Learning View-Invariant Features for Person Identification in Temporally Synchronized Videos Taken by Wearable Cameras

Kang Zheng[1], Xiaochuan Fan[2], Yuewei Lin[3], Hao Guo[1], Hongkai Yu[1], Dazhou Guo[1], Song Wang[1]
[1]University of South Carolina, [2]HERE North America, LLC, [3]Brookhaven National Laboratory
{zheng37,hguo,yu55,guo22}@email.sc.edu, {efan3000,ywlin.cq}@gmail.com, songwang@cec.sc.edu

Abstract

In this paper, we study the problem of Cross-View Person Identification (CVPI), which aims at identifying the same person from temporally synchronized videos taken by different wearable cameras. Our basic idea is to utilize the human motion consistency for CVPI, where human motion can be computed by optical flow. However, optical flow is view-variant – the same person's optical flow in different videos can be very different due to view angle change. In this paper, we attempt to utilize 3D human-skeleton sequences to learn a model that can extract view-invariant motion features from optical flows in different views. For this purpose, we use 3D Mocap database to build a synthetic optical flow dataset and train a Triplet Network (TN) consisting of three sub-networks: two for optical flow sequences from different views and one for the underlying 3D Mocap skeleton sequence. Finally, sub-networks for optical flows are used to extract view-invariant features for CVPI. Experimental results show that, using only the motion information, the proposed method can achieve comparable performance with the state-of-the-art methods. Further combination of the proposed method with an appearance-based method achieves new state-of-the-art performance.

1. Introduction

Associating persons from two different views, or person identification, is an important computer vision task. It can be used for surveillance in crowded areas such as airports, train stations and theaters. This task is very challenging because a person's appearance can change significantly due to variations of view angle, illumination, occlusion, background clutter and body pose. Most surveillance systems are based on multi-camera network which are fixed at specific locations and can only cover limited areas. Wearable cameras such as Google Glass can offer more flexibility and better capture scenes as the camera wearer can move freely. To better understand the ongoing events, we may use multiple wearable cameras to capture multiple videos from different view angles. For example, in a protest scene, police officers on site wear Google Glasses for surveillance and their captured videos from different view angles can provide complementary information for detecting/recognizing abnormal activities of people. In order to perform multi-view activity detection/recognition, we need to first identify the same person of common interest from multiple videos, i.e., to perform Cross-View Person Identification (CVPI).

(a) (b)

Figure 1. Examples of (a) matching and (b) non-matching pairs of synchronized videos. Matching pair of videos capture the same person with synchronized and consistent movement, while non-matching pair of videos capture two different persons whose movements are inconsistent.

As in [34], in this paper we assume the videos taken by multiple wearable cameras are *temporally synchronized*, i.e., these videos are aligned in a way that the corresponding frames in all these videos are taken at the same time. This can be achieved by synchronizing clocks in these cameras. This paper is focused on CVPI from a pair of temporally synchronized videos taken by wearable cameras, because it is easy to be extended to more than two synchronized videos. We assume persons in each video are already detected and tracked. From now on, we use the term "*video*" to represent an image sequence of localized regions containing a tracked person. If two temporally synchronized videos capture the same person, these two videos are called a matching or positive pair. Otherwise, they are called a non-matching or negative pair. Figure 1 shows an example of matching pair and non-matching pair of videos.

In [34], Zheng *et al.* propose to estimate the underly-

ing 3D human poses in each video and use them for CVPI. However, this method suffers from inaccurate pose estimation. In this paper, we propose to address CVPI by utilizing human motion consistency. For a pair of temporally synchronized videos that capture the same person, the underlying 3D motion must be consistent, i.e., identical and synchronized. Specifically, we extract optical flows to represent the human motion in each video. However, simply examining the similarity of optical flows is not reliable for CVPI because of two issues: 1) optical flows around human contains both the desired human motion and undesired camera motion; 2) optical flows computed from a matching pair of videos can be significantly different due to view angle difference between the two videos. The first issue can be addressed by camera-motion compensation algorithms, such as [18]. In this paper, we mainly focus on addressing issue 2) for better CVPI.

Our basic idea is to learn view-invariant motion features by introducing 3D human skeleton data into the training process. Specifically, we propose to learn common features shared by the optical flow sequences and the underlying 3D human skeleton sequence. Since 3D human skeletons are independent of view change, the learned features of optical flow sequences are view-invariant. We propose a Triplet Network (TN) which consists of two flow-stream sub-networks and one skeleton-stream sub-network, as shown in Fig. 2. After training the proposed TN, the two flow-stream sub-networks can be used to extract view-invariant features from two optical-flow sequences for CVPI.

The contributions of this paper are: 1) We propose a Triplet Network to extract view-invariant features from optical flows for CVPI; 2) Using synthetic optical flow data and the underlying 3D human skeleton data, we can achieve view invariance in the feature representation of optical flows; 3) We collect a new dataset of synchronized video pairs for evaluating CVPI performance. Experiments are conducted on the datasets used in [34] and our new dataset.

2. Related Work

CVPI in temporally synchronized videos taken by wearable cameras was first proposed in [34], which estimates the underlying 3D poses for the person in each video and uses Euclidean distances between 3D poses estimated in video pairs for CVPI. However, its performance is poor due to inaccurate pose estimation and the use of hand-crafted features and models. In this paper, we address these issues by utilizing motion consistency and developing a new deep-learning approach to feature representation.

Cross-view person identification has also been widely studied in the setting of person re-identification [7], which associates persons in different images or videos from non-overlapping camera views. However, the CVPI studied in this paper differs from the person re-identification in that CVPI is between *temporally synchronized* videos taken by wearable cameras, while person re-identification is between images or videos taken at different time. As a result, person re-identification cannot utilize the exact 3D motion consistency as in CVPI – it usually relies on appearance feature consistency [7, 3, 20, 17, 8, 33, 32, 15, 4] and/or general spatial-temporal feature consistency [35, 27, 19, 22, 30] for person identification. Based on these features, various kinds of distance metrics have been proposed for discriminating matching pairs and non-matching pairs [23, 36, 11, 27, 24, 29, 17, 31, 5].

In this paper, we develop a deep neural network for CVPI, which is inspired by the success of deep models in person re-identification. Li *et al.* [16] propose a Filter Pairing Neural Network (FPNN) to handle misalignment, photometric and geometric transforms, occlusions and background clutter in person re-identification. Ahmed *et al.* [2] introduce a new layer that computes similarities between mid-level features of image pairs. In [28], feature representations are learned from multiple domains/datasets and a Domain Guided Dropout (DGD) algorithm is proposed to improve the feature learning procedure. Wang *et al.* [26] exploit the connection between two formulations of person re-identification: single-image representation matching and cross-image representation classification. A joint learning framework of both formulations is then proposed to improve the features.

With recurrent connections, Recurrent Neural Network (RNN) and Long Short-Term Memory (LSTM) network [12] can learn features from sequence input such as videos, speeches and sentences and have been recently applied to video-based person re-identification [22, 10, 25, 30]. McLaughlin *et al.* [22] combines Convolutional Neural Network (CNN) with RNN to extract spatial features from image frames and temporal features from the whole sequence. A Siamese architecture is used to learn the distance metric. Haque *et al.* [10] applies Recurrent Attention Model (RAM) [10] to person re-identification for depth-based videos. In [25], Variro *et al.* divide the image into horizontal stripes and feed features of these stripes as a sequence to LSTM network. The output of LSTM is then used for person re-identification. Similarly, Yan *et al.* [30] use local color histogram and LBP features as input to LSTM network. In this paper, we also employ CNN and LSTM networks to extract features for CVPI, but using optical flows as input and also incorporating 3D human skeleton data as a stream of the network.

The triplet network developed in this paper shows certain similarity to the triplet network proposed in Hoffer *et al.* [13]. However, there are three major differences between them: 1) Our goal is to learn a better feature representation while the goal of [13] is to learn a distance metric; 2) The inputs of the three subnets in our paper have different modal-

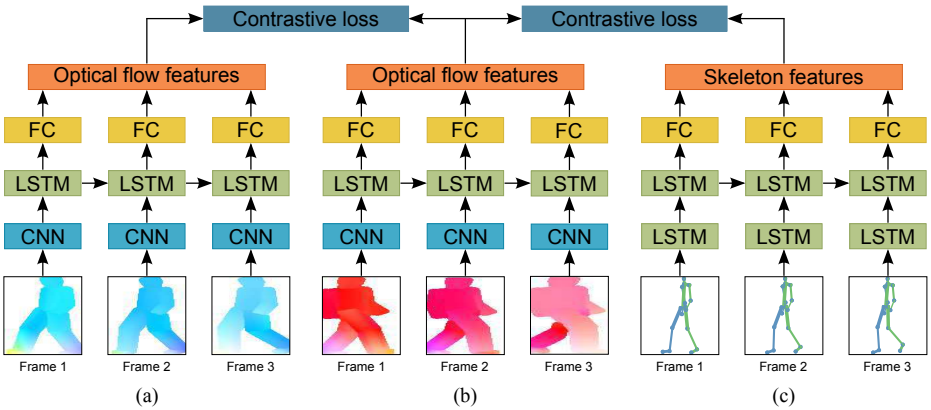

Figure 2. An illustration of the proposed Triplet Network. We input two sequences of synthetic optical flows from different camera views to the flow streams (a) and (b) respectively and the underlying 3D human skeleton sequence (from CMU Mocap database) to the skeleton stream (c). Contrastive loss between the pair of flow stream features, as well as contrastive loss between the flow stream features and the skeleton stream features, are used to train the network. We use both contrastive losses for all the frames in each flow sequence pair and each flow-skeleton sequence pair. For clarity, we only show three frames of optical flows and human skeletons.

ities and therefore, only two subnets share the same parameters and the third subnet does not, while the inputs of the three subnets in [13] have the same modality and their three subnets all share the same parameters; 3) The loss functions used in [13] and this paper are totally different.

3. Proposed Method

3.1. Overview

In this paper, we propose to utilize motion features based on optical flows in videos for CVPI. Because the same person's motion in synchronized videos are consistent, meaning that the same body parts should be moving the same way (upward, downward, etc). However, such high-level semantic features can not be obtained with existing optical-flow-based approaches. We propose to learn view-invariant features from optical flows by introducing 3D human skeleton data into the training process. Specifically, we propose a Triplet Network (TN) to learn view-invariant features from videos' optical flows for CVPI, as shown in Fig. 2. The proposed TN consists of two flow-stream sub-networks and one skeleton-stream sub-network. We feed two synchronized sequences of optical flows (from different views) and the corresponding sequence of 3D human skeletons to the three sub-networks respectively. The flow-stream sub-networks consist of identical CNN and LSTM networks and share parameters. The skeleton-stream sub-network only contains LSTM networks. A fully-connected layer is added to each of the three streams at the end, which outputs the feature embedding for each stream.

For training, we use contrastive loss between the pair of flow-stream features, as well as between flow-stream features and skeleton-stream features. We assume that the features of optical flows and the features of 3D human skeletons should be similar if they originate from the same person at the same time. Conversely, they should be dissimilar if they are from different persons or the same person at different times. We make the same assumption for two sequences of optical flows from different camera views. By simultaneously minimizing both contrastive losses, we can improve the view-invariance in the optical-flow-based features. Since no existing dataset contains optical flow data and corresponding synchronized 3D human skeleton data, we synthesize optical flows from 3D human skeleton sequences in CMU Mocap database [1]. To synthesize optical flows from different view angles, we project the same 3D human skeleton sequence with different camera parameters. The flow-stream sub-networks are further fine-tuned on the training data of person identification. We elaborate on the details of the proposed method in the following sections.

3.2. Sequence-Level Feature Extraction

Feature Extraction from Optical Flows. As shown in Fig. 2, we use the flow-stream sub-networks to extract features from a pair of optical flow sequences and obtain sequence-level features. The parameters of these two flow-stream sub-networks are shared. We use the same network architecture as in Long-term Recurrent Convolutional Networks (LRCN) [6]. The CNN network consists of five convolutional layers, followed by max-pooling layers and dropout layers. One layer of LSTM is followed by a dropout layer to avoid overfitting. We add a fully-connected layer after the LSTM layer to obtain the feature embedding for each optical flow sequence.

The input optical flow sequences are synthesized from 3D human skeleton sequences in CMU Mocap database [1]. Specifically, we first generate synthetic dense trajectories with the method proposed in [9]. A trajectory consists of L frames of displacement vectors representing the motion of a pixel over $L+1$ frames. Each displacement vector can be viewed as a flow vector. To synthesize optical flows from dense trajectories, we only need $L = 1$ frame of dense tra-

jectories since optical flow is the displacement of each pixel between two frames. To simulate optical flows viewed from different viewpoints, we synthesize dense trajectories with various camera settings. We set the polar angle to $\theta = \pi/2$ assuming that the person's videos are taken by cameras at similar height. The azimuthal angle ϕ is set to different values: $\phi = \{0, \pi/3, 2\pi/3, \pi, 4\pi/3, 5\pi/3\}$. To make optical flows similar to real-world video optical flows, we need to convert the dense trajectories from the world coordinate system to the image coordinate system. The spatial position of each trajectory will be converted to the pixel coordinates, and each trajectory will be re-scaled as the flow vector for this pixel. We use the center of the human bounding box as the center of the synthesized optical flow image. For each pixel in the image, we use k nearest trajectories (displacement vectors) to interpolate the flow vector of this pixel. In the experiments, we set $k = 4$. Figure 3 shows an example of the process of synthesizing optical flows from 3D human skeleton sequence.

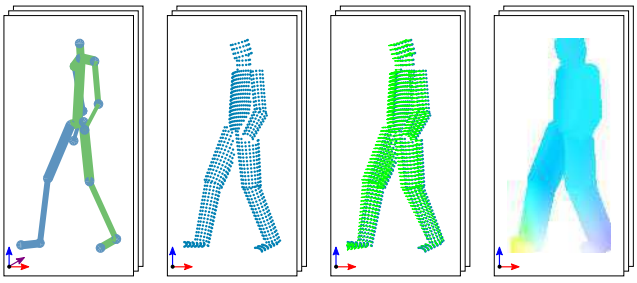

Figure 3. An illustration of the process of synthesizing optical flows from 3D human skeleton data. We first take the 3D human skeletons (a) and approximate the human body surface with cylinders. The human body surface is projected to 2D space as shown in (b). Then, we densely sample on the surface to generate dense trajectories between two frames as shown in (c), where dark blue points are sampled body surface points and green arrows are displacement vectors. Finally, we use interpolation to synthesize optical flows (d).

Following the process in LRCN [6], we formulate the optical flows as flow images and feed them to the flow-stream sub-networks. Specifically, we use the horizontal and vertical part of optical flow as the first two channels of the flow image. The magnitude of optical flow is used as the third channel. The flow images are resized to 227×227 pixels before they are fed to CNN network. Each frame of optical flow features from CNN are the input to an LSTM node, which represents that particular time step. Through the cell state in LSTM, information in early time steps can be propagated to later time steps. With the final fully-connected layer, we obtain the frame-level embedding of the optical flow features. We aggregate these frame-level features as the sequence-level features.

Feature Extraction from 3D Human Skeletons. Inspired by [21], we improve the sequence-level feature extraction from optical flows by introducing an additional modality of 3D human skeleton data, which are just the ones used for synthesizing the optical flows in the other two streams. Each sequence of 3D human skeleton is represented by $T \times J \times 3$ coordinates, where T is the number of frames in this sequence and J is the number of human joints. Since a person's action/motion is independent of its spatial position, 3D human skeleton locations are normalized to a person-centric coordinate system. More specifically, we set the hip joint as the origin and rotate the coordinates so that the person is always facing along positive x-axis. The coordinates of the joints are normalized into the range of $[0, 1]$ to remove subject variance. To extract sequence-level features from 3D human skeleton sequences, we use two-layer LSTM as shown in Fig. 2. This two-layer LSTM is also referred to as eLSTM. The output of eLSTM is also embedded through a fully-connected layer. Finally, sequence-level features of 3D human skeletons are obtained by aggregating each frame's features.

3.3. Network Training

The parameters of flow-stream sub-networks are initialized from LRCN [6] model, while the parameters of skeleton-stream sub-network are randomly initialized. For training process, we simultaneously minimize the contrastive loss between the pair of flow stream features, as well as between the flow stream features and the skeleton stream features. We denote the input optical flow sequences to the flow-stream sub-networks as $\mathbf{v}^{(a)}$ and $\mathbf{v}^{(b)}$ respectively. The input 3D human skeleton sequence is denoted by $\mathbf{P}$. The flow stream feature extraction is represented as a function $F_{\mathbf{v}}(\cdot)$, and the skeleton stream feature extraction is represented as a function $F_{\mathbf{P}}(\cdot)$. The contrastive losses are defined as follows:

$$L_{\mathbf{vv}} = y_1 d_{\mathbf{vv}}^2 + (1 - y_1)\max(m_1 - d_{\mathbf{vv}}, 0)^2, \quad (1)$$

$$d_{\mathbf{vv}} = \left\| F_{\mathbf{v}}(\mathbf{v}^{(a)}) - F_{\mathbf{v}}(\mathbf{v}^{(b)}) \right\|_2, \quad (2)$$

$$L_{\mathbf{vP}} = y_2 d_{\mathbf{vP}}^2 + (1 - y_2)\max(m_2 - d_{\mathbf{vP}}, 0)^2, \quad (3)$$

$$d_{\mathbf{vP}} = \left\| F_{\mathbf{v}}(\mathbf{v}^{(b)}) - F_{\mathbf{P}}(\mathbf{P}) \right\|_2. \quad (4)$$

Here, $d_{\mathbf{vv}}$ and $d_{\mathbf{vP}}$ are Euclidean distances. For positive pairs of optical flow sequences $y_1 = 1$, the features $F_{\mathbf{v}}(\mathbf{v}^{(a)})$ and $F_{\mathbf{v}}(\mathbf{v}^{(b)})$ are encouraged to be similar, while the features are encouraged to be separated by the margin m_1 for negative pairs $y_1 = 0$. Similarly, $F_{\mathbf{v}}(\mathbf{v}^{(b)})$ and $F_{\mathbf{P}}(\mathbf{P})$ are similar for $y_2 = 1$ and separated by a margin m_2 for $y_2 = 0$. Suppose we have a particular optical flow sequence $\mathbf{v}_i^{(b)}$ synthesized from $\mathbf{P}_i$, where i is the sample index. Positive pairs of optical flows are obtained by synthesizing optical flows from $\mathbf{P}_i$ with different camera parame-

ters. Negative pairs are obtained by randomly selecting the optical flows from other 3D skeleton sequences regardless of camera parameters. Similarly, we use the skeleton sequence $\mathbf{P}_i$ and optical flow sequence $\mathbf{v}_i^{(b)}$ to form a positive pair. We randomly select from human skeleton sequences in the Mocap database other than $\mathbf{P}_i$ to form a negative pair with $\mathbf{v}_i^{(b)}$. We do not use contrastive loss between $F_{\mathbf{v}}(\mathbf{v}^{(a)})$ and $F_{\mathbf{P}}(\mathbf{P})$ because it is implied in the two considered contrastive losses: the one between $F_{\mathbf{v}}(\mathbf{v}^{(b)})$ and $F_{\mathbf{P}}(\mathbf{P})$ and the one between $F_{\mathbf{v}}(\mathbf{v}^{(a)})$ and $F_{\mathbf{v}}(\mathbf{v}^{(b)})$.

Due to the difference between synthetic optical flow data and optical flows of real-world videos, we further fine-tune the network on the training data of CVPI video dataset. More specifically, we remove the skeleton stream from the network since there is no 3D human skeleton information for the training data of videos in CVPI dataset. We then sample positive pairs and negative pairs of optical flow sequences similar to sampling synthetic optical flow sequences. The fine-tuning is accomplished by minimizing contrastive loss between outputs of these two flow-stream sub-networks. For videos in CVPI datasets, the camera motion introduced by the movement of camera wearers can lead to some errors in optical flow computation. We use a simple motion compensation (MC) technique as in [18] to address this issue: we compute the average optical flow outside the human bounding box for each frame as the camera motion, which is then subtracted from the optical flow inside the human bounding box.

3.4. Cross-View Person Identification

After training the proposed network, we use the flow-stream sub-network to extract sequence-level features from optical flows in videos for person identification. The features of each video are represented by an $n \times p$ dimensional vector, where n is the number of frames in the video and p is the number of outputs in the final fully-connected layer. Features of the gallery videos are extracted and stored beforehand. For each probe video, we extract its features and compute the similarities between this video to all videos in the gallery set. We use the inverse of the Euclidean distance to measure similarity between videos in a pair. Suppose $F_i^{(a)}$ and $F_j^{(b)}$ represent the features of a video from camera a and a video from camera b respectively, the similarity score between them is defined as follows:

$$S_{i,j} = \frac{1}{\left\| F_i^{(a)} - F_j^{(b)} \right\|_2}. \tag{5}$$

The similarity score is further normalized to the range [0,1] as follows:

$$Score_{i,j}^{T} = \frac{S_{i,j}}{\max_{i,j} S_{i,j}} \tag{6}$$

4. Experiments

In this section we evaluate the proposed method on two datasets in [34], SEQ 1 and SEQ 2, as well as our newly collected dataset, SYN. We compare the proposed method with three state-of-the-art methods and analyze the effectiveness of the proposed method.

SEQ 1 and SEQ 2 contain 114 and 88 pairs of synchronized videos from two different cameras views respectively. The videos are taken by GoPro cameras which are mounted on the wearers' heads. The videos are taken in a football field where multiple pedestrians are present. There are totally 6 subjects walking around and recorded by the cameras. Each subject walks for 4 to 26 times and each video has 120 frames. The same person's videos taken at different times are considered to be non-matching pairs, since their movements are not synchronized and consistent with each other. In some videos, the subject is occluded by other pedestrians. All subjects in SEQ 1 and SEQ 2 are wearing white T-shirts and blue jeans.

We also collect a new dataset, which contains 208 pairs of synchronized videos from two camera views. We refer to this dataset as SYN. Compared to SEQ 1 and SEQ 2, SYN has more video pairs which can provide more reliable evaluation. Also, SYN dataset contains less camera motion which can facilitate the analysis of view-invariant feature learning with less impact by camera motion. The videos are taken in an outdoor environment near a building. There are totally 14 subjects, each of whom walks for 14 to 15 times. Each video has 120 frames. All subjects in this dataset are wearing dark jackets. In this dataset, there are no other pedestrians crossing through in each video.

For evaluation, we follow the generally adopted protocol and split each dataset into two subsets of equal size, i.e., one for training and one for testing. We use Cumulative Matching Characteristics (CMC) as the metric for evaluation. Videos from one camera are used as probe set and videos from the other camera are used as gallery set. For each probe video, we compute the similarity score of the true matching video and find its rank in all videos of gallery set. To obtain more stable results, we repeat the process over 10 random dataset splits and report the average CMC performance.

conv1	**conv2**	**conv3**	**conv4**	**conv5**	**fc6**	**lstm7**	**fc8**
7x7x96 stride 2 norm. pool 3x3	5x5x384 stride 2 norm. pool 3x3	3x3x512 stride 1	3x3x512 stride 1	3x3x384 stride 1 norm. pool 3x3	4096 dropout	256 dropout	512

Figure 4. The architecture of the flow-stream sub-network.

Implementation Details. We use Caffe [14] to implement the proposed Triplet Network (TN). Specifically, we fine-tuned the flow-stream sub-networks whose parameters are initialized from LRCN [6] model. The LRCN model is trained on optical flow sequences of UCF-101 action recog-

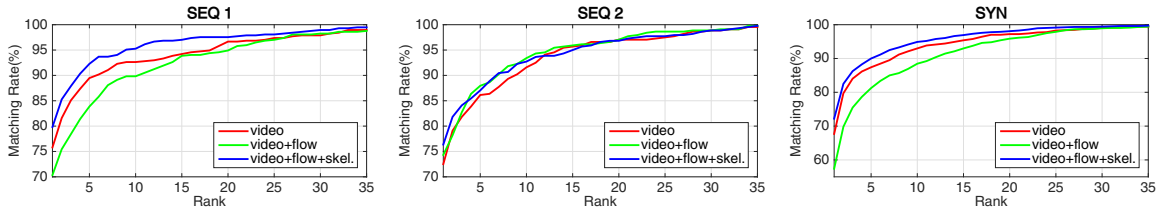

Figure 5. Comparison of the resulting CMC using different training data.

nition dataset. The detailed network architecture and settings are shown in Fig. 4. We empirically set the last fully-connected layer to 512 units, which outputs the feature embedding for each video sequence. For skeleton-stream sub-network, we use two LSTM layers to extract the features, where the first layer contains 1,024 units and the second layer contains 512 units. This LSTM is also followed by a fully-connected layer of 512 units. We use 16 time steps in LSTM layers for both flow streams and skeleton stream, which corresponds to 16 frames. To avoid gradient vanishing and gradient explosion problems in Back-Propagation Through Time (BPTT), we clip the gradients to 15 if they are larger than this value. Both the margins of the two contrastive losses are set to 1.

For training the network, we synthesize 9,654 optical flow sequences from CMU Mocap database. These optical flow sequences are synthesized with 6 different camera parameter settings as described in Section 3.2. Each frame of 3D human skeleton is described by $18 \times 3 = 54$ coordinates, where 18 is the number of joints used. In this work, we use equal length of optical flow sequences and 3D human skeleton sequences as input. Specifically, we use 112 frames in the experiments. The 3D human skeleton sequences in the CMU Mocap database have various length ranging from 2 frames to over 5,000 frames. We segment long sequences into equal-length 112-frame sequences and discard sequences shorter than 112 frames.

In the experiments, we train the network on a NVIDIA GTX 1070 GPU. The network is trained on Mocap synthetic optical flow and skeleton dataset for 2 epochs which takes about one day. We only train for 2 epochs because there are many Mocap sequences that are similar. We further fine-tune on training data of each video dataset for 400 epochs. This takes about 4 hours for a dataset of 200 image sequences. For testing, our method takes less than 1 second for feature extraction and similarity computation between each pair of videos.

4.1. Effectiveness of Synthetic Data for Training

As mentioned above, the network training consists of 1) using synthesized optical flow-data and their 3D skeleton data to train the network, and 2) using real training videos to fine-tune the flow-stream sub-networks. To evaluate the effectiveness of using synthetic optical flow dataset and 3D human skeleton dataset in training, we compare three variants of the proposed method: "*video+flow+skel.*" which runs both 1) and 2); "*video+flow*" which only uses synthesized optical flow data (without using 3D skeleton data) to training the flow-stream sub-networks, followed by running 2); and "*video*" which runs only 2). The evaluation is conducted on all three datasets. The results are shown in Fig. 5. We can see that training with synthetic optical flows and 3D human skeleton first can improve the performance on all the datasets. However, using only synthetic optical flow dataset for the first-step training does not improve the performance. This is mainly caused by the difference between synthetic optical flow and real-world video optical flow. Overall, adding 3D human skeleton can benefit the model and improve the matching rates. This proves the effectiveness of using 3D human skeleton data for CVPI.

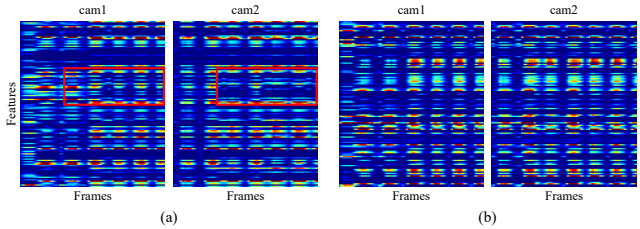

Figure 6. Visualization of the learned features of the same person under different cameras: (a) without using 3D skeleton data in training, (b) incorporating 3D skeleton data in training.

To better understand the effect of skeleton data during training, we visualize in Fig. 6 the features of the same person under different cameras using the proposed network with and without incorporating skeleton data for training. In this figure, the horizontal axis is the frames and vertical axis represents the dimension of features. We can see that the two feature maps in 6(b) are more similar than the two feature maps in 6(a). This indicates that we can extract features with better view invariance by incorporating the 3D skeleton data into training.

Table 1. Comparison of matching rates (%) of the proposed method with and without motion compensation (MC).

Rank	1	5	10	20
SEQ 1 w/o MC	72.28	90.88	94.04	98.25
SEQ 1 w MC	79.82	92.28	95.26	97.54
SEQ 2 w/o MC	75.68	86.82	92.05	97.05
SEQ 2 w MC	76.36	87.05	92.73	96.82
SYN w/o MC	71.06	88.17	92.69	96.63
SYN w MC	72.21	90.00	94.90	98.08

Table 2. Comparison of the proposed method with state-of-the-art methods on SEQ 1, SEQ 2 and SYN dataset in terms of Rank CMC (%)

Dataset	SEQ 1				SEQ 2				SYN			
CMC Rank	1	5	10	20	1	5	10	20	1	5	10	20
DVR [27]	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 [34]	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 [30]	68.42	96.84	98.25	**99.30**	69.77	**96.36**	98.41	**99.32**	56.83	92.40	97.02	98.85
Proposed (optical flow only)	79.82	92.28	95.26	97.54	76.36	87.05	92.73	96.82	72.21	90.00	94.90	98.08
Proposed+RFA ($\lambda = 2$)	79.82	97.19	**98.42**	**99.30**	79.77	95.91	**98.64**	**99.32**	70.67	96.73	98.56	99.71
Proposed+RFA ($\lambda = 1$)	85.09	97.02	98.25	99.30	82.05	95.68	97.73	99.32	76.92	97.31	**99.33**	**100**
Proposed+RFA ($\lambda = 0.5$)	**87.02**	**97.37**	97.89	98.95	**82.05**	94.32	96.59	99.32	**82.12**	**98.37**	**99.33**	**100**

4.2. Effectiveness of Motion Compensation

To alleviate the adverse effect caused by the motion of wearable cameras, we subtract the average optical flow outside the human bounding box from the optical flow inside human bounding box, as used in [18]. This technique is simple but effective. To prove this, we compare the performance of the fine-tuned models on training data with and without motion compensation when computing optical flows. As shown in Table 1, the matching rates are improved with motion compensation.

4.3. Effect of Video Length

We evaluate how the length of videos affects the final matching rates. Specifically, for each dataset, we evaluate the rank-1 matching rates given K frames in each video. We set K from 10 to 100 with the step size of 10. Since each video contains more than 100 frames, we randomly select K consecutive frames in each video pair for similarity calculation. The results are shown in Fig. 7. We can see that, using more frames from the video improves the matching rates. Notice that the matching rates of SYN dataset are lower than those of the other two datasets, especially when very few number of synchronized frames are available. This is because SYN dataset contains more subjects and it is more difficult to identify the same person from a larger set of subjects. We can also see that the matching rates do not increase much after the number of frames reaches 90.

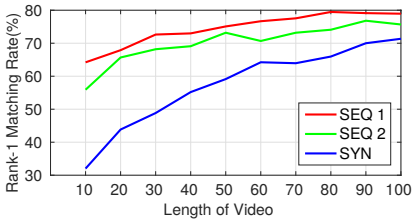

Figure 7. Rank-1 matching rates (%) on videos of different lengths.

4.4. Comparison to the State of the Art

We compare the proposed method with three state-of-the-art methods: 3D pose estimation for person identification (3DHPE) [34], Discriminative Video Ranking (DVR) [27] and Recurrent Feature Aggregation (RFA) [30]. For fair comparison, all the results are obtained by training and testing using the same dataset split. Since 3DHPE method in [34] is unsupervised, we only use the test data to evaluate this method. For RFA method, we resize each image frame to the size of 128×64 pixels, and then extract the color and LBP features to feed to RFA. The results are shown in Table 2. The proposed method has much higher CMC performance than 3DHPE and DVR. This is because we fine-tuned the proposed model from LRCN [6] model, which has been trained on a large amount of data. Compared to RFA, the proposed method achieves much higher Rank-1 matching rates, which has over 10% improvement. For higher ranks, the proposed method has slightly lower matching rates. This is very impressive considering that only motion information (optical flow) is used.

We further combine the proposed TN's output similarity scores with the output scores of RFA as follows:

$$Score = Score^T + \lambda Score^R, \tag{7}$$

where $Score$ is the combined similarity, $Score^T$ is the normalized similarity score computed by the proposed TN method, as defined in Eq. (6), and $Score^R$ is the normalized similarity score computed by the comparison RFA method. By setting $\lambda = \{2, 1, 0.5\}$, the results are shown in the bottom three rows of Table 2. Clearly, combining RFA's output with TN's output can improve the CMC performance significantly and the weight $\lambda \leq 1$ leads to the best performance. This is because the appearance-based features used in RFA and the motion-based features used in TN are complementary to each other in CVPI. Table 3 shows the results when varying the weight λ in a smaller step length for SYN dataset. We can see that $\lambda < 1$ leads to higher matching rates, which indicates relatively more important role of the TN features when combined to the RFA features for CVPI.

Table 3. Rank-1 matching rates (%) using the combined similarity scores with different weights.

λ	0.1	0.3	0.5	0.7	0.9	1.0
Rate	81.34	**83.46**	82.11	79.90	77.88	76.92
λ	1.3	1.5	1.7	2.0	2.3	2.5
Rate	74.42	73.46	71.63	70.67	69.42	68.94

(a) SEQ 1 (b) SEQ 2 (c) SYN

Figure 8. Matching examples. (a), (b), (c) are from SEQ 1, SEQ 2 and SYN datasets respectively. Top two rows show the failure matching examples, while bottom two rows correspond to successful matching examples.

4.5. Qualitative Results

In this section we discuss several correct and incorrect matching examples, as shown in Fig. 8. Examples in three columns are from SEQ 1, SEQ 2 and SYN datasets respectively. Correct matching examples are shown in the bottom two rows, while incorrect ones are shown in the top two rows. Probe sequences are in the first and third rows. For the correct matching examples, all pairs of persons have consistent movement. For the failure case of SEQ 1 example, these two persons are walking very consistently though they are not the same person. It is also easy to confuse motion of the left leg with the right leg because of the ambiguity in projecting 3D human skeleton onto 2D image planes. As shown in the failure case of SEQ 2 example, these two persons have very similar motions. While the person in probe video moves his left leg, the person in the matched gallery video moves his right leg. The incorrect matching example from SYN dataset can be caused by the variance of features in the true matching gallery video. Camera motion is another factor that can corrupt the proposed method, which relies on motion information.

4.6. Cross-Dataset Testing

We also investigate the proposed method's generality by exploring the cross-dataset performance. Specifically, we first train the proposed TN network on synthetic Mocap optical flow and skeleton dataset, using LRCN model for initialization. Then, we fine-tune this model on SEQ 1 dataset and test on SEQ 2 dataset. To better understand the performance of the proposed method, we compare to TN model without Mocap data for training and RFA [30]. The results are shown in Table 4. Clearly, the cross-dataset performance is worse than the within-dataset performance, due to dataset bias. Although we propose to handle view changes by different cameras, the learned representation is still not perfect to generalize over other datasets due to factors such as motion pattern difference and pose variance. Nevertheless, we can see that the proposed method improves the matching rates with additional Mocap data for training, and outperforms RFA as well. This proves the generality of our method on view-invariant feature learning.

Table 4. Cross-dataset performance in terms of Rank CMC(%).

Rank	1	5	10	20
TN w Mocap	11.36	25.00	38.64	63.64
TN w/o Mocap	4.55	11.36	27.27	50.00
RFA [30]	5.00	14.77	32.50	61.14

5. Conclusion

In this paper, we studied the Cross-View Person Identification (CVPI) problem by identifying the same person in temporally synchronized videos taken by different wearable cameras. We proposed a Triplet Network (TN) for CVPI using only motion information. We also proposed to synthesize optical flow dataset from CMU Mocap database for training the network, where the underlying 3D human skeleton data are used as a third stream of the proposed network, which we found that can help learn more view-invariant features for person identification. Experiments on three datasets showed that, using only motion information, the proposed method can achieve comparable results with the state-of-the-art methods. Further combination of the proposed method with an appearance-based method achieves the new state-of-the-art performance. From the experimental results, we can also conclude that motion features and appearance features are complementary to each other for CVPI.

Acknowledgment This work was supported in part by UES Inc./AFRL-S-901-486-002, NSF-1658987, NSFC-61672376 and NCPTT-P16AP00373.

References

[1] CMU Motion Capture Database. http://mocap.cs.cmu.edu. 3

[2] E. Ahmed, M. Jones, and T. K. Marks. An improved deep learning architecture for person re-identification. In *CVPR*, 2015. 2

[3] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Person re-identification using Haar-based and DCD-based signature. In *AVSS*, 2010. 2

[4] S. Bak, E. Corvee, F. Bremond, and M. Thonnat. Person re-identification using spatial covariance regions of human body parts. In *AVSS*, 2010. 2

[5] D. Chen, Z. Yuan, B. Chen, and N. Zheng. Similarity learning with spatial constraints for person re-identification. In *CVPR*, 2016. 2

[6] J. Donahue, L. Hendricks, S. Guadarrama, M. Rohrbach, S. Venugopalan, K. Saenko, and T. Darrell. Long-term recurrent convolutional networks for visual recognition and description. In *CVPR*, 2015. 3, 4, 5, 7

[7] M. Farenzena, L. Bazzani, A. Perina, V. Murino, and M. Cristani. Person re-identification by symmetry-driven accumulation of local features. In *CVPR*, 2010. 2

[8] D. Gray and H. Tao. Viewpoint invariant pedestrian recognition with an ensemble of localized features. In *ECCV*, 2008. 2

[9] 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 *CVPR*, 2014. 3

[10] A. Haque, A. Alahi, and L. Fei-Fei. Recurrent attention models for depth-based person identification. In *CVPR*, 2016. 2

[11] M. Hirzer, P. M. Roth, M. Köstinger, and H. Bischof. Relaxed pairwise learned metric for person re-identification. In *ECCV*, 2012. 2

[12] S. Hochreiter and J. Schmidhuber. Long short-term memory. *Neural Computation*, 1997. 2

[13] E. Hoffer and N. Ailon. Deep metric learning using triplet network. In *ICLR Workshop*, 2015. 2, 3

[14] Y. Jia, E. Shelhamer, J. Donahue, S. Karayev, J. Long, R. Girshick, S. Guadarrama, and T. Darrell. Caffe: Convolutional architecture for fast feature embedding. In *ACM MM*, 2014. 5

[15] R. Layne, T. M. Hospedales, S. Gong, and Q. Mary. Person re-identification by attributes. In *BMVC*, 2012. 2

[16] W. Li, R. Zhao, T. Xiao, and X. Wang. DeepReID: Deep filter pairing neural network for person re-identification. In *CVPR*, 2014. 2

[17] S. Liao, Y. Hu, X. Zhu, and S. Z. Li. Person re-identification by local maximal occurrence representation and metric learning. In *CVPR*, 2015. 2

[18] Y. Lin, K. Abdelfatah, Y. Zhou, X. Fan, H. Yu, H. Qian, and S. Wang. Co-interest person detection from multiple wearable camera videos. In *ICCV*, 2015. 2, 5, 7

[19] K. Liu, B. Ma, W. Zhang, and R. Huang. A spatio-temporal appearance representation for video-based pedestrian re-identification. In *ICCV*, 2015. 2

[20] B. Ma, Y. Su, and F. Jurie. BiCov: A novel image representation for person re-identification and face verification. In *BMVC*, 2012. 2

[21] B. Mahasseni and S. Todorovic. Regularizing long short term memory with 3D human-skeleton sequences for action recognition. In *CVPR*, 2016. 4

[22] N. McLaughlin, J. Martinez del Rincon, and P. Miller. Recurrent convolutional network for video-based person re-identification. In *CVPR*, 2016. 2

[23] B. Prosser, W.-S. Zheng, S. Gong, T. Xiang, and Q. Mary. Person re-identification by support vector ranking. In *BMVC*, 2010. 2

[24] P. M. Roth, M. Hirzer, M. Koestinger, C. Beleznai, and H. Bischof. Mahalanobis distance learning for person re-identification. In *Person Re-Identification*. 2014. 2

[25] R. R. Varior, B. Shuai, J. Lu, D. Xu, and G. Wang. A Siamese long short-term memory architecture for human re-identification. In *ECCV*, 2016. 2

[26] 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 *CVPR*, 2016. 2

[27] T. Wang, S. Gong, X. Zhu, and S. Wang. Person re-identification by video ranking. In *ECCV*, 2014. 2, 7

[28] T. Xiao, H. Li, W. Ouyang, and X. Wang. Learning deep feature representations with domain guided dropout for person re-identification. In *CVPR*, 2016. 2

[29] F. Xiong, M. Gou, O. Camps, and M. Sznaier. Person re-identification using kernel-based metric learning methods. In *ECCV*, 2014. 2

[30] Y. Yan, B. Ni, Z. Song, C. Ma, Y. Yan, and X. Yang. Person re-identification via recurrent feature aggregation. In *ECCV*, 2016. 2, 7, 8

[31] L. Zhang, T. Xiang, and S. Gong. Learning a discriminative null space for person re-identification. In *CVPR*, 2016. 2

[32] R. Zhao, W. Ouyang, and X. Wang. Person re-identification by salience matching. In *ICCV*, 2013. 2

[33] R. Zhao, W. Ouyang, and X. Wang. Unsupervised salience learning for person re-identification. In *CVPR*, 2013. 2

[34] K. Zheng, H. Guo, X. Fan, H. Yu, and S. Wang. Identifying same persons from temporally synchronized videos taken by multiple wearable cameras. In *CVPR Workshop*, 2016. 1, 2, 5, 7

[35] L. Zheng, Z. Bie, Y. Sun, J. Wang, C. Su, S. Wang, and Q. Tian. Mars: A video benchmark for large-scale person re-identification. In *ECCV*, 2016. 2

[36] W.-S. Zheng, S. Gong, and T. Xiang. Person re-identification by probabilistic relative distance comparison. In *CVPR*, 2011. 2