Multi-Video Temporal Synchronization by Matching Pose Features of Shared Moving Subjects

Xinyi Wu[1], Zhenyao Wu[1], Yujun Zhang[2], Lili Ju[1,*], Song Wang[1,*]
[1]University of South Carolina, USA [2]Tianjin University, China
{zhenyao, xinyiw}@email.sc.edu, yujunzhang@tju.edu.cn, ju@math.sc.edu, songwang@cec.sc.edu

Abstract

Collaborative analysis of videos taken by multiple motion cameras from different and time-varying views can help solve many computer vision problems. However, such collaborative analysis usually requires the videos to be temporally synchronized, which can be inaccurate if we solely rely on camera clock. In this paper, we propose to address this problem based on video content. More specifically, if multiple videos cover the same moving persons, these subjects shall exhibit identical pose and pose change at each aligned time point across these videos. Based on this idea, we develop a new Synchronization Network (SynNet) which includes a feature aggregation module, a matching cost volume and several classification layers to infer the time offset between different videos by exploiting view-invariant human pose features. We conduct comprehensive experiments on SYN, SPVideo and MPVideo datasets. The results show that the proposed method can accurately synchronize multiple motion-camera videos collected in real world.

1. Introduction

Motion cameras, such as wearable cameras of Google Glass and GoPro, provide a new perspective to video information collection and analysis and has found many important civil, military, security and law-enforcement applications [38, 40, 39, 37]. On one hand, motion cameras can flexibly cover more areas that are not pre-specified than traditional fixed cameras. On the other hand, by moving to the right positions and view angles with the holder or camera wearer, they may better capture the subjects and activities of interest. By combining the videos taken by multiple motion cameras, e.g., several camera-wearing police officers work together to process an incident, video information processing capability can be significantly enhanced by collaboratively exploring these videos that may record the same subjects or scene from different and time-varying views [3, 40].

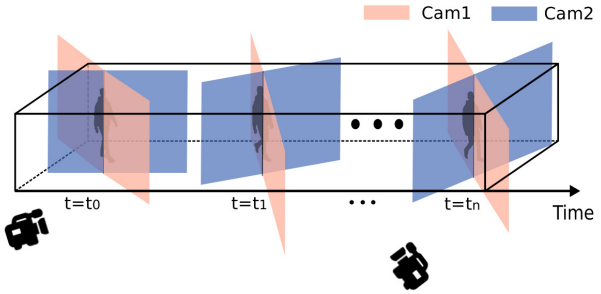

Figure 1. An illustration of two synchronized videos with consistent human poses. Blue and red rectangles indicate the time-varying view angles of the two wearable cameras, respectively.

Collaborative analysis of multiple videos usually requires accurate temporal synchronization of these videos [14, 3, 40], since frames taken at different and unknown time do not show information correspondence and therefore cannot be well integrated for video analysis. For example, motion features extracted from non-synchronized videos may correspond to different stages of the subject's activity and therefore could not be combined for better human activity recognition. Another example is that non-synchronized videos may capture the moving subjects' shapes at different time. Since they do not follow epipolar geometry, they could not be used for facilitating multi-view 3D human reconstruction [6, 12].

While the temporal synchronization of fixed cameras can be easily solved by wired connection and shared clock, motion-camera synchronization is a very challenging problem. Clearly, with limited time accuracy, we could not directly rely on the built-in clock in the camera for accurate synchronization. Using WiFi and Bluetooth for camera-connection and clock sharing may suffer from communication delays and interruptions, as well as limited ranges. In this paper, we propose to synchronize multiple motion cameras based on their video contents. The basic idea is to identify a moving person or multiple moving persons, which we call subjects in this paper, that are present in all the videos.

*Co-corresponding authors.

If these videos are synchronized, their time-varying poses should be consistent in 3D space at any frame across the videos, as shown in Figure 1, and in the ideal cases, we can extract human pose in 3D space frame by frame on each video and then perform cross-video pose matching for synchronizing them. The effectiveness of this idea has been verified in previous works where accurate human pose is constructed by manually annotating joints [27, 31, 30].

However, the effectiveness of this idea is still unknown when using automatically estimated poses. 3D pose estimation from an image or a video itself is a very difficult problem and even the state-of-the-art algorithms may produce large pose estimation errors [7, 17, 22, 5]. The main goal of this paper is *to find out whether such inaccurately estimated poses can still be used for accurately synchronizing videos, by integrating multiple-frame information and employing advanced deep-learning techniques*. More specifically, we propose a new Synchronization Network (SynNet) which exploits *view-invariant* 2D human pose features of the subjects and then develop a feature aggregation module, consisting of deep feature extraction, global feature encoding and temporal encoding, to encode the pose features along the videos. Finally, we build a matching cost volume to learn the view-invariant pose features across two videos and perform classification to identify the time offset between the two videos.

We evaluate the performance of the proposed method on SYN, SPVideo and MPVideo dataset with promising results. The main contributions of this paper are:

- We find that, two motion cameras can be synchronized by matching the pose features shared in the videos, even if the estimated pose features are inaccurate.
- We propose a new deep network called SynNet to synchronize multiple motion-camera videos by exploiting and matching view invariant pose features.
- We collect two new wearable-camera video datasets that can be used for evaluating the performance of video synchronization.

2. Related Work

Many methods have been developed to synchronize multiple fixed cameras by correlating the motion features of their videos [20, 33, 1]. However, these methods are not applicable to our task of motion-camera synchronization – extracted motion features mix inconsistent camera motions and cannot be correlated across different cameras. In this paper, we formulate video synchronization as matching the frames between videos with different temporal offsets and then finding the optimal one. From this perspective, the long-line of research work on image/video matching is related to our work, including the line of video synchronization works in computer graphics [9, 29, 32]. However, most of these methods [9, 29, 10, 32] aim to match frames between videos using appearance features and cannot well address our problem – cameras view difference may make the appearance similarity of matched frames between synchronized videos much lower than the appearance similarity between different frames caused by a small temporal offset.

Additional information sources have also been used to help video synchronization. In [25, 23], flashes or abrupt light changes present in (or added to) the video are detected using special sensors or image processing algorithms for video synchronization. In [15], video synchronization is achieved by combining visual and auditive elements, when a video contains an audio channel. Different from these methods, in this paper we do not use any additional information sources which may not be available in real applications. We synchronize videos only based on their visual content by assuming that they capture at least one same person simultaneously. This is a very reasonable assumption – if there is no any shared person present in multiple videos, the synchronization and collaborative analysis of these videos may not be of much interest.

The proposed work is inspired by previous researches on using pose and pose-trajectory matching for temporally synchronizing independent motion cameras [27, 31, 30]. But all these methods require manual annotation of the important body-joints on all or many video frames, which is clearly not feasible in most applications. The main motivation of this paper is to study whether automatically estimated poses, which is clearly not as accurate as manual annotations [16], can still be used for synchronizing motion cameras. *This is a non-trivial problem – video synchronization needs to discriminate the small temporal offsets while the large pose-estimation error may dominate the pose difference and prevent the discrimination of small offsets. In this paper, we will leverage the information redundancy in multiple videos frames, as well as using deep-learning approach for deriving pose heatmaps, to address this problem.*

Also related are the synchronization of the actions of different people shown in different videos [4, 21, 19], where 2D point trajectories are extracted from each video and then used to align actions in different videos. This action synchronization is different from our problem of synchronizing videos of the same subject. In addition, these 2D-trajectory based methods cannot handle well the motion camera synchronization with large view difference. Our work is also different from prior works on relating first and third-person videos [24, 36]. In essence, we are synchronize multiple third-person videos with at least a shared subject. One goal of the proposed work, as reported in the later experiments, is for the task of accurate frame-by-frame 3D reconstruction of a motion subject, using multiple videos taken from different views. Prior researches have shown that accurate

video synchronization plays a critical role in this task [2, 8].

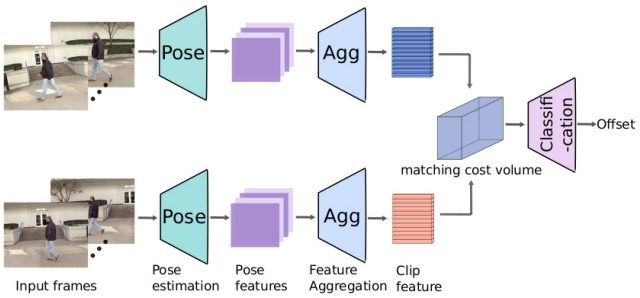

Figure 2. Pipeline of the proposed SynNet.

3. Our Approach

Figure 2 shows the pipeline of the proposed SynNet for synchronizing two motion-camera videos. First, we feed two video clips into two weight-sharing branches, respectively, each of which consists of a network for pose estimation and a feature-aggregation module. We then construct a matching cost volume using the obtained features, followed by final classification to infer the temporal offset between two input video clips. For pose features, we can use any existing pose estimation network, e.g., [5], that can produce a heatmap for each body joint. In the following, we elaborate on the feature aggregation module and the matching cost volume construction.

3.1. Feature aggregation

The feature aggregation module combines the heatmaps of all the joints over all the frames of the input videos. As shown in Figure 4, this module consists of deep feature extraction, global feature encoding and temporal feature encoding.

Deep Feature Extraction By using the pose estimation method in [5], we extract 19 heatmap channels, one for each of 18 joints and the remaining one is for the background. These 18 joints are three on each of the limbs, five on the head, and another one on the neck. In this paper, we only use 18 heatmap channels by excluding the one for the background. As illustrated in Figure 4, we first use the ResNet-50 [13] to encode the 18 channels of heatmap on each frame of the video clip for deep feature extraction. The features extracted by ResNet-50 have dimension $c \times w \times h$, where c is the number of channels, and w and h are the width and height of the ResNet-50 output. The parameters of the deep feature extraction are shared across all the frames in the clip.

Global Feature Encoding Inspired by previous methods which have incorporated full image encoders for improving depth estimation and semantic segmentation [11, 18], we further add a global feature encoding for processing pose features, by using the Convolution-ReLU-Pooling layers. Specifically, we use the 3×3 convolutional layer and the max pooling layer with the kernel size 2×2 and stride 2×2 to reduce the spatial dimensions. In total, five Convolution-ReLU-Pooling structures are used to get the output with a dimension of $F = 256$. The parameters of global feature encoding are shared across all the frames in the clip.

Temporal feature encoding To take advantage of the spatio-temporal information between adjacent frames, we further add a bi-directional convolutional LSTM layer [35, 26] to encode the pose features along each video clip, and then convert the output feature into a vector with the size of $F = 256$ for every frame. The convolutional LSTM which contains convolution operation in its transitions can encode the temporal information while preserving the spatial information. The bi-convLSTM structure doesn't have any information exchange between each pair of two directional LSTM units and the output features produced from the forward and the backward units are then combined to be the final output for each frame.

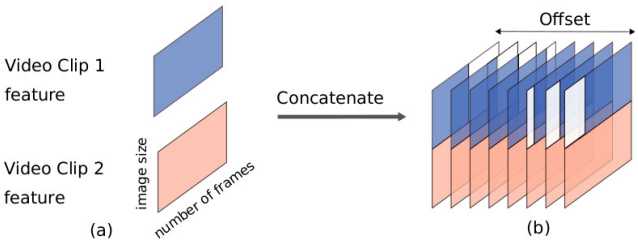

Figure 3. An illustration of constructing the matching cost volume. (a) The two rectangles represent the features of two video clips. (b) Matching cost volume is formed by concatenating features shown in (a) under different offsets.

3.2. Matching cost volume

We concatenate the dimension-F global feature vector from global feature encoding and the dimension-F temporal feature vector from temporal feature encoding on every frame, and then an FC-layer is used to convert the dimension of the concatenated feature to F. The converted features on each frame are finally concatenated to form the video-clip feature of dimension $n \times F$, where n is the number of the frames in a video clip.

We construct a matching cost volume to comprehensively represent the pose information across two video clips by traversing all possible offsets. As shown in Figure 3, we concatenate the blue and red matrices ($(n \times F)$-dimensional features for two input video clips, respectively), with one on the top of the other, by an offset m. This will produce a matrix of dimension $n \times 2F$, if we fill m blank columns (white rectangles in Figure 3(b)) with zero and prune m columns in the other end. By varying m in $[-M, M-1]$, we obtain $2M$ such concatenated matrices, which are stacked sequentially to construct a 3D matching cost volume as shown in

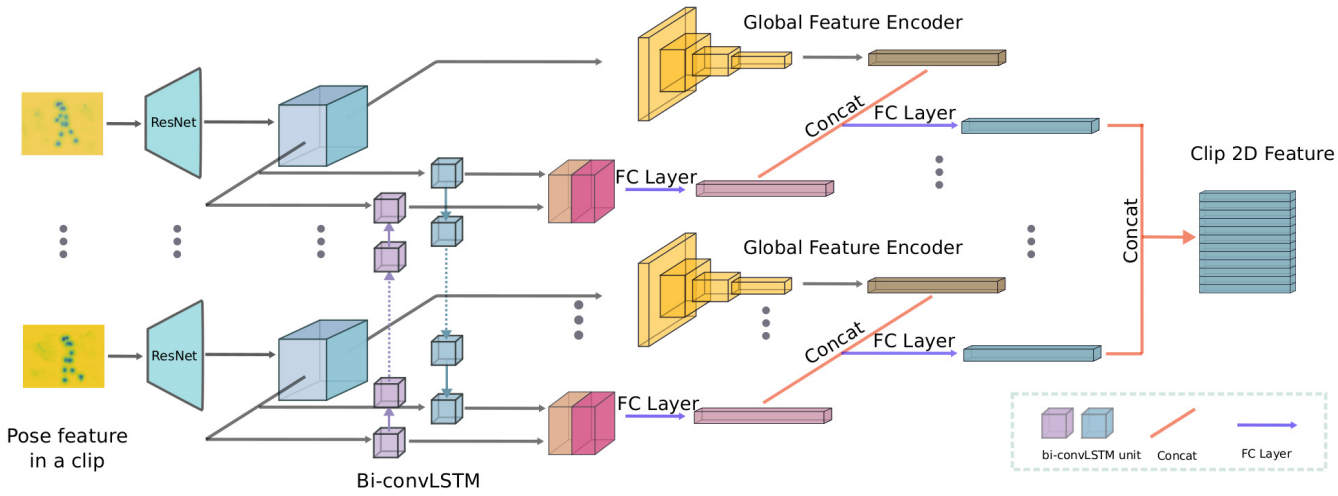

Figure 4. An illustration of the proposed feature aggregation sub-network.

Figure 3(b). The dimension of the matching cost volume is $2M \times n \times 2F$.

3.3. Classification

By quantifying the possible temporal offset to a set of pre-specified integer values, $\{-M, -M+1, \cdots, 0, \cdots, 1, \cdots, M-1\}$, where $M > 0$, we can formulate the problem of video synchronization into a classification problem with $2M$ class labels. Figure 5 gives an illustration of all possible ways to align two videos with $M = 4$. Note that, the selection of a different value for M's requires retraining the SynNet in our method because of the change of class definitions. For classification, the matching cost volume is fed to a batch normalization layer, and then we empirically add three FC-layers to output the probability vector with the size of $2M$ for every possible temporal offset.

3.4. Loss function

Based on the above formulation of the multi-classification problem, we propose a new loss function $Loss$ for SynNet. By combining the cross-entropy loss and a penalty term, this loss function is defined by

$$Loss = \alpha L_{c-entropy} + (1-\alpha) L_{penalty}, \qquad (1)$$

where $0 \leq \alpha \leq 1$ is the weighting parameter for balancing the two loss terms. The first term is the classical cross-entropy loss:

$$L_{c-entropy}(\mathbf{x}, l) = -w_l \log \frac{e^{x_l}}{\sum_{j=1}^{M} e^{x_j}} \qquad (2)$$

where $\mathbf{x} = (x_0, x_1, \cdots, x_{2M-1})^T \in \mathbb{R}^{2M}$ is the probability vector from the final FC-layer, the class label l takes value in $[0, 2M-1]$ where each label represents a possible offset. w_l is the weight for each class label and we directly set $w_l = 1$ for all labels, because we have no prior knowledge on which offset may occur more often than the others.

The second loss term $L_{penalty}$ is designed to penalize all offset misclassifications according to the difference between the predicted result and the ground truth and it is defined by

$$L_{penalty} = \frac{1}{2M-1}|l - x|, \qquad (3)$$

where x is the label with the maximum probability in the final FC-layer.

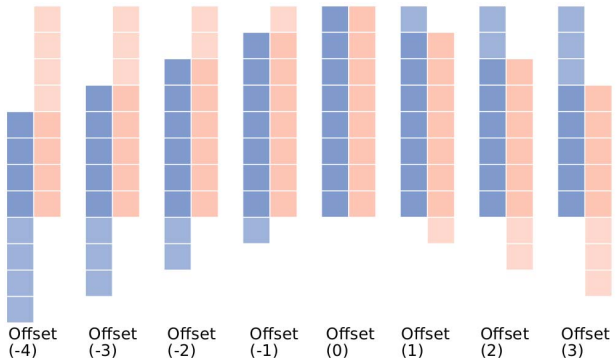

Figure 5. An illustration of possible offsets between two videos. The blue sequence stands for the reference video, and the red one stands for the other video. There are in total eight possible ways to align these two video clips by setting $M = 4$.

4. Experiments

In this section, we evaluate the proposed SynNet on three datasets. One is SYN dataset [39] collected for cross-video person identification. The other two are the SPVideo and MPVideo datasets which we newly collected for this work.

We perform ablation studies using these datasets to investigate the influence of feature types, the number of joints in heatmaps and the video down-sampling rates on the performance of SynNet. We also show comparison results with other existing methods. To show the importance of accurate video synchronization, we finally apply synchronization results for video-based 3D human reconstruction.

4.1. Datasets

SYN [39] SYN contains 208 pairs of temporally synchronized videos taken by two GoPro cameras with different and time-varying views. It has 14 subjects and all the subjects in the dataset are wearing dark jacket. Each video has 120 frames. We choose a total of 70 synchronized video pairs, temporally down-sample them and then select 70×23 synchronized pair of subsequences with different starting frame for training data construction. The typical setting is to use a down-sampling rate of 4, i.e., keeping 1 frame for every 4 frames, and a subsequence length of 8 as shown in Figure 5. Given M being the range of offset, we construct $70 \times 23 \times 2M$ video-clip pairs with ground-truth temporal offsets for training. We then select 56 other video pairs in SYN, follow the same processing as in training set construction. This way, we construct $56 \times 2M$ video-clip pairs for testing.

SPVideo We collect a new dataset SPVideo, in which each video only contain one subject. It contains 120 pairs of synchronized videos taken by two GoPro cameras mounted on two wearers' head. Compared to SYN, SPVideo has much more complicated human movement, including playing the smartphone while walking, taking photos, picking up the phones, jumping, taking off the jacket, and walking. The videos are taken in an outdoor environment near a building and each video has 120 frames. Follow the way we did for SYN dataset, we construct $10 \times 23 \times 2M$ video-clip pairs for training and $8 \times 2M$ video-clip pairs for testing.

MPVideo MPVideo is another dataset we collect for experiments. There are three or four people present in each video. We use this dataset to evaluate the proposed SynNet in the case of multiple subjects shared in different videos. This dataset contains 4 long synchronized videos taken by two GoPro cameras mounted on two wearers' head. The videos are taken in an outdoor environment with 240fps. We construct $38 \times 2M$ video-clip pairs for training and $15 \times 2M$ video-clip pairs for testing.

4.2. Evaluation criteria

We introduce two criteria, the accuracy rate and the SynError, for performance evaluation. The accuracy rate is used to measure the proportion of the correctly synchronized video pairs in the testing set:

$$\text{Accuracy} = N_c/N, \tag{4}$$

where N_c is the number of test video pairs with correctly identified offset and N is the total number of the test video pairs. The higher the value of Accuracy, the better the performance. The synchronization error (in seconds) is used to measure the time deviation between the predicted offset and the true offset for the test video pairs:

$$\text{SynError} = \left(\frac{1}{\text{N}} \sum_{\text{i}=1}^{\text{N}} |\text{R}_\text{i} - \text{T}_\text{i}| \right) \times \frac{\text{r}}{\text{fps}}, \tag{5}$$

where R_i is the offset predicted from the final classification for the i-th video pair, T_i is the true offset for that video pair, r is the video down-sampling rate and fps is the number of frames per second for the input videos. The smaller the value of SynError, the better the performance.

4.3. Model specifications

The proposed SynNet was implemented using PyTorch. Specifically, we first run the widely-used human pose estimator [5] to obtain heatmaps for human joints on each frame. We train SynNet using SGD for 500 epochs for all the following experiments with a batch size of 1 and a constant learning rate of 0.0001. Once that we have precomputed the heatmaps for every video clip pairs, it takes approximatively 2 days to finish the training on the training data constructed from SYN, SPVideo and MPVideo respectively on an NVIDIA GTX 1080 GPU.

4.4. Ablation study

Choices of the features In the proposed SynNet, we first use a network for extracting pose features. We also try the use of appearance and motion features instead of pose features as the input. Specifically, for using appearance features, we simply remove the pose estimation subnetwork by directly feeding input RGB images to the encoder. For using motion features, we compute the optical flow [28] on each frame and feed directly to the encoder. We can also combine all three or any two of these features and evaluate their influence to the final synchronization. In this experiment, we set $M = 4$ and temporally down-sample all the videos by rate 4 before feeding to SynNet. The evaluation results on the test set constructed from SYN and SPVideo after training for 500 epochs from scratch are numerically reported in Table 1.

For the SYN dataset, we observe that the use of pose features as the input attained the highest accuracy rate of 92.63% and the lowest Syn-error of 0.022s at the same time, which gives the best performance. This result indicates that the proposed SynNet can capture the view-invariant information to synchronize the motion-camera videos. For the SPVideo dataset, the performance of using pose features is also better than the use of other features. In both of the two datasets, it shows that the use the appearance features leads

to very poor performance. This verifies our analysis in Section 2 that appearance features, which many of the previous image/video matching methods are based on, are not suitable for our task. The use of optical flow also leads to very poor performance. We conjecture this is due to the mixture of person's movements and the camera movements. Such mixture movement makes it very difficult for the network to learn view-invariant motion features for video synchronization. In the remaining experiments, we only use the pose features for SynNet.

Table 1. Comparison of the accuracy rate and SynError of the proposed SynNet with different input features after training for 500 epochs and M is set to 4. A, F & P represent appearance, optical flow & pose features respectively.

	SYN		SPVideo	
Features	Accuracy	SynError	Accuracy	SynError
A	13.26%	0.2915	10.94%	0.2833
F	26.78%	0.2438	17.18%	0.3227
P	92.63%	0.0220	67.19%	0.1042
A+F	6.34%	0.3661	9.38%	0.3122
P+F	50.52%	0.1820	28.13%	0.2486
P+A	36.63%	0.2241	21.88%	0.2764
P+A+F	30.62%	0.2525	14.06%	0.2949

Different down-sampling rate In the above experiment, we temporally down-sample the videos by rate 4 to construct the training and testing videos from SYN and SPVideo. We also try different down-sampling rates r and Table 2 reports the resulting accuracy rate and SynError on the testing data constructed from SYN and SPVideo datasets. With different down-sampling rates, a unit of offset represents different number of frames in the original video – with down-sampling rate of k, one unit of offset represents a possible maximum of $(k-1)$-frame offset for the original video pair before down-sampling. Based on these results, in the following experiments, we always set down-sampling rate to 4 when M is set to 4.

Table 2. Comparison of accuracy rate and SynError of the proposed SynNet by varying down-sampling rate for training and testing and M is set to 4.

Down-samp.	SYN		SPVideo	
r	Accuracy	SynError	Accuracy	SynError
5	89.29%	0.0435	57.81%	0.1172
4	92.63%	0.0220	67.19%	0.1042
3	83.93%	0.0350	35.94%	0.1125
2	74.55%	0.1510	21.88%	0.1218
1	14.29%	0.0938	12.50%	0.0966

Number of joints We also examine the impact of the number of joints (channels) selected for the joint heatmaps to the performance of SynNet. For this purpose, we also try the cases of using 5 channels (hips, shoulders and neck), 9 channels (neck, shoulders, wrists, hips, ankles) and 13 channels (all the joints except for those on the face) for joint heatmaps and the results are shown in Table 3. We can see that the use of more joints usually lead to better performance of video synchronization. In this paper, we use all 18 joints.

Table 3. Comparison of accuracy rate and SynError of the proposed SynNet by using different number of joints and M is set to 4.

	SYN		SPVideo	
# Joints	Accuracy	SynError	Accuracy	SynError
5	71.56%	0.0669	21.88%	0.2437
9	87.05%	0.0375	35.94%	0.1499
13	91.52%	0.0303	57.81%	0.0937
18	92.63%	0.0220	67.19%	0.1042

Model variants We compare a number of model variants of SynNet by removing one key component at a time and the results are shown in Table 4. These results show that all our proposed components in SynNet contribute positively to the final performance.

Table 4. Performance of SynNet variants. For SynNet without bi-convLSTM, we use a one direction convLSTM instead. For SynNet without matching cost volume, we use simple feature concatenation instead. For the variant of 'Without the penalty term', we simply use the cross-entropy loss for optimization.

	SYN	
Method	Accuracy	SynError
w/o bi-convLSTM	85.04%	0.0303
w/o temporal feature encoding	69.87%	0.0345
w/o global feature encoding	79.42%	0.0301
w/o global & temporal encoding	26.04%	0.2833
w/o matching cost volume	76.79%	0.0529
w/o the penalty term	87.50%	0.0333
SynNet	92.63%	0.0220

4.5. Comparison with other methods

We choose two correlation-based methods [32, 15] for comparison study. For [32], we implement its feature embedding network to extract features from both input videos. Then we implement its correlation operation to compute the correlation between the features of the two videos by applying different offsets. Finally, the offset with the maximum correlation is taken for synchronization. For [15], we use its video subnetwork to encode the two input video clips, and then use its objective contrastive loss to minimize the distance of their encoded features by applying different off-

sets. Finally, the offset with the minimum loss is taken for synchronization. The results are shown in Table 5.

Table 5. Comparison results between the proposed SynNet and two correlation-based methods on SYN dataset.

Method	Accuracy	SynError
Wieschollek *et al*. [32]	26.04%	0.2883
Korbar *et al*. [15]	22.15%	0.2816
SynNet	92.63%	0.0220

We can see that the proposed method is more accurate for video synchronization by exploring view-invariant pose features, while the correlation-based methods do not work very well by feature correlation, especially when the two input videos are taken from significantly different view angles, leading to very large difference on appearance and background. Figure 6 (a-c) show the frame-by-frame cost matrix between the features extracted from two synchronized video clips, each of which consists of 8 frames, by using the proposed SynNet and the feature extraction methods in [32], and [15], respectively. The ij-th element in the cost matrix is the Euclidean distance between the feature extracted from frame i of the first video and the feature extracted from frame j of the second video. We can see the pose-based features extracted by SynNet lead to better matching of the two input videos by highlighting more on the diagonal elements of the cost matrix. However, this diagonal highlighting is not perfect and we further build matching cost volume for better classifying their offset in SynNet.

4.6. Real-world videos

To further evaluate the proposed SynNet, we extend our experiments to real-world videos, including 1) the presence of multiple subjects (using the MPVideo dataset), and 2) longer video clips with more frames. For 1), for each subject shared between two videos, we apply SynNet to compute their temporal offset. Then we compute the average offset over all shared subjects for final synchronization. Table 6 reports the synchronization results on MPVideo dataset with multiple subjects. For 2), we can apply synchronized temporal sliding windows with step length of 1 frame on both input videos. On each pair of corresponding windowed video clips, we can apply SynNet to identify the offset. Finally we compute the average offset over all the windows for final synchronization. Figure 7 shows a pair of longer video clip (25s for each) that are synchronized by SynNet and Figure 6 (d) shows the (normalized) confusion matrix in terms of predicted and ground-truth offsets over all the short (8 frames) clips windowed from the original 25s videos. In general, we can see that the presence of more subjects and using longer videos provide richer information that can further improve video synchronization.

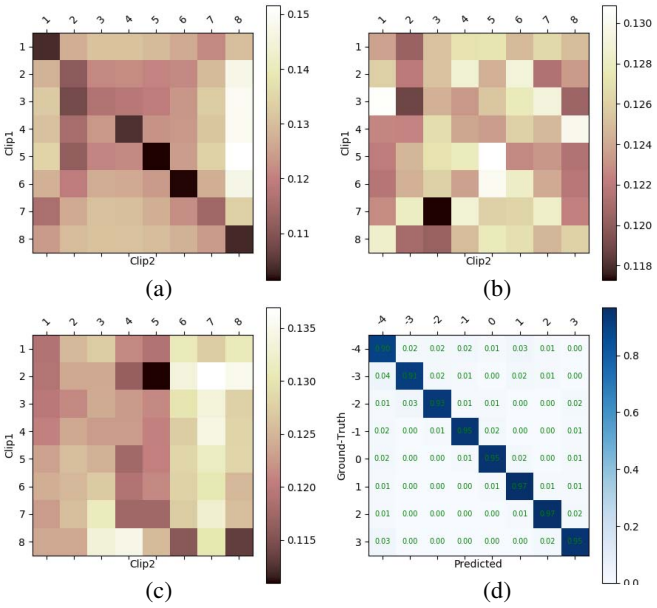

Figure 6. (a-c) Feature cost matrices of two synchronized videos from the proposed SynNet, [15] and [32], respectively. (d) (Normalized) confusion matrix in terms of predicted and ground-truth offsets over all the short (8 frames) clips windowed from a pair of 25s' long videos constructed from the SYN dataset.

Table 6. Synchronization results of multi-subject videos in MPVideo dataset.

Subjects	Accuracy	SynError
Person 1	43.53%	0.1574
Person 2	53.61%	0.1162
Combined	61.81%	0.0742

4.7. Evaluation on 3D human reconstruction

Accurate video synchronization is particularly important for reconstructing 3D moving subjects, where different videos represent different views [8]. Without knowing the camera parameters and poses (cameras are moving), multi-view 3D reconstruction is difficult. In this paper, we use SC-GAN [34] to estimate the depth map of a video frame by combining information from its adjacent frames. We then combine the depth maps of the aligned frames in synchronized videos by converting them into point clouds and then manually assembling them in 3D space. Sample results are displayed in Figure 8. We can see that, with accurate synchronization, the point clouds from two videos can be well assembled to reconstruct larger areas of the human body. If two videos are not well synchronized, even with very small temporal offset, the point clouds sampled from corresponding frames in two videos imply different human poses and cannot be well assembled for 3D reconstruction.

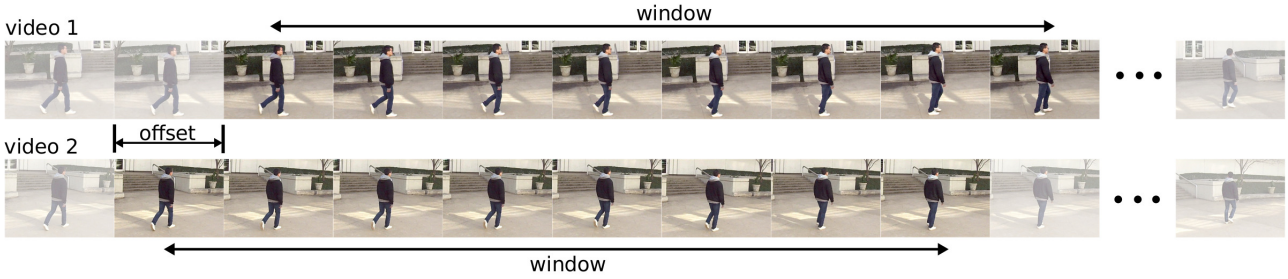

Figure 7. Synchronization of a pair of long videos (25s each) in SYN dataset by the proposed SynNet. For each video we highlight here a windowed clip of 8 frames.

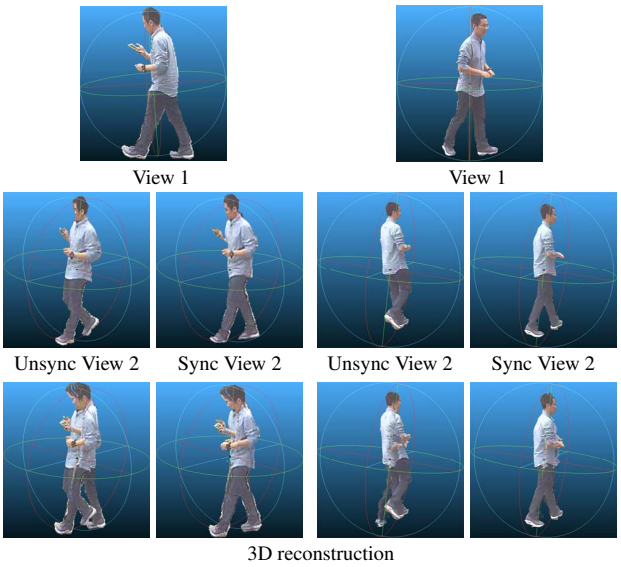

Figure 8. Sample results of 3D human reconstruction by combining two views (corresponding frames of two videos). Top row shows two images selected for the first view image. Second row shows the corresponding second-view images – we try both the one synchronized and the one unsynchronized with the first-view image, and the corresponding 3D reconstructions are shown in the bottom row.

4.8. WiFi-based camera synchronization

To get an idea of motion-camera synchronization accuracy by using WiFi, we build a simple networked system consisting of two GoPro cameras, three smartphones, and a millisecond clock, as shown in Figure 9. The test consists of the following steps: 1) Two GoPros G_A & G_B are connected to two smartphones S_A & S_B respectively through WiFi, which enables the display of what GoPro sees on the screen of its connected smartphone. 2) Let the two GoPros shoot at the same millisecond clock. 3) The third smartphone S_C shoots at the the screens of the two GoPro-connected smartphones. From the image taken by S_C, we can compute the synchronization error between videos taken by GoPro cameras if they are WiFi-connected. We use camera S_C to take 100 images and the average synchronization error is 0.662 seconds with variance of 0.729. From Table 4, we can see that our proposed SynNet can get an average synchronization error of 0.022 seconds, which is much lower than using WiFi for synchronization, not to mention that the WiFi network may experience more delays with crowded users.

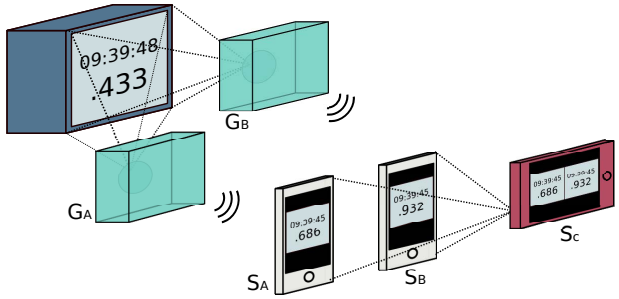

Figure 9. An illustration of WiFi-based test for camera synchronization.

5. Conclusion

In this paper, we proposed a SynNet to temporally synchronize multiple motion-camera videos based on moving subjects shared in these videos. We reformulate this video synchronization problem to a classification problem by identifying underlying temporal offset between two videos. Using a deep neural network structure, SynNet starts with a pose estimation subnetwork to extract view-invariant pose features, which are then encoded using a feature aggregation module. Encoded features from two videos are combined into a matching cost volume to traverse all possible temporal offsets, followed by final classification layers. Experiments on the three datasets, including two new datasets we collected for the proposed work, showed that the use of pose features leads to better video synchronization than the use of appearance and motion features.

References

[1] Cenek Albl, Zuzana Kukelova, Andrew Fitzgibbon, Jan Heller, Matej Smid, and Tomas Pajdla. On the two-view geometry of unsynchronized cameras. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 4847–4856, 2017.

[2] Thiemo Alldieck, Marcus Magnor, Weipeng Xu, Christian Theobalt, and Gerard Pons-Moll. Video based reconstruction of 3d people models. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Luca Ballan, Gabriel J Brostow, Jens Puwein, and Marc Pollefeys. Unstructured video-based rendering: Interactive exploration of casually captured videos. *ACM Transactions on Graphics (TOG)*, 29(4):87, 2010.

[4] Jean-Charles Bazin and Alexander Sorkine-Hornung. Actionsnapping: Motion-based video synchronization. In *European Conference on Computer Vision (ECCV)*, pages 155–169. Springer, 2016.

[5] Zhe Cao, Tomas Simon, Shih-En Wei, and Yaser Sheikh. Realtime multi-person 2d pose estimation using part affinity fields. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[6] Christopher B Choy, Danfei Xu, JunYoung Gwak, Kevin Chen, and Silvio Savarese. 3d-r2n2: A unified approach for single and multi-view 3d object reconstruction. In *European Conference on Computer Vision (ECCV)*, pages 628–644. Springer, 2016.

[7] Yong Du, Wei Wang, and Liang Wang. Hierarchical recurrent neural network for skeleton based action recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[8] Tobias Duckworth and David J Roberts. Camera image synchronisation in multiple camera real-time 3d reconstruction of moving humans. In *IEEE/ACM International Symposium on Distributed Simulation and Real Time Applications*, pages 138–144. IEEE Computer Society, 2011.

[9] Ahmed Elhayek, Carsten Stoll, Kwang In Kim, H-P Seidel, and Christian Theobalt. Feature-based multi-video synchronization with subframe accuracy. In *Joint DAGM (German Association for Pattern Recognition) and OAGM Symposium*, pages 266–275. Springer, 2012.

[10] Ido Freeman, Patrick Wieschollek, and Hendrik PA Lensch. Robust video synchronization using unsupervised deep learning. *CoRR*, 2016.

[11] Huan Fu, Mingming Gong, Chaohui Wang, Kayhan Batmanghelich, and Dacheng Tao. Deep ordinal regression network for monocular depth estimation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2002–2011, 2018.

[12] Andreas Geiger, Julius Ziegler, and Christoph Stiller. Stereoscan: Dense 3d reconstruction in real-time. In *IEEE Intelligent Vehicles Symposium (IV)*, pages 963–968. Ieee, 2011.

[13] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[14] Wei Jiang and Jinwei Gu. Video stitching with spatial-temporal content-preserving warping. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 42–48, 2015.

[15] Bruno Korbar, Du Tran, and Lorenzo Torresani. Co-training of audio and video representations from self-supervised temporal synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

[16] Guoqiang Liang, Xuguang Lan, Kang Zheng, Song Wang, and Nanning Zheng. Cross-view person identification by matching human poses estimated with confidence on each body joint. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2018.

[17] Jun Liu, Amir Shahroudy, Dong Xu, and Gang Wang. Spatio-temporal lstm with trust gates for 3d human action recognition. In *European Conference on Computer Vision (ECCV)*, pages 816–833. Springer, 2016.

[18] Jonathan Long, Evan Shelhamer, and Trevor Darrell. Fully convolutional networks for semantic segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3431–3440, 2015.

[19] Cheng Lu and Mrinal Mandal. A robust technique for motion-based video sequences temporal alignment. *IEEE Transactions on Multimedia*, 15(1):70–82, 2013.

[20] Dmitry Pundik and Yael Moses. Video synchronization using temporal signals from epipolar lines. In *European Conference on Computer Vision (ECCV)*, pages 15–28. Springer, 2010.

[21] Cen Rao, Alexei Gritai, Mubarak Shah, and Tanveer Syeda-Mahmood. View-invariant alignment and matching of video sequences. In *IEEE International Conference on Computer Vision (ICCV)*. IEEE, 2003.

[22] Amir Shahroudy, Jun Liu, Tian-Tsong Ng, and Gang Wang. Ntu rgb+d: A large scale dataset for 3d human activity analysis. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[23] Prarthana Shrestha, Hans Weda, Mauro Barbieri, and Dragan Sekulovski. Synchronization of multiple video recordings based on still camera flashes. In *ACM International Conference on Multimedia*, pages 137–140. ACM, 2006.

[24] Gunnar A Sigurdsson, Abhinav Gupta, Cordelia Schmid, Ali Farhadi, and Karteek Alahari. Actor and observer: Joint modeling of first and third-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7396–7404, 2018.

[25] Matej Šmıd and Jirı Matas. Rolling shutter camera synchronization with sub-millisecond accuracy. In *International Conference on Computer Vision Theory and Applications*, page 8, 2017.

[26] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *European Conference on Computer Vision (ECCV)*, pages 715–731, 2018.

[27] Philip A Tresadern and Ian D Reid. Video synchronization from human motion using rank constraints. *Computer Vision and Image Understanding*, 113(8):891–906, 2009.

[28] Limin Wang, Yuanjun Xiong, Zhe Wang, Yu Qiao, Dahua Lin, Xiaoou Tang, and Luc Van Gool. Temporal segment

networks: Towards good practices for deep action recognition. In *European Conference on Computer Vision (ECCV)*, pages 20–36. Springer, 2016.

[29] Oliver Wang, Christopher Schroers, Henning Zimmer, Markus Gross, and Alexander Sorkine-Hornung. Videosnapping: Interactive synchronization of multiple videos. *ACM Transactions on Graphics (TOG)*, 33(4):77, 2014.

[30] Xue Wang, Jianbo Shi, Hyun Soo Park, and Qing Wang. Motion-based temporal alignment of independently moving cameras. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(11):2344–2354, 2017.

[31] Xue Wang and Qing Wang. Video synchronization with trajectory pulse. In *Chinese Conference on Intelligent Visual Surveillance*, pages 12–19. Springer, 2016.

[32] Patrick Wieschollek, Ido Freeman, and Hendrik PA Lensch. Learning robust video synchronization without annotations. In *IEEE International Conference on Machine Learning and Applications (ICMLA)*, pages 92–100. IEEE, 2017.

[33] Lior Wolf and Assaf Zomet. Wide baseline matching between unsynchronized video sequences. *International Journal of Computer Vision*, 68(1):43–52, 2006.

[34] Zhenyao Wu, Xinyi Wu, Xiaoping Zhang, Song Wang, and Lili Ju. Spatial correspondence with generative adversarial network: Learning depth from monocular videos. In *IEEE International Conference on Computer Vision (ICCV)*, 2019.

[35] SHI Xingjian, Zhourong Chen, Hao Wang, Dit-Yan Yeung, Wai-Kin Wong, and Wang-chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *Advances in Neural Information Processing Systems*, pages 802–810, 2015.

[36] Mingze Xu, Chenyou Fan, Yuchen Wang, Michael S Ryoo, and David J Crandall. Joint person segmentation and identification in synchronized first-and third-person videos. In *European Conference on Computer Vision (ECCV)*, pages 637–652, 2018.

[37] Takuma Yagi, Karttikeya Mangalam, Ryo Yonetani, and Yoichi Sato. Future person localization in first-person videos. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[38] Yan Yan, Elisa Ricci, Gaowen Liu, and Nicu Sebe. Egocentric daily activity recognition via multitask clustering. *IEEE Transactions on Image Processing*, 24(10):2984–2995, 2015.

[39] Kang Zheng, Xiaochuan Fan, Yuewei Lin, Hao Guo, Hongkai Yu, Dazhou Guo, and Song Wang. Learning view-invariant features for person identification in temporally synchronized videos taken by wearable cameras. In *IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[40] Kang Zheng, Hao Guo, Xiaochuan Fan, Hongkai Yu, and Song Wang. Identifying same persons from temporally synchronized videos taken by multiple wearable cameras. In *IEEE Conference on Computer Vision and Pattern Recognition Workshops*, pages 810–818, 2016.