta = \{\boldsymbol{C}, \boldsymbol{R}, \boldsymbol{T}\}$. To improve the temporal smoothness of the estimated 3D pose and the sparsity of the recovery process, the prior on θ is defined as

$$\mathcal{R}(\theta) = \mu_1 \|\boldsymbol{C}\|_1 + \frac{\mu_2}{2} \|\nabla_t \boldsymbol{C}\|_F^2 + \frac{\mu_3}{2} \|\nabla_t \boldsymbol{R}\|_F^2 \tag{6}$$

where $\|\cdot\|_1$ denotes the ℓ_1-norm, ∇_t is the discrete temporal derivative operator, μ_1, μ_2 and μ_3 are the balance coefficients for different terms.

The problem in Eq. (5) is a non-convex problem with respect to $\theta = \{\boldsymbol{C}, \boldsymbol{R}, \boldsymbol{T}\}$. It is solved via block coordinate descent [61], i.e., alternately updating one of $\boldsymbol{C}$, $\boldsymbol{R}$, or $\boldsymbol{T}$ while fixing the other two. As stated in [40] and [61], this algorithm is guaranteed to converge since the objective function in (5) is non-increasing with respect to one parameter when fixing others. Refer to [40] for more details.

If only 2D heat-maps are given, like our situation, an Expectation-Maximization algorithm is used. First, the expectation of 2D pose is calculated given 2D heat-maps and current estimated 3D pose. Second, this expectation is used in the minimization of Eq. (5). Through iterations over these two steps, the 3D pose can be recovered. In our experiment, the optimization algorithm usually converges in 10 iterations with CPU time less than 80s for a sequence of 120 frame on a two Intel E5-2620 2.4G CPU workstation.

After the 3D pose parameters θ of a sequence are obtained, the final 3D pose at frame t can be represented as

$$\boldsymbol{S}_t = \sum_{i=1}^{k} c_{it} \boldsymbol{B}_i. \tag{7}$$

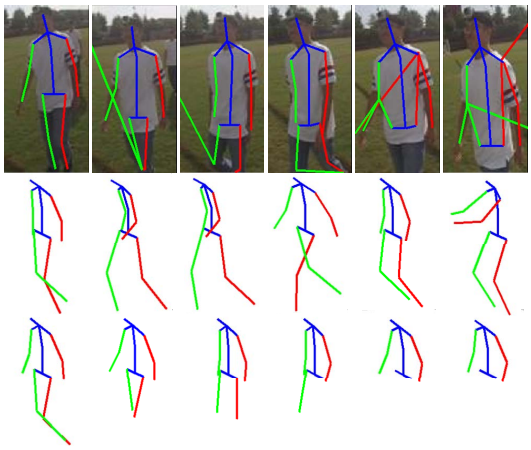


Fig. 3. An illustration of 3D HPE results. Top: original image sequence with estimated 2D pose; Middle and Bottom: estimated 3D human pose without and with the adaption to handle missing body parts. Green lines represent left limbs while the red lines represent the right limbs.

Since this 3D pose is rebuilt with a 3D pose dictionary learned from a real human pose dataset, it is more likely for the recovered 3D pose to satisfy the structure constraints of human body.

C. Adaption to Handle Missing Body Parts

In [40], an important assumption is that all human joints in all frames are captured by the camera. In practice, some human joints may not be viewable in some frames of a video due to view-angle changes and occlusions. This joint missing situation will be more common in videos captured by wearable cameras because of larger view-angle changes. In this case, the 2D HPE algorithm introduced in Section IV-A may return incorrect locations for these missing joints, as shown in the top row of Fig. 3. Such incorrect joint locations may violate the structure constraint of body parts and prevent from rebuilding the correct 3D pose using 3D pose dictionary. Even if an eclectic 3D pose is recovered, it will not conform to the true 3D pose in the image, as shown in the second row of Fig. 3. Using these false 3D poses may seriously hurt the performance of CVPI based on 3D human pose matching.

To solve the above problem, we adapt the original 3D pose estimation method by introducing the concept of validity, which describes the correctness of estimated 2D location of each joint in every frame of a video. First, we assign a binary validity label to every joint. Its definition is based on a common fact – If the returned heat-map of a joint is incorrect, its peak value is always much smaller than that of correct heat-maps in CNN-based 2D HPE algorithms. So a binary validity label is assigned to every joint by comparing the peak value of its heat-map with a fixed threshold. Namely, the label is 1 if the peak value is larger than this threshold, and 0 otherwise. In our experiments, this threshold is 0.1, which is selected through experiments on SEQ1 and is directly used to all datasets. The influence of this threshold will be discussed in the experiment section. Then, we add the validity label into Eq. (5),

leading to

$$L(\theta; \boldsymbol{P}) = \frac{\nu}{2}\sum_{t=1}^{n}\|(\mathbf{1}\phi_t)\circ(\boldsymbol{P}_t-\boldsymbol{R}_t\sum_{i=1}^{k}c_{it}\boldsymbol{B}_i-\boldsymbol{T}_t\mathbf{1}^T)\|_F^2+\mathcal{R}(\theta) \tag{8}$$

where $\phi_t \in R^{1\times J}$ is the validity label of all joints at time t, $\circ$ denotes the element-wise multiplication of two matrices, i.e. Hadamard multiplication. Eq. (8) can be rewritten as

$$L(\theta; \boldsymbol{P}) = \frac{\nu}{2}\sum_{t=1}^{n}\|(\mathbf{1}\phi_t)\circ\boldsymbol{P}_t-\boldsymbol{R}_t\sum_{i=1}^{k}c_{it}((\mathbf{1}\phi_t)\circ\boldsymbol{B}_i) -(\mathbf{1}\phi_t)\circ(\boldsymbol{T}_t\mathbf{1}^T)\|_F^2+\mathcal{R}(\theta). \tag{9}$$

As in [40], Eq. (9) can be minimized by updating one of $\boldsymbol{C}$, $\boldsymbol{R}$ or $\boldsymbol{T}$ alternately while fixing the other two. The first term $(\mathbf{1}\phi_t)\circ\boldsymbol{P}_t$ in Eq. (9) represents the valid 2D locations of joints at frame t. The term $(\mathbf{1}\phi_t)\circ\boldsymbol{B}_i$ denotes the corresponding 3D pose dictionary. These two terms can be calculated in advance. Therefore, the iterative algorithm for minimizing Eq. (9) is the same as that for minimizing Eq. (5) in the original method except for changing the dictionary and 2D pose using the validity label in advance. In detail, for each frame of a sequence, we first calculate the valid 2D poses and specified 3D pose dictionary using the validity label. Then, the adapted 2D pose and dictionary are used to replace the $\boldsymbol{P}_t$ and $\boldsymbol{B}_i$ in the original iterative algorithm. Other things, like the value of parameters, are kept the same as in [40]. Sample results of our adapted HPE algorithm are shown in the bottom row of Fig. 3, where we just show the locations of valid joints.

V. Confidence Metric to 3D HPE

This section introduces the confidence metric 3D pose estimation at each joint and this metric can consider three aspects: confidence of 2D HPE, confidence of 3D HPE from 2D heat-maps, and temporal confidence. The first two focus on the partial space while the last one is on the temporal space.

A. Confidence of 2D HPE

The confidence of 2D HPE is the certainty of estimated 2D location of a joint. In CNN-based 2D HPE, a heat-map is defined as a per-pixel likelihood for a joint's location. So we can use the value of a heat-map as the confidence of corresponding 2D location. However, the resolution of heat-maps is a quarter of that of input images, indicating that the value of heat-maps will not be very smooth. To improve robustness, the confidence of 2D location p is defined as a weighted average value of its four neighbors and itself. Supposing the j-th joint is located at location p at time t, its 2D confidence is defined as

$$W_t^{2D}(j) = w_{cent}\times Y_t(p)+(1-w_{cent})\sum_{p'\in N(p)}0.25\times Y_t(p') \tag{10}$$

where $W_t^{2D}(j)\in R$ is the 2D confidence of the j-th joint at time t, $Y_t(p)$ denotes the value of heat-map Y at location p at time t, $N(p)$ represents the four neighbors of p. w_{cent} is the weight of center pixel of the local region, which is empirically selected to be 0.5 based on SEQ1 and then applied to all the datasets. Such 2D confidence reflects the certainty of the map when visual appearance is projected to 2D locations.

B. Confidence of 3D HPE

For a fixed 2D pose, there may exist multiple possible 3D poses. As a result, 3D HPE is generally formulated as a selection process which aims to find the best matching 3D pose given a 2D pose sequence. Considering this, confidence of 3D HPE can be defined as the matching level between the recovered 3D pose and the estimated 2D pose, which is related to the 3D HPE algorithm.

Based on the objective function (9), our current 3D HPE confidence is inversely proportional to the distance between the 2D pose estimated from images and the 2D pose projected from 3D pose, i.e.,

$$W_t^{3D}(j) = -\|(\mathbf{1}\phi_t(j))\circ[\boldsymbol{P}_t(j)-\boldsymbol{R}_t\boldsymbol{S}_t(j)-\boldsymbol{T}_t(j)]\|_F^2 \tag{11}$$

where $W_t^{3D}(j)\in R$ is the 3D confidence of the j-th joint at time t. Obviously, the 3D confidence for joints with invalid location is zero, which means no impact of these incorrect joint's location on pose matching.

C. Temporal Confidence

The above two kinds of confidence focus on spatial space. To model the consistency of joint locations over time, we design a temporal confidence, which describes the smoothness of a joint's location over time. As a result, the temporal confidence is defined as the distance between the joints' locations in adjacent frames

$$W_t^T(j) = -\phi_t(j)\phi_{t-1}(j)\|\boldsymbol{S}_t(j)-\boldsymbol{S}_{t-1}(j)\|_2 \tag{12}$$

where $W_t^T(j)\in R$ is the temporal confidence of the j-th joint at time t, whose 3D location is $\boldsymbol{S}_t(j)$, and ϕ_t is used to remove the impact of invalid joint location. According to Eq. (12), a sudden change of a joint's location means lower confidence, which conforms to the motion pattern of human. In the experiment, we use the projected 2D pose $\boldsymbol{R}_t\boldsymbol{S}_t(j)+\boldsymbol{T}_t(j)$ to replace the 3D pose $\boldsymbol{S}_t(j)$ since the depth estimation is not very accurate. This replacement can improve the performance by 1% to 2%.

VI. Video Distance Computation

Now we have the 3D poses along a video and their corresponding confidence metrics. To accomplish CVPI, we need to compute pose-based video distance using Eq. (1). In this section, we will introduce the two ways for constructing function F in (1) – one is based on multiplication and the other is based on supervised deep learning.

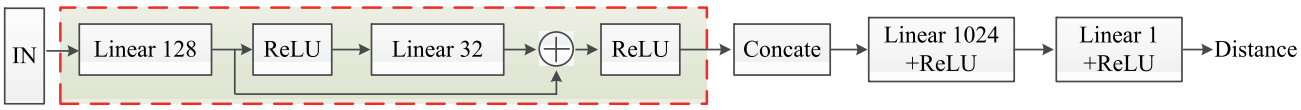


Fig. 4. Network architecture for computing the distance of a pair of videos. A input vector is built to represent the pose distance and confidence metrics of two corresponding frames. The vectors of all the frames are processed by the module in red box simultaneously. Then the outputs are concatenated and fed to the remaining layers.

A. Multiplication Based Video Distance

Since confidence metrics measure the reliability of estimated 3D locations of human joints, we can simply use them as weights. Specifically, we can directly multiply the confidence metrics of each joint and its distance in two corresponding frames. Then we can obtain the pose-based distance between two videos by adding the weighted distances of all the joints over all the frames. However, the value ranges of the three confidence metrics are different in our experiment. W^{2D} is a probability value in [0, 1]. W^{3D} represents the distance between the observed 2D pose and the projected 2D pose, which is in the range of $[-30, 0]$. W^T is the variation of joint locations over time, which varies in [-10, 0]. Obviously, it is improper to multiply them directly. They should be transformed into the same range. Considering this, we fuse the three confidence metrics by

$$W_t(j) = \phi_t(j) \times W_t^{2D}(j) \times e^{\alpha^{3D} W_t^{3D}(j)} \times e^{\alpha^T W_t^T(j)} \quad (13)$$

where $W_t(j) \in R$ is the final confidence of the j-th joint at time t, $\alpha^{3D}, \alpha^T \in R$ are the coefficients for different confidence metrics. These two coefficients and exponent functions are used to transform the values of temporal and 3D HPE confidences to the range of [0, 1], which can then multiply with the 2D HPE confidence directly. Note that we use the same coefficients for all the joints in Eq. (13). In the experiments, these two coefficients are selected by using the training dataset of SEQ1 - α^{3D} and α^T are set to 0.2 and 0.5 respectively for all the experiments.

After obtaining the fused confidence, we can rewrite the video distance in Eq. (1) as

$$D_P = \sum_t \sum_{j=1}^{J} min(W_t^1(j), W_t^2(j)) \| (\boldsymbol{S}_t^1(j) - \boldsymbol{S}_t^2(j)) \| \quad (14)$$

where t represents the index of video frame. For each joint, we compare its confidences in the two considered videos and select the smaller one as the final weight. The final distance D_P between a pair of videos is the sum of pose distance over all the frames. According to Eqs. (13) and (14), only the subset of joints that are captured in both two corresponding frames are used to compute the distance of a pair of videos. If a joint is detected in one video but not the other, this joint will not be considered in computing video distance.

B. Deep Learning Based Video Distance

The fusing method in the above subsection is handcrafted, which may not be optimal and affect the performance of CVPI. To address this problem, we employ a CNN to more optimally fuse the poses and corresponding confidence metrics to compute the video distance by learning from the dataset directly. In other word, F in Eq. (1) is a CNN. Specifically, it takes pose difference of two videos and their corresponding confidence metrics as input and outputs the pose-based distance. A naive idea is to directly concatenate the pose differences and confidence metrics in all frame to build the input of CNN. However, this will result in an input vector with very large dimension. In our experiment, the dimension of pose difference and confidence metrics of a frame is 80. If we concatenate the pose differences and confidence metrics of all 120 frames, the length of input vector would be 9600. Thus, the number of CNN parameters will be very large, which means more training examples are needed to avoid over-fitting. On the other hand, the operation of each frame should be the same if we ignore the temporal relationship of frames. So, we propose to apply the same operation to every frame and then concatenate the output features. The resulting network architecture is shown in Fig 4. The modules bounded by a red box is operated on every frame independently. Then the output features are concatenated and fed to the remaining layers as shown in Fig. 4 to merge the information of all the frames to generate the final distance. The neural network in Fig. 4 contains one residual module, whose effectiveness has been verified in [62].

According to the definition of CVPI, two videos capturing the same person constitute a positive example pair while two other arbitrary videos from different cameras make up a negative pair. Therefore, a training example consists of a pair of videos and a binary label, which is 1 for a matching pair and 0 otherwise. For two videos, the input vector x_t of the CNN at frame t is built via the following equation

$$x_t = [\boldsymbol{S}_t^{dif} \;\; W_t^{2D} \;\; W_t^{3D} \;\; W_t^T \;\; \phi_t] \quad (15)$$

where $\boldsymbol{S}_t^{dif}(j) = |\boldsymbol{S}_t^1(j) - \boldsymbol{S}_t^2(j)|$ denotes the 3D location difference of two videos, $\boldsymbol{S}_t^{dif}$, W_t^{2D}, W_t^{3D}, W_t^T and ϕ_t are the concatenation of corresponding pose difference, confidence metrics and validity labels of all the joints, respectively. As in Eq. (14), the smaller confidence value of the two videos are selected as the final confidence in Eq. (15). Element-wise operation is used to get the final validity label. So the input feature x consists of pose difference, confidence metrics and the validity labels. To identify the true matching video from a gallery set, the distance of the positive pair should be smaller than that of the negative pairs. As a result, we use the contrast loss function to learn the parameters of CNN, i.e.

$$L_{CNN} = \frac{1}{N} \sum_{i=1}^{N} 2 y_i d_i^2 + (1 - y_i) max(m - d_i, 0)^2 \quad (16)$$

Algorithm 1 CNN Training Algorithm

Input: learning rate; total epoch number N_e; number of videos in a camera N_v; number of hard negative examples N_h; pose, confidence metrics and validity label of each video
Output: Network Parameters
1: Initialize network parameters
2: Build input feature vector for positive examples
3: **for** $iter = 1 : N_e$ **do**
4: //Mining hard negative examples
5: **for** $iv = 1 : N_v$ **do**
6: Compute the distance between iv-th video and all videos except for the iv-th from another camera
7: Select the videos with top N_h largest distance
8: Build input vector for negative video pairs
9: **end for**
10: Train parameters of the network
11: **end for**
12: **return** Network Parameters

where y_i is the label of i-th example, d_i is the corresponding distance computed by the CNN, m is a margin for video distance. In our experiment, m is set to 100. According to Eq. (16), the distance for positive examples should be close to 0 while the distance of negative examples should be larger than the margin. This is just the requirement of CVPI.

According to the definition of examples, the ratio between the number of positive examples and negative examples will be $\frac{1}{N-1}$ if there are N videos under a single camera. This severe data imbalance will hurt the performance as shown in many CNN literatures. A natural choice is to select some hard negative examples. It is obvious that a fixed selection of negative examples will not reflect the real distribution of entire examples. We utilize a dynamic hard negative example mining strategy which iteratively mines hard negative examples in every training epoch of the CNN. The final training algorithm of the proposed network is shown in Algorithm 1, in which lines 5 to 8 describes the hard negative example mining strategy. The algorithm inputs are some hyper-parameters, person identity of each video, estimated pose and corresponding confidence metrics. The output is the learned parameters of CNN. First, we build the input vector according to Eq. (15). Then we use current CNN parameters to compute the distance of each pair of videos in line 6. Based on the obtained distance, just several most similar video pairs are selected as negative examples. Next, the positive examples and the selected negative examples are used to update the CNN parameters. By only selecting several hardest negative examples, we can reduce the computation time largely. For inference, we execute line 5 to 6 of Alg. 1 on test videos and store the distance values. For a query video, the video with the smallest distance in the gallery dataset is taken as the matched video.

Implementation Details: Since training examples in current CVPI datasets are not enough, learning parameters only on these datasets will lead to over-fitting. We synthesize 978 pairs of sequences with 120 frames from Human 3.6M dataset [4], where four cameras capture the same person simultaneously from different views. Specifically, only two temporal synchronized videos from two cameras are regarded as a true matching pair, otherwise they are not a matching pair. We employ the Matconvnet ver1.0-beta25 [63] to implement the CNN architecture in Fig. 4. The Rmsprop algorithm [64] is used for learning the parameters. In the experiment, we first train the network on synthesized sequences, then fine-tune the parameters on training set of each dataset.

VII. Experiments

In this section, we first describe the datasets and evaluation metric. Then we discuss the influence of network configuration on CVPI performance and finally report the experimental results.

A. Datasets and Evaluation Metric

The proposed method is evaluated on three datasets: SEQ1, SEQ2 [2] and SYN [1], all of which are human-walking videos. These datasets are taken by two temporally synchronized GoPro cameras with different views. As a result, they consist of video pairs, each of which actually captures the same walking subject from different views. In all datasets, the GoPro cameras are mounted on the wearers' heads. The length of each video is 120 frames. All subjects wear similar clothes - T-shirts and blue jeans in SEQ1 and SEQ2, dark jackets in SYN. There are 114 and 88 video pairs in SEQ1 and SEQ2, respectively, all of which are performed by 6 subjects in a football field. Besides the people of interest, there are other pedestrians which may cross the people of interest. Together with the camera angle issues, this may make portions of human body invisible in some video frames. SYN contains 208 video pairs performed by 14 subjects near a building. Compared to the first two datasets, SYN has less camera motion. Besides, all the human body parts in SYN are visible. For fair comparison with previous methods, the frame resolution of all the videos is normalized to 64×128. Note that video datasets that are widely used for evaluating person Re-ID, such as PRID 2011, ILIDS-VID, MARS and SDU-VID are usually taken by multiple cameras without spatial overlap and temporal synchronization. Therefore, they do not satisfy the requirement for the proposed CVPI task and are not suitable for evaluating the CVPI algorithms. Besides, more complex and uncommon person actions in a video actually provide more pose information for CVPI. In our experiments, all our test videos only contain the most common action of person walking, which actually increases the difficulty of CVPI by showing relatively simple and similar poses for different people.

Following the evaluation protocol in [1] and [2], we randomly split the dataset into two equal-size parts for training and testing respectively. One camera's videos are probe set and the others are gallery set. For multiplication based video distance, we select the coefficients in Eq. (13) on the training set of SEQ1. Then, these selected parameters are used to evaluate our method on testing sets of all three datasets. For deep learning based video distance, the CNN parameters are learned on synthesized sequences and fine-tuned on the

TABLE I
CMC PERFORMANCE OF SEQ1 USING DIFFERENT WEIGHT VALUES OF 2D CONFIDENCE

w_{cent}	0.0	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1.0
Rank 1	29.12	29.30	29.30	29.30	30.00	30.00	30.00	29.77	29.77	29.42	29.42
Rank 5	67.02	66.84	66.67	66.84	66.84	67.02	67.02	67.02	66.84	66.84	66.67
Rank 10	78.77	79.12	79.12	79.12	79.12	79.30	79.30	79.47	78.95	78.77	78.77
Rank 20	88.77	88.77	88.77	88.77	88.77	88.77	88.60	88.60	88.25	88.60	88.60
Average	65.92	66.01	65.97	66.01	66.18	66.27	66.23	66.22	65.95	65.91	65.87

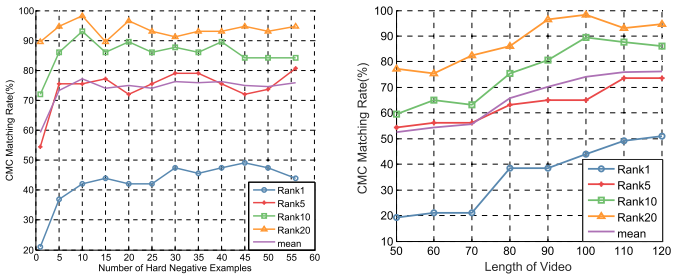
training set of each dataset. Then we test the learned network on corresponding testing set and report the results. Like previous methods [1], [2], we use the Cumulative Matching Characteristics (CMC) ranks [65] as our metric for CVPI performance evaluation. CMC at rank k means the predicted results are right if the correct matching video is within the k-best results. We choose k to be 1, 5, 10 and 20, respectively in our experiments. To reduce the impact of random split on performance, we execute the random partitioning for ten times and report the average CMC score unless otherwise stated.

Based on the code released by Newell *et al.* [30] and Zhou *et al.* [60], we implement our model with Matlab except for the 2D heat-maps estimation module, which is implemented with torch. On a two Intel E5-2620 2.4G CPU workstation with an NVIDIA Tesla K40m GPU, it needs 6.52 seconds to estimate 2D heat-maps of joints for a sequence with 120 frames. 3D human pose estimation from 2D heat-maps takes about 96.44 seconds. Confidence metric computation needs 15 ms since they just involve simple arithmetic operation. It needs about 10 hours for training the CNN parameters in deep learning based distance computation method. The computation of distance between two videos takes 0.4ms and 10ms for multiplication based and deep learning based methods, respectively. Since the 3D human pose and confidence metric of each gallery video can be computed in advance, most time is spent on the pose estimation and confidence metric computation for the probe video. After that, the matching process will take $10N$ or $0.4N$ ms if there are N gallery videos. The slowest part is 3D HPE, which could be replaced with the newest CNN-based method to improve the speed and performance.

B. Influence of Parameters

In this subsection, we discuss the influence of some parameters on CVPI performance. All experiments in this subsection are conducted on the SEQ1 dataset. Since the training of CNN takes long time, we just run the deep learning based method using one random split for the first two parameters. For the remaining parameters, we use multiplication based distance to show their influence.

1) Influence of the Number of Hard Negative Examples: We propose a dynamic hard negative example mining strategy, which includes a parameter N_h representing the number of selected hard negative examples. To investigate the influence of this parameter, we run multiple experiments with different values of N_h. The results are shown in Fig. 5(a). In this figure, the most important CMC Rank 1 performance increases when



(a) (b)

Fig. 5. CMC performance on SEQ1 using different configurations. (a) Influence of negative examples. (b) Influence of video length.

TABLE II
CMC ACCURACY USING DIFFERENT THRESHOLDS FOR VALIDITY LABEL

Threshold	0	0.1	0.2	0.3
Rank 1	28.42	**30.00**	29.12	28.73
Rank 5	58.60	67.02	**67.19**	65.96
Rank 10	70.53	79.30	78.60	**81.05**
Rank 20	83.16	**88.77**	88.25	85.53
Average	60.18	**66.27**	65.79	65.32

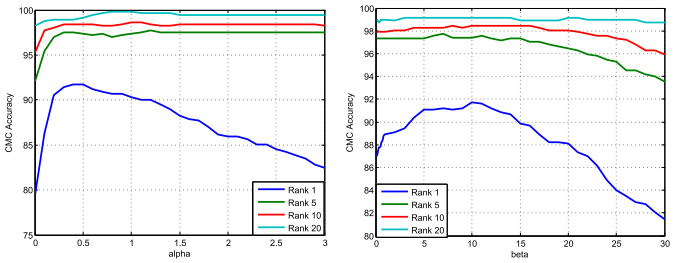
N_h is smaller than 45. It stops growing after N_h reaches 45. When N_h equals 56 (the number of all training negative examples in SEQ1), all negative examples in SEQ1 are used. In other word, the dynamic example mining strategy is not used. As expected, the performance is worse than that of using dynamic negative example mining strategy. The average performance shows similar trend except that the inflection point is 10. In the following experiments, N_h is set to 45 for SEQ1 and SYN, 35 for SEQ2 since there are fewer examples in SEQ2.

2) Influence of Video Length: Then, we discuss the influence of video length on CVPI performance. We just change the length of input videos and keep all the other settings the same. The results of new models are shown in Fig. 5(b). The performance increases when the video length increases from 50 to 120. This is a natural phenomenon since more frames mean richer information. In the following sections, we always set video length to be 120.

3) Influence of the Weight of 2D Confidence: The definition of 2D confidence in Eq. (10) contains a parameter w_{cent}, which denotes the weight of center pixel of the local region. To show its influence on the CVPI performance, we conducted experiments on SEQ1 using different values for w_{cent} and the results are shown in Table I. From this table, we can see that

TABLE III
CMC PERFORMANCE OF THE MULTIPLICATION-BASED CVPI WHEN USING DIFFERENT CONFIDENCE METRICS

dataset	SEQ1				SEQ2				SYN			
CMC Rank	1	5	10	20	1	5	10	20	1	5	10	20
Pose	12.81	38.60	54.56	65.61	18.64	42.27	55.00	66.14	52.50	73.17	81.15	89.90
Pose+2D	25.26	**68.95**	78.77	**90.53**	**30.68**	62.27	73.86	87.27	63.94	**87.98**	92.79	**95.58**
Pose+3D	22.98	66.49	77.19	86.32	24.77	54.09	69.09	82.05	59.42	82.31	91.83	94.90
Pose+T	22.46	59.12	75.79	87.02	28.86	55.91	70.68	82.95	57.02	81.93	89.62	94.13
Pose+2D+T	29.42	67.19	76.84	90.35	29.09	63.64	75.91	**87.95**	**65.38**	87.21	92.50	95.29
Pose+3D+T	25.61	65.96	75.79	87.72	30.45	60.00	72.95	83.86	62.50	86.44	91.92	95.00
Pose+2D+3D	26.14	68.60	78.42	89.65	30.45	63.64	75.68	87.27	64.04	86.44	**92.88**	95.77
CPose	**30.00**	67.02	**79.30**	88.77	29.55	**65.68**	**76.36**	**87.95**	63.65	85.96	92.31	**95.58**



(a) (b)

Fig. 6. CMC performance on SEQ1 using different α and β. (a) Accuracy curve with different α when $\beta = 10$. (b) Accuracy curve with different β when $\alpha = 0.5$.

the best performance is achieved when the value of w_{cent} is 0.5, which we will use in the following experiments..

4) Influence of Threshold for Validity Label: In Section IV-C, a threshold is used to assign a binary validity label. Table II shows the CMC accuracy by using different thresholds for assigning the validity label. We can see that the best performance is achieved when the threshold is set to 0.1 and we choose this threshold value in the following experiments. When the threshold is 0, our adaption is degenerated to the original HPE with much poorer CMC performance. This verifies the effectiveness of our adaption.

C. Effects of Different Confidence Metrics

This section investigates the effects of different confidence metrics on CVPI performance. For simplicity, we just use multiplication based video distance to illustrate the influence. We run multiple experiments with different configurations of confidence metrics, whose results are shown in Table III. In this table, 'Pose' denotes the CVPI performance only using 3D human pose without any confidence. 'Pose+2D', 'Pose+3D' and 'Pose+T' are the performance using 3D pose weighted by the confidence of 2D HPE, the confidence of 3D HPE, and temporal confidence, respectively. 'Pose+2D+T', 'Pose+3D+T', 'Pose+2D+3D' are the performance by using three ways of combining two confidence metrics, respectively. 'CPose' is the performance by fusing all three confidence metrics.

Since some body parts are occluded or missing in the videos of SEQ1 and SEQ2, the CVPI accuracy is much lower than that of SYN. The low accuracy shows it is difficult to accurately estimate 3D human pose from videos captured by wearable cameras. For all the three datasets, adding any one of the three confidence metrics can improve the pose-based CVPI performance substantially. This verifies the combination of the inaccurately estimated pose with a joint-based confidence can boost the CVPI performance. From this table, we also find that the 2D confidence can help improve the CVPI performance more than the other two kinds of confidence. Since 2D pose estimation is the first step of 3D pose estimation and determines the accuracy of estimated 3D pose, the confidence of 2D HPE is more valuable than that of 3D HPE and temporal. In most cases, the method using the fused confidence metric leads to the best performance.

D. Quantitative Comparison

In this section, we compare our method with several other state-of-the-art methods. We use CPose and LPose to denote the proposed multiplication based method and deep learning based method respectively. The compared methods include discriminative video ranking (DVR) [66], 3D pose estimation for person identification (3DHPE) [2], recurrent feature aggregation (RFA) [1], [59] and view invariant features from optical flow (Flow) [1]. RFA [57] use an LSTM network to extract discriminative feature directly from the RGB image sequence, which represents the appearance of person. In these methods, 3DHPE and CPose are unsupervised since they do not need the information of person identity. Other methods need the training dataset to learn the parameters.

Table IV shows the experimental results. Both CPose and LPose achieve much higher performance than DVR and 3DHPE, which validates the effectiveness of the combination of human pose with confidence metrics. Since CPose does not use other large-scale dataset for training, the accuracy is inferior to that of Flow and RFA. Even so, it still outperforms RFA in SYN in term of CMC rank 1. The reason may be that SYN contains less occlusions. Compared with CPose, LPose obtains higher performance in most cases. For example, the CMC rank 1 accuracy gain on SEQ1 is more than 17.54%. The performance gain on SYN is less than that of SEQ1 and SEQ2. The reason might be that the reliability of estimated pose on SYN is much higher than that of SEQ1 and SEQ2 and

TABLE IV
COMPARISON OF THE PROPOSED METHOD WITH STATE-OF-THE-ART METHODS ON SEQ 1, SEQ 2 AND SYN DATASETS IN TERMS OF CMC RANK

dataset	SEQ1				SEQ2				SYN			
CMC Rank	1	5	10	20	1	5	10	20	1	5	10	20
DVR	16.14	50.53	66.84	82.83	11.14	34.09	53.64	77.05	12.69	41.83	59.04	75.87
3DHPE	16.14	50.70	67.02	81.93	17.95	51.82	71.14	89.55	8.65	35.67	50.48	64.52
RFA	68.42	96.84	98.25	99.30	69.77	96.36	98.41	99.32	56.83	92.40	97.02	98.85
Flow	79.82	92.28	95.26	97.54	76.36	87.05	92.73	96.82	72.21	90.00	94.90	98.08
Flow+RFA	87.02	97.37	97.89	98.95	82.05	94.39	96.59	99.32	82.12	98.37	99.33	**100**
CPose	30.00	67.02	79.30	88.77	29.55	65.68	76.36	87.95	63.65	85.96	92.31	95.58
CPose+RFA	84.56	97.19	98.24	99.65	77.05	**98.19**	**99.32**	**100**	84.51	98.72	99.60	**100**
CPose+Flow	85.97	93.86	97.37	98.07	80.45	91.36	95.23	98.64	84.14	95.69	98.44	99.71
CPose+RFA+Flow	**91.75**	97.37	98.42	99.12	86.36	96.13	98.63	99.77	**91.54**	**99.81**	**100**	**100**
LPose	47.54	82.63	88.25	94.21	35.91	71.59	84.77	95.23	63.17	89.15	93.51	96.56
LPose+RFA	86.47	97.74	98.50	99.50	83.44	95.78	98.38	99.68	85.52	98.31	99.58	**100**
LPose+Flow	87.22	96.46	**98.75**	99.25	76.62	93.18	97.40	98.70	81.87	94.11	97.34	99.30
LPose+RFA+Flow	**91.75**	**98.50**	**98.75**	**99.75**	**87.34**	96.10	99.03	99.68	90.31	99.45	99.86	**100**

TABLE V
AVERAGE CMC SCORE OF SEQ1 USING DIFFERENT α AND β

β \ α	0.2	0.3	0.4	0.5	0.6	0.7	0.8
7	96.50	96.45	96.40	96.14	95.57	95.31	95.04
8	96.65	96.66	96.64	96.58	96.45	96.18	95.83
9	96.45	96.49	96.58	96.67	96.67	96.49	96.40
10	96.58	96.49	96.49	96.67	96.67	96.54	96.40
11	96.40	96.54	96.58	96.58	96.67	96.58	96.62
12	96.45	96.45	96.45	96.54	96.62	96.67	96.62
13	96.40	96.54	96.49	96.45	96.40	96.49	96.58

TABLE VI
CROSS-DATA PERFORMANCE IN TERMS OF CMC RANK

Rank	1	5	10	20
3DHPE	17.95	51.82	71.14	**89.55**
RFA	5.00	14.77	32.50	63.63
Flow	11.36	25.00	38.64	63.64
CPose	**29.55**	**65.68**	**76.36**	87.95
LPose	11.36	44.23	53.41	71.59

the benefit of using the confidence metrics on SYN is not that significant.

We further combine the pose-based method, appearance-based and flow-based method using Eq. (2). In Eq. (2), α and β are used to handle the different value ranges of pose distance, appearance distance and motion distance, in case one overly dominates the other two. For example, in our experiment, the values of CPose distance, motion distance and appearance distance are around 2, 22 and 46 respectively. If we directly add them without any weights, the appearance may contribute much more than pose and motion to the final CVPI accuracy.

The value of α and β are selected on SEQ1 and we then use these values for experiments on all the three datasets. For showing the effect of these coefficients on CVPI performance, we conduct experiments on SEQ1 using different values for α and β for combing CPose with appearance and flow. The results are shown in Fig. 6. From this figure, we can see that the CMC accuracy achieves the best when α and β equal 0.5 and 10 respectively. If they deviate too much from these values, the performance will decrease severely. A more detailed CMC accuracy variation is shown in Table V. For saving space, we just give the average scores over CMC Rank 1, 5, 10 and 20. From Table V, we can find that the CVPI performance is not very sensitive when the values of α and β vary in the range of [0.2, 0.8] and [7, 13] respectively. The results of LPose are similar except that the α and β are in different ranges since the value ranges of CPose and LPose are different. As a result, we set α and β to 0.5 and 10 respectively for CPose, and 0.9 and 28 respectively for LPose.

In the bottom eight rows of Table IV, we give the final results for combined methods - combine CPose or LPose with RFA(Flow). CPose and LPose show similar performance. Adding arbitrary kind of pose to Flow (or RFA) leads to better performance than the original Flow (or RFA). Combining pose, Flow and RFA achieves the highest performance in most cases, which verifies that the human poses, although inaccurately estimated, can still complement motion and appearance features for improving CVPI.

E. Cross-Dataset Testing

As in [1], we also compare cross-dataset performance. For fair comparison, we perform this testing on SEQ2 using the parameters fine-tuned on SEQ1 dataset. The results are shown in Table VI. Note that for the two unsupervised methods, the accuracy keeps unchanged. The proposed CPose achieves the best cross-dataset testing performance and the performance gain is over 11% in term of CMC rank 1. Besides, we can see that the two unsupervised methods, 3DHPE and CPose, show much better cross-data testing performance than the appearance or optical-flow based methods. Although SEQ1 and SEQ2 share similar background and subjects, the supervised methods including RFA, Flow and LPose still do not perform well on cross-dataset testing. This may be caused by overfitting in training due to the large number of trained parameters and the small number of training samples. Even so, the pose-based supervised method LPose obtains much better accuracy than Flow and RFA in Rank 5, 10 and 20. This shows that pose is a more robust cue with higher generalization ability.

Fig. 7. Sample matching results. Three columns are from SEQ1, SEQ2 and SYN dataset respectively. Top two rows are correct matching video pairs. One row is probe input video and the other is the returned matching video. The bottom three rows show failure cases. The third row is the probe input video. The fourth row is the returned matching video, which is incorrect, and the last row is the true matching video for the probe in the third row.

Fig. 8. Matching examples with different model settings. The sequences bounded by a red rectangle are the true-matched ones.

F. Qualitative Results

In Fig. 3, we show the 3D human pose estimation results for a video sequence. From this figure, we can find that some body parts of this sequence are not captured by cameras and the number of missing body parts is changing over time. As shown in the top row, the predicted 2D pose are superposed

on the images. We can see that the 2D locations for missing body joints are severely incorrect. The 3D pose obtained by the original method is given in the second row. Due to the influence of the false 2D locations, the 3D locations of visible joints are also incorrect. The last row gives the 3D pose estimated by our adapted method. We can find our adaption can return reasonable pose for visible human body parts at any time, which can be used for pose-based CVPI.

Sample matching results are shown in Fig. 7. The three columns are samples from SEQ1, SEQ2 and SYN respectively. The top two rows are correct matching video pairs. Incorrect matching video pairs are shown in the bottom three rows, where the third row are probe inputs, the fourth row gives the returned false sequences and the true matching sequences are shown in the fifth row. The 3D human movements in the matched video pairs are completely consistent. In the correctly matched videos from SEQ1 and SEQ2, some body parts are missing. Nevertheless, our algorithm still returns the correct matching results. This shows that using some of body parts is sufficient for CVPI. For failure cases, the main reasons include the missing of too many key body parts and the overly large difference of the camera views. For example, the failed matching in SEQ1 in Fig. 7 may be caused by the totally opposite view angles of the true matched video pairs, as indicated in rows 3 and 5 in Fig. 7. As a result, the visible parts in the probe are invisible in the true matching video, which leads to a false matching. Similarly, due to the camera-view difference, many key body parts are occluded in the probe video of SEQ2, which results in a false matching. The false matching video in SYN has very similar movement as the probe video.

To further analyze the effects of different model settings, we show the returned videos by different models in Fig. 8, where the videos bounded by a red rectangle are the true-matching ones. Note that the frame indices of the query video and the returned one are the same. From this figure, we can find that the incorporation of confidence can improve the matching results. In the first example, the returned video by using a single confidence metric is incorrect, but combining all three confidence metrics leads to a correct CVPI result. In the second example, the video returned by CPose is very similar to the query video, but it is still not the true matching one. However, LPose returns the true video, which illustrates the effectiveness of the proposed deep-learning method for confidence-metric fusing.

VIII. Conclusion

In this paper, we proposed a confidence-weighted human pose matching method for cross-view person identification (CVPI), i.e., identifying the same person from temporally synchronized videos. Considering the high inaccuracy of 3D human pose estimation (HPE), we develop a new metric of confidence to 3D HPE, which measures the estimation confidence of each joint. The proposed confidence metric combines 2D HPE confidence, 3D HPE confidence and temporal confidence. We proposed two methods to combine the inaccurately estimated 3D human pose with the confidence metric for CVPI. We found that the derived confidence metric can promote the pose-based CVPI. Finally, we integrate the estimated pose features into motion and appearance features and found that they can well complement each other and the integration of all three leads to a new state-of-the-art CVPI performance. Currently, each module is trained individually, which may restrict the performance. For the future work, we plan to design an end-to-end CNN, which integrates 3D pose estimation, confidence metrics computation and video distance calculation in a single architecture. Besides, we will construct more datasets with other human actions to better evaluate different methods.

References

[1] K. Zheng *et al.*, "Learning view-invariant features for person identification in temporally synchronized videos taken by wearable cameras," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2017, pp. 2858–2866.

[2] K. Zheng, H. Guo, X. Fan, H. Yu, and S. Wang, "Identifying same persons from temporally synchronized videos taken by multiple wearable cameras," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. Workshop*, Jun./Jul. 2016, pp. 105–113.

[3] G. Pavlakos, X. Zhou, K. G. Derpanis, and K. Daniilidis, "Coarse-to-fine volumetric prediction for single-image 3D human pose," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1263–1272.

[4] C. Ionescu, D. Papava, V. Olaru, and C. Sminchisescu, "Human3.6M: Large scale datasets and predictive methods for 3D human sensing in natural environments," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 7, pp. 1325–1339, Jul. 2014.

[5] X. Zhou, Q. Huang, X. Sun, X. Xue, and Y. Wei, "Towards 3D human pose estimation in the wild: A weakly-supervised approach," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 398–407.

[6] G. Liang, X. Lan, K. Zheng, W. Song, and N. Zheng, "Cross-view person identification by matching human poses estimated with confidence on each body joint," in *Proc. AAAI*, 2018, pp. 7089–7097.

[7] A. Gupta, J. Martinez, J. J. Little, and R. J. Woodham, "3D pose from motion for cross-view action recognition via non-linear circulant temporal encoding," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 2601–2608.

[8] S. Li, M. Shao, and Y. Fu, "Cross-view projective dictionary learning for person re-identification," in *Proc. IJCAI*, 2015, pp. 2155–2161.

[9] S. Li, M. Shao, and Y. Fu, "Person re-identification by cross-view multi-level dictionary learning," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2963–2977, Dec. 2018.

[10] W. Zhang, B. Ma, K. Liu, and R. Huang, "Video-based pedestrian re-identification by adaptive spatio-temporal appearance model," *IEEE Trans. Image Process.*, vol. 26, no. 4, pp. 2042–2054, Apr. 2017.

[11] Y. Yang, L. Wen, S. Lyu, and S. Z. Li, "Unsupervised learning of multi-level descriptors for person re-identification," in *Proc. AAAI*, 2017, pp. 4306–4312.

[12] Y. Suh, J. Wang, S. Tang, T. Mei, and K. M. Lee, "Part-aligned bilinear representations for person re-identification," in *Proc. Eur. Conf. Comput. Vis.*, 2018, pp. 402–419.

[13] Y.-J. Cho and K.-J. Yoon, "PaMM: Pose-aware multi-shot matching for improving person re-identification," *IEEE Trans. Image Process.*, vol. 27, no. 8, pp. 3739–3752, Aug. 2018.

[14] K. Li, Z. Ding, S. Li, and Y. Fu, "Discriminative semi-coupled projective dictionary learning for low-resolution person re-identification," in *Proc. AAAI*, 2018, pp. 2331–2338.

[15] J. Dai, Y. Zhang, H. Lu, and H. Wang, "Cross-view semantic projection learning for person re-identification," *Pattern Recognit.*, vol. 75, pp. 63–76, Mar. 2018.

[16] J. Chen, Y. Wang, J. Qin, L. Liu, and L. Shao, "Fast person re-identification via cross-camera semantic binary transformation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 5330–5339.

[17] Z. Wang *et al.*, "Person reidentification via discrepancy matrix and matrix metric," *IEEE Trans. Cybern.*, vol. 48, no. 10, pp. 3006–3020, Oct. 2017.

[18] C. Sun, D. Wang, and H. Lu, "Person re-identification via distance metric learning with latent variables," *IEEE Trans. Image Process.*, vol. 26, no. 1, pp. 23–34, Jan. 2017.

[19] Y. Yang, Z. Lei, S. Zhang, H. Shi, and S. Z. Li, "Metric embedded discriminative vocabulary learning for high-level person representation," in *Proc. AAAI*, 2016, pp. 3648–3654.

[20] S. Bak and P. Carr, "One-shot metric learning for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 2990–2999.

[21] Z. Wang *et al.*, "Zero-shot person re-identification via cross-view consistency," *IEEE Trans. Multimedia*, vol. 18, no. 2, pp. 260–272, Feb. 2016.

[22] M. Ye *et al.*, "Person reidentification via ranking aggregation of similarity pulling and dissimilarity pushing," *IEEE Trans. Multimedia*, vol. 18, no. 12, pp. 2553–2566, Dec. 2016.

[23] W. Chen, X. Chen, J. Zhang, and K. Huang, "A multi-task deep network for person re-identification," in *Proc. AAAI*, 2017, pp. 3988–3994.

[24] F. Wang, W. Zuo, L. Lin, D. Zhang, and L. Zhang, "Joint learning of single-image and cross-image representations for person re-identification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 1288–1296.

[25] A. J. Ma, J. Li, P. C. Yuen, and P. Li, "Cross-domain person reidentification using domain adaptation ranking SVMs," *IEEE Trans. Image Process.*, vol. 24, no. 5, pp. 1599–1613, May 2015.

[26] M. Ye, A. J. Ma, L. Zheng, J. Li, and P. C. Yuen, "Dynamic label graph matching for unsupervised video re-identification," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 5152–5160.

[27] M. Ye, X. Lan, J. Li, and P. C. Yuen, "Hierarchical discriminative learning for visible thermal person re-identification," in *Proc. AAAI*, 2018, pp. 7501–7508.

[28] W. Zhang, L. Shang, and A. B. Chan, "A robust likelihood function for 3D human pose tracking," *IEEE Trans. Image Process.*, vol. 23, no. 12, pp. 5374–5389, Dec. 2014.

[29] M. Andriluka, L. Pishchulin, P. Gehler, and B. Schiele, "2D human pose estimation: New benchmark and state of the art analysis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3686–3693.

[30] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 483–499.

[31] X. Chu, W. Yang, W. Ouyang, C. Ma, A. L. Yuille, and X. Wang, "Multi-context attention for human pose estimation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 1831–1840.

[32] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5659–5670, Dec. 2015.

[33] S. Li and A. B. Chan, "3D human pose estimation from monocular images with deep convolutional neural network," in *Proc. Asian Conf. Comput. Vis.*, 2014, pp. 332–347.

[34] M. F. Ghezelghieh, R. Kasturi, and S. Sarkar, "Learning camera viewpoint using CNN to improve 3D body pose estimation," in *Proc. Int. Conf. 3D Vis.*, Oct. 2016, pp. 685–693.

[35] B. Tekin, I. Katircioglu, M. Salzmann, V. Lepetit, and P. Fua. (2016). "Structured prediction of 3D human pose with deep neural networks." [Online]. Available: https://arxiv.org/abs/1605.05180

[36] B. Tekin, P. Marquez-Neila, M. Salzmann, and P. Fua, "Learning to fuse 2D and 3D image cues for monocular body pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jul. 2017, pp. 3941–3950.

[37] M. Lin, L. Lin, X. Liang, K. Wang, and H. Cheng, "Recurrent 3D pose sequence machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2017, pp. 810–819.

[38] L. Sigal, A. O. Balan, and M. J. Black, "HumanEva: Synchronized video and motion capture dataset and baseline algorithm for evaluation of articulated human motion," *Int. J. Comput. Vis.*, vol. 87, nos. 1–2, pp. 4–27, 2010.

[39] C.-H. Chen and D. Ramanan, "3D human pose estimation = 2D pose estimation + matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jul. 2017, pp. 7035–7043.

[40] X. Zhou, M. Zhu, S. Leonardos, K. G. Derpanis, and K. Daniilidis, "Sparseness meets deepness: 3D human pose estimation from monocular video," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4966–4975.

[41] S.-E. Wei, V. Ramakrishna, T. Kanade, and Y. Sheikh, "Convolutional pose machines," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4724–4732.

[42] J. Martinez, R. Hossain, J. Romero, and J. J. Little, "A simple yet effective baseline for 3D human pose estimation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Jun. 2017, pp. 2640–2649.

[43] A. Bansal, A. Farhadi, and D. Parikh, "Towards transparent systems: Semantic characterization of failure modes," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 366–381.

[44] S. Daftry, S. Zeng, J. A. Bagnell, and M. Hebert, "Introspective perception: Learning to predict failures in vision systems," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst.*, Oct. 2016, pp. 1743–1750.

[45] C. Gurău, D. Rao, C. H. Tong, and I. Posner, "Learn from experience: probabilistic prediction of perception performance to avoid failure," *Int. J. Robot. Res.*, vol. 37, no. 9, pp. 981–995, 2018.

[46] D. Mund, R. Triebel, and D. Cremers, "Active online confidence boosting for efficient object classification," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2015, pp. 1367–1373.

[47] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2016, pp. 1135–1144.

[48] R. Wang and B. Bhanu, "Learning models for predicting recognition performance," in *Proc. IEEE Int. Conf. Comput. Vis.*, vol. 2, Oct. 2005, pp. 1613–1618.

[49] M. Boshra and B. Bhanu, "Predicting performance of object recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 9, pp. 956–969, Sep. 2000.

[50] N. Jammalamadaka, A. Zisserman, M. Eichner, V. Ferrari, and C. V. Jawahar, "Has my algorithm succeeded? An evaluator for human pose estimators," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 114–128.

[51] S. Amin, P. Müller, A. Bulling, and M. Andriluka, "Test-time adaptation for 3D human pose estimation," in *Proc. German Conf. Pattern Recognit.*, 2014, pp. 253–264.

[52] A. Dutta, R. Veldhuis, and L. Spreeuwers. (2015). "Predicting face recognition performance using image quality." [Online]. Available: https://arxiv.org/abs/1510.07119

[53] V. Drevelle and P. Bonnifait, "Localization confidence domains via set inversion on short-term trajectory," *IEEE Trans. Robot.*, vol. 29, no. 5, pp. 1244–1256, Oct. 2013.

[54] P. Zhang, J. Wang, A. Farhadi, M. Hebert, and D. Parikh, "Predicting failures of vision systems," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 3566–3573.

[55] Y. Gal and Z. Ghahramani, "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning," in *Proc. Int. Conf. Mach. Learn.*, 2016, pp. 1050–1059.

[56] Y. Gal, R. Islam, and Z. Ghahramani. (2017). "Deep Bayesian active learning with image data." [Online]. Available: https://arxiv.org/abs/1703.02910

[57] A. Kendall, V. Badrinarayanan, and R. Cipolla. (2015). "Bayesian SegNet: Model uncertainty in deep convolutional encoder-decoder architectures for scene understanding." [Online]. Available: https://arxiv.org/abs/1511.02680

[58] E. S. L. Ho, J. C. P. Chan, D. C. K. Chan, H. P. H. Shum, Y.-M. Cheung, and P. C. Yuen, "Improving posture classification accuracy for depth sensor-based human activity monitoring in smart environments," *Comput. Vis. Image Understand.*, vol. 148, pp. 97–110, Jul. 2016.

[59] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang, "Person re-identification via recurrent feature aggregation," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 701–716.

[60] X. Zhou, M. Zhu, S. Leonardos, and K. Daniilidis, "Sparse representation for 3D shape estimation: A convex relaxation approach," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 39, no. 8, pp. 1648–1661, Aug. 2017.

[61] Y. Xu and W. Yin, "A block coordinate descent method for regularized multiconvex optimization with applications to nonnegative tensor factorization and completion," *SIAM J. Imag. Sci.*, vol. 6, no. 3, pp. 1758–1789, 2013.

[62] K. He, X. Zhang, S. Ren, and J. Sun, "Identity mappings in deep residual networks," in *Proc. Eur. Conf. Comput. Vis.*, 2016, pp. 630–645.

[63] A. Vedaldi, K. Lenc, and A. Gupta, "MatConvNet: Convolutional neural networks for MATLAB," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 689–692.

[64] G. Hinton, N. Srivastava, and K. Swersky. (2012). *Lecture 6A Overview of Mini–Batch Gradient Descent, Coursera Lecture Slides*. [Online]. Available: https://class.coursera.org/neuralnets-2012-001/lecture

[65] H. Moon and P. J. Phillips, "Computational and performance aspects of PCA-based face-recognition algorithms," *Perception*, vol. 30, no. 3, pp. 303–321, 2001.

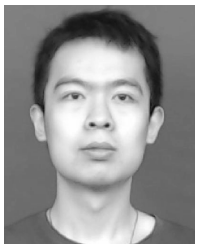
[66] T. Wang, S. Gong, X. Zhu, and S. Wang, "Person re-identification by video ranking," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 688–703.

Guoqiang Liang received the B.S. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from Xi'an Jiaotong University, Xi'an, China, in 2012 and 2018, respectively.

In 2017, he was a Visiting Ph.D. Student with the University of South Carolina, Columbia, SC, USA. He is currently a Post-Doctoral Researcher with the School of Computer Science and Engineering, Northwestern Polytechnical University, Xi'an. His research interests include human pose estimation and human action classification.

Xuguang Lan (M'06) received the Ph.D. degree in pattern recognition and intelligent systems from Xi'an Jiaotong University (XJTU), Xi'an, China, in 2005.

He held a post-doctoral position with the Department of Computer Science, XJTU, from 2005 to 2008. He was a Visiting Scholar with Northwestern University, Evanston, IL, USA, from 2013 to 2014, and the École Centrale de Lyon, Écully, France, in 2005. In 2005, he joined the Institute of Artificial Intelligence and Robotics, XJTU, where he is currently a Professor. His current research interests include computer vision, machine learning, pattern recognition, human–robot collaboration, and content-based image/video coding.



Xingyu Chen received the B.S. degree in software engineering and the M.S. degree in pattern recognition and intelligent systems from the Nanjing University of Science and Technology, Nanjing, China, in 2012 and 2015, respectively. He is currently pursuing the Ph.D. degree with the School of Electronic and Information Engineering, Xi'an Jiaotong University.

His current research interests include face recognition and deep neural networks.

Kang Zheng received the B.E. degree in electrical engineering from the Harbin Institute of Technology in 2012. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA.

His research interests include computer vision, image processing, and deep learning.

Song Wang (SM'13) received the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana–Champaign (UIUC), Champaign, IL, USA, in 2002.

He was a Research Assistant with the Image Formation and Processing Group, Beckman Institute, UIUC, from 1998 to 2002. In 2002, he joined the Department of Computer Science and Engineering, University of South Carolina, Columbia, SC, USA, where he is currently a Professor. His current research interests include computer vision, image processing, and machine learning.

Dr. Wang is a member of the IEEE Computer Society. He is currently serving as the Publicity/Web Portal Chair for the Technical Committee of Pattern Analysis and Machine Intelligence of the IEEE Computer Society and as an Associate Editor for the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, *Pattern Recognition Letters*, and *Electronics Letters*.

Nanning Zheng (SM'93–F'06) received the B.S. and M.S. degrees in information and control engineering from Xi'an Jiaotong University (XJTU), Xi'an, China, in 1975 and 1981, respectively, and the Ph.D. degree in electrical engineering from Keio University, Yokohama, Japan, in 1985.

In 1975, he joined XJTU, where he is currently a Professor and the Director of the Institute of Artificial Intelligence and Robotics. His current research interests include computer vision, pattern recognition, image processing, and hardware implementation of intelligent systems.

Dr. Zheng became a member of the Chinese Academy of Engineering in 1999 and the Chinese Representative on the Governing Board of the International Association for *Pattern Recognition*. He also serves as the Executive Deputy Editor for the *Chinese Science Bulletin* and as an Associate Editor for the IEEE TRANSACTIONS ON INTELLIGENT TRANSPORTATION SYSTEMS.