Advancing Video Synchronization with Fractional Frame Analysis: Introducing a Novel Dataset and Model

Yuxuan Liu[1,2], Haizhou Ai[1,2], Junliang Xing[1,2*], Xuri Li[3], Xiaoyi Wang[4], Pin Tao[1,2],

[1]Key Laboratory of Pervasive Computing, Ministry of Education
[2]Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China
[3]Beijing University of Technology, Beijing 100124, China
[4]Unaffiliated Scholar, Haidian District, Beijing, China
liu-yx22@mails.tsinghua.edu.cn, ahz@tsinghua.edu.cn, jlxing@tsinghua.edu.cn,
bjutlxr@emails.bjut.edu.cn, wangxiaoyi00@gmail.com, taopin@tsinghua.edu.cn

Abstract

Multiple views play a vital role in 3D pose estimation tasks. Ideally, multi-view 3D pose estimation tasks should directly utilize naturally collected videos for pose estimation. However, due to the constraints of video synchronization, existing methods often use expensive hardware devices to synchronize the initiation of cameras, which restricts most 3D pose collection scenarios to indoor settings. Some recent works learn deep neural networks to align desynchronized datasets derived from synchronized cameras and can only produce frame-level accuracy. For fractional frame video synchronization, this work proposes an Inter-Frame and Intra-Frame Desynchronized Dataset (IFID), which labels fractional time intervals between two video clips. IFID is the first dataset that annotates inter-frame and intra-frame intervals, with a total of $382,500$ video clips annotated, making it the largest dataset to date. We also develop a novel model based on the Transformer architecture, named InSynFormer, for synchronizing inter-frame and intra-frame. Extensive experimental evaluations demonstrate its promising performance. The dataset and source code of the model are available at https://github.com/yuxuan-cser/InSynFormer.

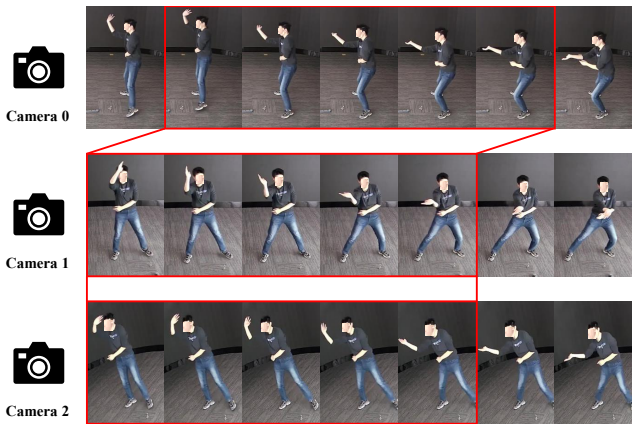

Figure 1: Video clips of an actor performing Tai Chi with a masked face. The clips are captured from three different camera views. Without connected hardware synchronization devices, there is temporal desynchronization in the recordings from different views.

Introduction

In the monocular 3D pose estimation, the occlusion issue has always been challenging to overcome (Cheng et al. 2021; Fang et al. 2018; Cheng et al. 2020). Estimating 3D pose from multiple views (Rhodin, Salzmann, and Fua 2018; Rhodin et al. 2018; Mitra et al. 2020) can solve this problem effectively since the occluded part in one view may become visible in other views (Zheng et al. 2023). Multi-view videos require recording by multiple cameras, but the start-up of multiple cameras may be desynchronized, as shown in Figure 1. However, multi-view 3D pose estimation must simultaneously utilize the pose information. The desynchronized start will cause such tasks to be greatly troubled by temporal desynchronization. The majority of existing solutions use hardware synchronization devices to synchronize cameras. However, as is well-known, hardware synchronization devices are costly. They can not be easily deployed in outdoor scenes or large-scale sports venues, which have limited existing datasets that require synchronized collection for multi-view tasks in small and medium-sized indoor venues with highly constrained scenes. Current synchronization also includes methods based on WiFi signals or Bluetooth signals. These methods are greatly affected by network fluctuations (Wu et al. 2019). Besides, synchronization based on audio is limited by the desynchronization of audio and video. Therefore, video synchronization based on videos is essential.

However, labeling the time intervals between different views is challenging. One intuitive method uses synchronization devices to record synchronized videos and shifts an integral number of frames as the time interval (Wu et al. 2019; Yin et al. 2022; Boizard et al. 2023). However, this method results in discrete labeled time intervals, with lengths that can only be multiples of a single frame. Under natural conditions, the time intervals should be random and continuous, as shown in Figure 2. Especially when estimating the 3D pose in a state of intense movement, the error caused by intra-frame can not be ignored (Shuai et al. 2022).

*Corresponding author

The networks trained on datasets obtained by merely shifting an integral number of frames from synchronized videos are inadequate to satisfy video synchronization task' needs.

In this paper, we utilize multi-view cameras with a stable frame rate, without the connection of hardware synchronization devices, to record many video clips. We also perform precise time interval annotations. Since there is no connection through hardware devices, the time intervals are random and are no longer limited to integer multiples of a single frame duration. Accordingly, we treat inter-frame and intra-frame intervals as a hierarchical classification problem. Through continuous observation, we have set the classification accuracy as 0.1 frame, and the error range caused by classification is within an acceptable range. We design the InSynFormer network utilizing information from 2D poses to synchronize the videos, which adopts an Encoder-Decoder structure. In the Encoder stage, each video frame perceives the change information within its view video clip through intra-view information interaction. In the Decoder stage, each video frame interacts through cross-view information interaction to perceive the time difference between views. Due to the hierarchical relationship between inter-frame and intra-frame intervals, we use the hierarchical cross-entropy loss (Bertinetto et al. 2020) for supervision. Our model achieves the best performance on the IFID dataset, representing a significant improvement compared to existing networks. We have demonstrated the effectiveness of each method separately. We also applied the model to multi-view desynchronized 3D human pose estimation, which significantly helped improve the task performance.

In summary, our contributions are as follows:

- Research on inter-frame and intra-frame time intervals in multi-view videos: We first focus on the inter-frame and intra-frame time intervals of multi-view videos. This perspective provides a new understanding and approach to handling the video synchronization problem.
- Collection of an innovative dataset: We have constructed and released a new dataset with vast potential for various applications. This dataset is the largest one to date and will facilitate further studies in this research direction.
- Proposal of the InSynFormer model with hierarchical cross-entropy loss: We introduced the innovative InSynFormer model that utilizes hierarchical cross-entropy loss for supervision. This model shows exceptional performance in video synchronization tasks.

Related Work

3D Human Pose Estimation

3D human pose estimation aims to predict body joint positions in 3D space, thereby obtaining more comprehensive human body structure information. Monocular pose estimation is relatively simple, inferring the 3D pose of the human body by inputting RGB images from a single view. Some work (Chen and Ramanan 2017; Li and Lee 2019; Martinez et al. 2017; Moreno-Noguer 2017; Tekin et al. 2017) perform 3D pose estimation in two stages: the first stage obtains 2D pose feature, and then 2D to 3D lifting is used to obtain 3D pose in the second stage. On the other hand, other works (Li, Zhang, and Chan 2015; Pavlakos, Zhou, and Daniilidis 2018; Pavlakos et al. 2017) directly predict the 3D human pose from the image. However, neither of these two methods can effectively solve the occlusion problem. Even though graph convolutional networks are introduced to the 3D human pose estimation (Ci et al. 2019; Zhao et al. 2019), the occlusion problem still can not be effectively solved. The natural solution to overcome the occlusion problem is estimating a 3D human pose from multiple views, as the occluded part in one view may be visible in others. Thus, multi-view 3D pose estimation is receiving increasing attention. In the task of multi-view 3D human pose estimation, methods based on convolutional neural network (Dong et al. 2019) and based on attention mechanism (Zhang et al. 2021) have been developed. 3D human pose estimation has made further progress. However, multi-view desynchronized cameras have synchronization errors, and multi-view desynchronized images or videos will severely affect the accuracy of multi-view 3D pose estimation (Shuai et al. 2022). When the human body is in a state of high-speed motion, the errors caused by desynchronization are severe, and even within one frame interval, it can cause a significant impact.

Figure 2: Two different types of frame intervals. Assuming the camera frame rate is 25 fps, the frame interval will be 40 milliseconds. When the time interval is 80 milliseconds, it is the situation in the green area, exactly 2 frames. When the time interval is close to 56 milliseconds, it is the situation in the blue area, approximately 1.4 frames.

Video Temporal Synchronization

Since many years ago, the issue of video synchronization has already attracted researchers' attention. Early methods (Elhayek et al. 2012; Pundik and Moses 2010) mainly relied on traditional digital image features for synchronization, which often required fixed cameras and the input of camera parameters, significantly limiting their practicality. The first work (Wu et al. 2019) that employed deep learning methods used synchronized videos to slide integer frames to obtain desynchronized videos. For the first time, a desynchronized dataset was created. However, the dataset is of a small scale, and its time intervals are all multiples of frame intervals. Even so, this study was the first to view integer frames corresponding to time intervals as a classification problem and achieved satisfactory results. Using convolutional neural networks, features were extracted from two views to get pose features. Then, a bidirectional LSTM layer (Song et al. 2018; Shi et al. 2015) was used to fuse temporal features

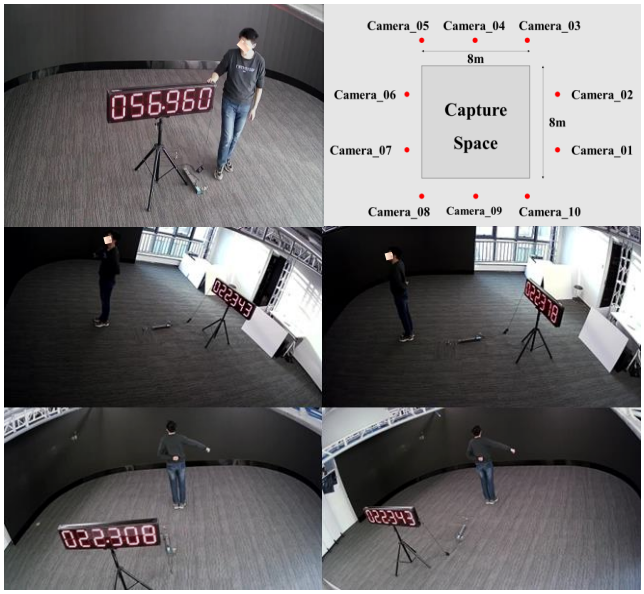

Figure 3: The actor in the top left image turns the timer so that cameras from all views can capture the time on the timer. The top right image shows the ID and position of the cameras in the experiment site. From the four images at the bottom, it is evident that the cameras can capture both the timer reading and the human pose.

before performing the classification. The experiment part of the work also demonstrated that compared to methods based on appearance features and optical flow features, synchronization based on pose features performs better. Therefore, subsequent research mainly adopted the strategy of synchronization based on pose features. As the dataset (Wu et al. 2019) is not released to the public, the NTU RGB+D dataset (Shahroudy et al. 2016) and CMU Panoptic Studio dataset (Joo et al. 2015) was processed to obtain NTU-SYN Dataset and CMU-SYN Dataset (Yin et al. 2022). Unfortunately, the time intervals in the dataset are still integer multiples of the frame interval. Additionally, with the 2D human pose as input for feature embedding, SeSyn-Net is developed, and a series of self-supervised losses are designed to extract the view-invariant effectively but time-discriminative representation for video synchronization. The above two works are the two key works of the existing video time synchronization methods based on human pose features, and the two works have the best performance.

Hierarchical Cross-Entropy Loss

As surveyed in (Silla and Freitas 2011), it is necessary to consider the hierarchy of classes when designing classifiers. Take a simple classification problem as an example: misclassifying a cat as a dog or a flower should incur different penalties because cats and dogs belong to the animal category. Different hierarchical structures can act as prior knowledge to better assist model training. Recently, hierarchical cross-entropy loss based on cross-entropy loss was developed (Bertinetto et al. 2020), which expands each class probability into the chain of conditional probabilities defined by its lineage in a given hierarchy tree, thus effectively improving the cross-entropy loss. In this paper, we consider inter-frame intervals and intra-frame intervals as hierarchical classifications, achieving excellent results.

Method

We first provided a comprehensive definition of the fractional frame video synchronization issue and elucidated our problem. Given our focus on intra-frame intervals in multi-view videos, we have built a relevant experiment site, as shown in Figure 3, and recorded videos using cameras that were not connected through hardware synchronization devices. Relying on a high-precision digital display, we designed a simple and feasible experiment plan and completed the annotation work. Based on the Transformer's attention mechanism (Vaswani et al. 2017), we designed temporal self-attention (TSA) and cross-view attention (CVA) modules and adopted hierarchical cross-entropy loss for supervision. Our model achieved excellent results.

Formulation

The issue of video synchronization is receiving increasing attention. However, up to now, there has not been a complete definition of the problem. Previous work focused on synchronization at inter-frame intervals, while we are the first to address fractional frame interval synchronization. Here, we provide a unified definition for the fractional frame video synchronization problem.

Parameter Definition Given a set of videos $\mathbf{Q} = \{\mathbf{Q_1}, \mathbf{Q_2}, \cdots, \mathbf{Q_N}\}$ collected from N cameras located in the same experimental site. For $\forall i, j \in [1, N] \wedge i \neq j$, we can derive $\mathbf{Q_i}, \mathbf{Q_j} \in \mathbf{Q}$, where $\mathbf{Q_i} = \{Q_i^1, Q_i^2, \cdots, Q_i^{M_1}\}$, $\mathbf{Q_j} = \{Q_j^1, Q_j^2, \cdots, Q_j^{M_2}\}$. M_1 and M_2 are the respective frame counts of the two videos. We use the function $\mathcal{T}$ to represent the world time corresponding to any given frame. When $\{(\mathcal{T}(Q_i^{M_1}) > \mathcal{T}(Q_j^1)) \wedge \{(\mathcal{T}(Q_j^1) > \mathcal{T}(Q_i^1))\}\}$, we say that the two video clips overlap. Using two overlapping videos, we can predict the time interval.

$$\mathcal{T}(Q_i^{k_1}) - \mathcal{T}(Q_j^{k_2}) = f(Q_i^{k_1}, Q_j^{k_2}, \phi), \tag{1}$$

where $k_1 \in [1, M_1], k_2 \in [1, M_2]$. ϕ is the video synchronization model, and f maps the model's prediction results to time intervals.

Technical Details We denote the duration of one frame in a video with a stable frame rate as τ. In previous studies, time intervals have always been an integral multiple of the frame interval, which can be formulated as

$$|\mathcal{T}(Q_i^{k_1}) - \mathcal{T}(Q_j^{k_2})| = k \cdot \tau, k \in \mathbf{N}. \tag{2}$$

In our work, we estimate the time intervals more precisely. The accuracy of our time intervals is $\frac{\tau}{10}$. We obtain the final time interval using the predicted inter-frame interval P_{ex} and the intra-frame interval P_{in}.

$$|\mathcal{T}(Q_i^{k_1}) - \mathcal{T}(Q_j^{k_2})| = P_{ex} \cdot \tau + P_{in} \cdot \frac{\tau}{10}. \tag{3}$$

Additionally, we set $M_1 = M_2 = 5$ for rapid video synchronization with only a limited number of frames.

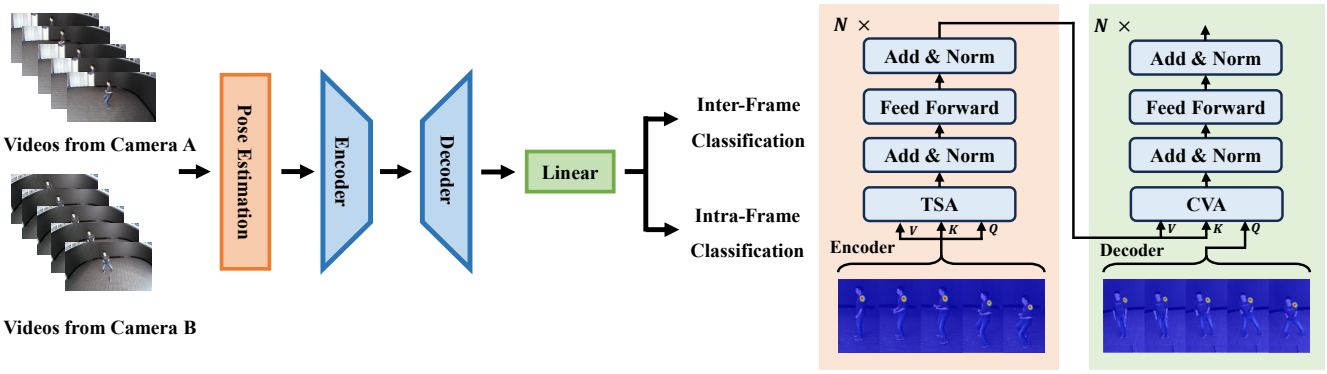

Figure 4: Videos from two different views first pass through the pose estimation network to get pose features. Next, we exchange information within and between views using the Encoder-Decoder architecture. Finally, the linear layer is used to classify the results. The specific structures of the Encoder and Decoder are illustrated on the right side.

IFID Dataset

In a square area with a side length of 8 meters, we deployed 10 Hikvision cameras with a stable frame rate, as shown in Figure 3, enabling us to obtain 45 groups of inter-frame and intra-frame intervals for each experiment. To increase the focus on intra-frame intervals, we set the frame rate to 10 fps to increase the degree of change in human pose between two adjacent frames. To obtain intra-frame intervals, we placed a timer with a precision of 10 ms and brought the current time by reading the timer's display. We reduced the camera's exposure time to 1 ms to record the timer readings. To balance brightness, we used a large aperture on the camera. Sample images taken by the camera can be seen in Figure 3. Each time we start recording, we slowly rotate the timer in a wide range so that cameras from all views can capture the timer's readings, as shown in Figure 3. This operation will help with annotating the intra-frame intervals later. Next, after moving the timer out of the field of view, the actors begin to perform actions. During the annotation, we use the moment when the actor starts to act as the start of the video. Based on the number of frames between the frame when the camera captures the timer reading and the starting frame, we can annotate the time of each frame. In this way, we can obtain the annotations of inter-frame and intra-frame intervals.

InSynFormer

Figure 4 shows the pipeline of the proposed InSynFormer for synchronizing two camera videos. First, we input two video clips from two different views and perform pose estimation separately to obtain the heatmap for each view. Next, we construct temporal self-attention modules in the encoder and cross-view attention modules in the decoder. Through the temporal self-attention module, the model perceives the changes of each joint within the view and further interacts with the pose information of the same joint between different views through the cross-view attention module. We predict the time interval based on each joint and finally fuse the prediction results of each joint.

Temporal Self-Attention Temporal information is essential for perceiving changes in video sequences. Therefore, in the encoder, we adopt temporal self-attention to extract the temporal information of the video sequences fully. The joint heatmap inferred by the pose estimation network undergoes linear embedding to obtain the query, with the key and value computed through the same joint from this view.

$$\mathcal{TSA}(Q_i, K_i, V_i) = \text{softmax}\left(\frac{Q_i K_i^T}{\sqrt{d_k}}\right) V_i. \quad (4)$$

Here, Q_i, K_i, and V_i come from the same view. Q_i is obtained by the continuous 5-frame heatmaps, corresponding to the i-th joint within the view, separately processed through a linear embedding layer. K_i and V_i are also calculated from the i-th joint within the video clip of that view.

Cross-View Attention After processing by TSA, each frame within the same view has already obtained information through intra-view information interaction. However, video synchronization across different views requires information interaction between views to get the final result. Therefore, we propose the cross-view attention method in the decoder to implement information interaction between views. We obtain the key and value from the feature map that has just completed the intra-view feature interaction and calculate them with the query received from the corresponding feature map of another view.

$$\mathcal{CVA}(Q'_i, K_i, V_i) = \text{softmax}\left(\frac{Q'_i K_i^T}{\sqrt{d_k}}\right) V_i. \quad (5)$$

In this case, K_i and V_i are calculated from the feature map of the i-th joint from one view, while Q'_i is calculated from the feature map of the i-th joint from a different view. Through TSA and CVA, joints across different views have sufficiently interacted with the information. Based on the interaction results of all joints, we use a linear layer to merge the feature information of each joint, thereby obtaining the predicted results.

Hierarchical Cross-Entropy Loss

In estimating time intervals, as the intra-frame interval is subordinate to the inter-frame interval, we can represent this

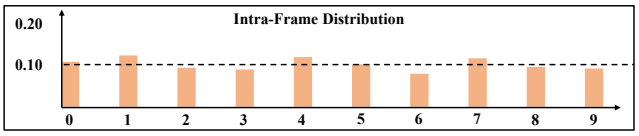

Figure 5: The distribution of the intra-frame interval.

problem using a tree structure. We define the root node of the tree as $C^{(0)} = R$, and $P(R) = 1$. On this basis, the height of the tree is 2. The first layer $C^{(1)} = \{C_1, C_2, ..., C_m\}$ represents the m categories of whole frames corresponding to the range of time intervals. The second layer $C^{(2)} = \{C_{11}, C_{12}, \ldots, C_{mn}\}$ represents the n categories within a frame corresponding to the range of time intervals. The probability of class C_{ij} can be factorised as

$$P(C_{ij}) = P\left(C_i \mid C_0\right) \cdot P\left(C_{ij} \mid C_i\right). \tag{6}$$

The conditionals can conversely be written in terms of the class probabilities as

$$P\left(C_{ij} \mid C_i\right) = \frac{P(C_{ij})}{P(C_i)}. \tag{7}$$

Consequently, we obtain the hierarchical cross-entropy.

$$\mathcal{L}_{\mathrm{HXE}}(P, C) = -\sum_{l=0}^{h-1} \lambda\left(C^{(l)}\right) \log P\left(C^{(l)} \mid C^{(l+1)}\right), \tag{8}$$

where $\lambda(C^{(l)})$ is the weight associated with the edge node $C^{(l+1)} \rightarrow C^{(l)}$. We take the weights as

$$\lambda(C) = \exp(-\alpha h(C)), \tag{9}$$

where $h(C)$ is the height of node C and $\alpha > 0$ is a hyper-parameter that controls the extent to which information is discounted down the hierarchy. As the value of α increases, the loss from classification errors away from the root node decreases, and the model's classification tends to be generic instead of fine-grained. The model achieves excellent results when α is set to 0.1.

Experiments

Dataset

Aside from our dataset, six other datasets are related to our task. Among them, the SYN dataset is derived (Wu et al. 2019) from synchronized videos (Zheng et al. 2017), and the SPVideo dataset and MPVideo dataset are collected (Wu et al. 2019) from synchronized devices. A synchronized video dataset was also collected for video synchronization in the work (Boizard et al. 2023). However, these four datasets are not released to the public. The NTU-SYN and CMU-SYN datasets were processed (Yin et al. 2022) and are now released to the public. We will subsequently introduce these two datasets along with our own and present the results of different methods applied to these three datasets.

NTU-SYN Dataset The NTU-SYN dataset is built based on the NTU RGB+D dataset (Shahroudy et al. 2016). They selected $5,560$ pairs of synchronized videos, including $3,762$ pairs of videos as the training set and $1,798$ as the testing set. For initially synchronized videos, they randomly set a time offset in the $[-30, 30]$ frames for constructing each video pair in both the training and testing sets. We narrow the time offset to $[-5, 5]$ frames for fairness.

CMU-SYN Dataset The CMU-SYN dataset is built based on the CMU Panoptic Studio dataset (Joo et al. 2015). They randomly select 74 pairs of videos to construct the training set and the remaining 32 to construct the testing set. By setting 6 different time offsets for each pair of videos, they finally get 74×6 pairs of videos as the training set, and 32×6 pairs of videos as the testing set, and the range of time offset for each video pair is also $[-30, 30]$ frames. We also narrow the time offset to $[-5, 5]$ frames for fairness.

IFID Dataset The IFID dataset is a naturally collected multi-view desynchronized dataset. We recorded $8,500$ grouped videos, of which $6,092$ were used for training, $1,216$ for validation, and $1,192$ for testing. Since we have 10 views, there are $\mathcal{C}_9^2 = 45$ different combinations in a grouped video, so we have a total of $352,500$ video groups. For each pair of videos, we first synchronize them to the same frame, and the intra-frame interval is still retained. Next, we randomly set a time offset in the $[-5, 5]$ frames range. By placing 2 different time offsets for each pair of videos, we finally get $274,140 \times 2$ pairs of videos as the training set, $54,720 \times 2$ pairs of videos as the validation set, and $53,640 \times 2$ pairs of videos as the testing set. After statistics, our different intra-frame interval categories are approximately uniformly distributed, as shown in Figure 5.

Implementation Details

For all datasets, we resize the length and width of each image to half of the original and feed the image into the model after data augmentation. We use the pre-trained model (Zhou, Wang, and Krähenbühl 2019) for pose estimation and fix the weights during training. After multiple experiments, we found that setting the TSA and CVA module number to 4 can achieve efficient performance. We also supervise the classification of the order of the two views through cross-entropy loss, that is, $\mathcal{L} = \mathcal{L}_{HXE} + \beta \cdot \mathcal{L}_{CE}$, where β is set to 0.1. Since the accuracy is high, we do not elaborate on this. Furthermore, since there is no intra-frame interval in NTU-SYN and CMU-SYN, we only train on the inter-frame intervals with cross-entropy loss.

Evaluation Metrics

Metric: Frame error. The frame error is calculated as $Frm.err. = |(P_{ex} + P_{in}) - (\mathcal{GT}_{ex} + \mathcal{GT}_{in})|$, P_{ex} and $\mathcal{GT}_{ex}$ respectively represent the inter-frame intervals for prediction and ground-truth, while P_{in} and $\mathcal{GT}_{in}$ respectively represent the intra-frame intervals for prediction and ground-truth. And frame accuracy is the proportion of test cases in which inter-frame and intra-frame errors are less than a specified range concerning all cases. $Acc_{ex}@i$ is the

Method	Acc_{ex}@1 ↑	Acc_{ex}@3 ↑	Acc_{in}@1 ↑	Acc_{in}@3 ↑	Acc_{in}@5 ↑	$Frm.err.$ ↓	Para.(MB)	FLOPs(Giga)
SynNet	60.41%	91.37%	31.78%	70.12%	85.62%	1.26	240	238
SeSyn-Net	79.93%	92.44%	–	–	–	0.87	122	196
CNNSiamese	36.42%	76.34%	–	–	–	2.04	**40**	**170**
Ours	**80.86%**	**94.35%**	**61.30%**	**90.69%**	**95.49%**	**0.83**	138.49	212

Table 1: Comparative results of different methods on the IFID.

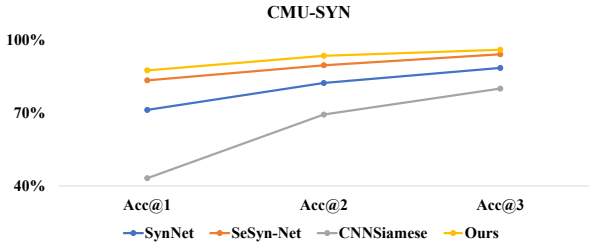

Figure 6: Comparative results of different methods on the CMU-SYN.

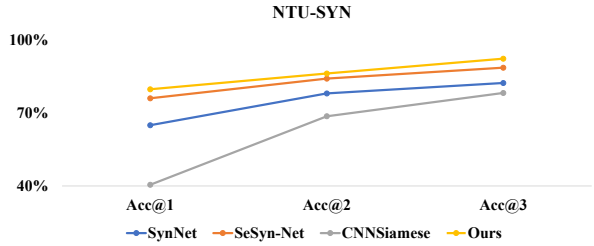

Figure 7: Comparative results of different methods on the NTU-SYN.

number of video pairs satisfying $|P_{ex} - \mathcal{GT}_{ex}| < i$, and $Acc_{in}@j$ is the number of cases satisfying $|P_{in} - \mathcal{GT}_{in}| < j$.

Competing methods

We compare our method with all existing deep learning-based video synchronization methods, including SynNet (Wu et al. 2019), SeSyn-Net (Yin et al. 2022), and CNNSiamese (Boizard et al. 2023). As the comparison can not be carried out directly, we will detail the comparison process.

SynNet This method uses the heatmap obtained from the pose estimation network as input. The model predicts the time interval between two videos after interacting with the temporal information through a bidirectional LSTM (Song et al. 2018; Shi et al. 2015). The model infers the time interval by classifying the number of frame intervals and uses cross-entropy loss for supervision. When comparing the IFID dataset, we added classifiers to predict the intra-frame interval and used hierarchical cross-entropy loss for supervision to standardize the comparison. In comparing the NTU-SYN and CMU-SYN datasets, since there is no intra-frame interval, we only compared the evaluation metrics of inter-frame intervals.

SeSyn-Net This method also uses the results of pose estimation as input. It learns the spatiotemporal information of the pose through ST-GCN (Yan, Xiong, and Lin 2018) and then conducts self-supervised training by matching the features of video frames from different views. Since this method must conduct training and inference on a whole-frame basis, we only compare the evaluation metrics of the inter-frame interval for the IFID dataset. For $Frm.err.$, we only calculate the inter-frame part. Moreover, the length of video clips in NTU-SYN and CMU-SYN are not fixed, and this method does not have requirements for the length of the video clips. To be fair, we set the length of input video clips to 5 for all datasets.

CNNSiamese Unlike the previous ones, this method directly applies a CNN network for feature extraction. After comparing features between video clips from different views, it selects the match with the highest probability. On the IFID dataset, we also only conduct a similar comparison. When testing on the NTU-SYN and CMU-SYN, we uniformly set the length of input videos to 5 frames.

Table 1 shows the comparison results on the IFID dataset. Our methods achieved the best performance. Our model can also predict the intra-frame interval well. Figure 7 and Figure 6 show that our model can also achieve excellent performance when only classifying inter-frame intervals.

Qualitative Results and Visualizations

We recorded the actor performing archery actions in the experiment site using 4 calibrated cameras from different views. Our model sequentially synchronizes the videos based on the ascending order of camera ID from the recorded videos. We separately input the synchronized and original videos and camera parameters into a pre-trained VoxelPose (Tu, Wang, and Zeng 2020) for 3D pose estimation and project the results onto Camera-01. The visualization results are shown in Figure 8. Video synchronization has a significant positive impact on 3D pose estimation from multi-view desynchronized videos. When using desynchronized videos, various issues arise, such as keypoints appearing ahead of time, lagging, or errors in the pose.

Since the input of existing multi-view 3D pose estimation models is synchronized video, we are unable to synchronize the intra-frame part of ours. We mitigate the impact of intra-frame intervals by increasing fps and slowing down the actor's performance speed. In future research, we will focus on leveraging intra-frame intervals to perform 3D pose estimation on desynchronized multi-view videos.

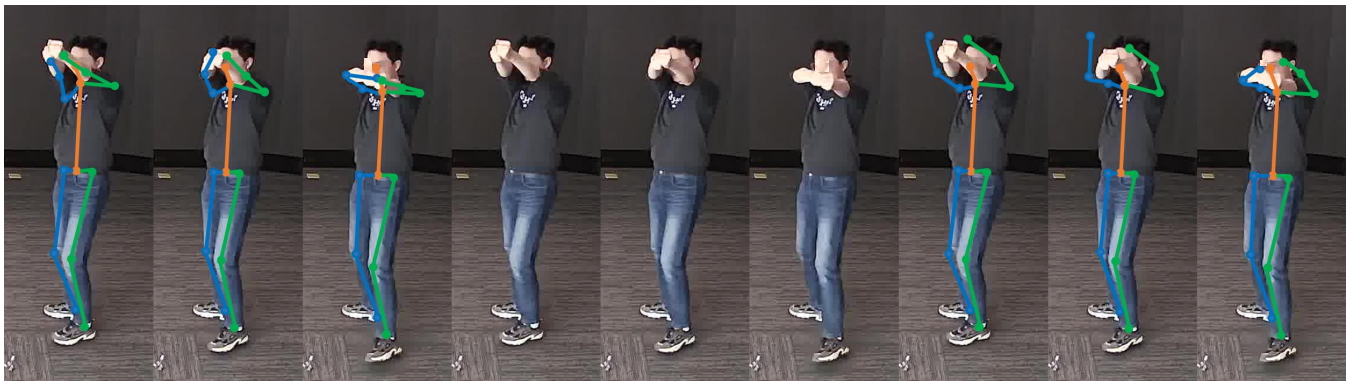

Figure 8: Three consecutive frames of an actor performing an archery action. The three images on the left are the results projected onto Camera-01 after performing 3D pose estimation on the multi-view videos that have been synchronized; in the middle are the original images; on the right are the results obtained directly from the original video synchronization.

Method	Acc_{ex}@1 ↑	Acc_{ex}@3 ↑	Acc_{in}@1 ↑	Acc_{in}@3 ↑	Acc_{in}@5 ↑	$Frm.err.$ ↓
SynNet + $\mathcal{L}_{CE}$	59.78%	90.91%	20.56%	57.84%	79.40%	1.27
SynNet + $\mathcal{L}_{HXE}$	**60.41%**	**91.37%**	**31.78%**	**70.12%**	**85.62%**	**1.26**
InSynFormer + $\mathcal{L}_{CE}$ + MHSA	75.83%	91.10%	19.78%	57.89%	83.58%	1.15
InSynFormer + $\mathcal{L}_{HXE}$ + MHSA	75.88%	91.22%	35.65%	74.23%	87.40%	1.13
InSynFormer + $\mathcal{L}_{CE}$ + TSA + CVA	80.27%	**94.35%**	49.84%	77.62%	92.63%	**0.83**
InSynFormer + $\mathcal{L}_{HXE}$ + TSA + CVA	**80.86%**	**94.35%**	**61.30%**	**90.69%**	**95.49%**	**0.83**

Table 2: The impact of Temporal Self-Attention, Cross-View Attention, and Hierarchical Cross-Entropy Loss on the results.

Ablation Study

Effectiveness of Temporal Self-Attention and Cross-View Attention. Table 2 shows the improvement of using the TSA and CVA module over the traditional MHSA method. For a fair comparison, we ensure that the number of MHSA modules equals the sum of the TSA and CVA modules when conducting the experiments. For the MHSA method, we concatenate clips from different views and directly feed them into the MHSA layer. It is not sufficient for cross-view information interaction. Moreover, the model learns less information about intra-frame intervals. The performance has improved significantly by changing the MHSA module to TSA and CVA modules.

Effectiveness of Hierarchical Cross-Entropy loss. Table 2 also shows the different results of using hierarchical cross-entropy loss and cross-entropy loss. The hierarchical relationship between inter-frame and intra-frame intervals is ignored when training with a regular CE loss. However, if they are added together by weight, the training effect of inter-frame interval is good, while the training effect of intra-frame interval declines. HXE loss effectively uses the hierarchical relationship to ensure good training of both.

Generalization Evaluation. We collected nearly 100 sets of videos in outdoor scenes using the same approach. We tested the model's performance, and the specific details will be elaborated upon in the supplementary materials.

Conclusion

This paper focuses on the intra-frame synchronization problem of multi-view cameras under natural shooting conditions. We first provided a unified definition for Fractional Frame Video Synchronization. To address this problem, we constructed the IFID dataset, which annotates inter-frame and intra-frame time intervals. We also proposed the InSynFormer model, which demonstrates excellent performance. Through ablation studies, we verified the effectiveness of the designed modules and loss function. In the future, we plan to combine our synchronization model to develop a multi-view 3D pose estimation algorithm, thereby allowing for 3D pose estimation from videos that have been recorded without being connected through hardware synchronization devices.

Limitations Our work has three main limitations. Firstly, although we classify frame intervals with finer granularity, a slight error still exists that cannot be eliminated. Ideally, the video synchronization model should be able to predict the time intervals directly. Secondly, like most previous works, we rely on human pose information in the videos for synchronization. If no people are in the video, video synchronization can not occur. Lastly, current multi-view 3D pose estimation models cannot process desynchronized videos, which also results in the intra-frame interval not effectively utilized. Our future work will primarily address these issues.

Acknowledgements

The Natural Science Foundation of China partly supports this work under Grant 62076238 and Grant 62222606.

References

Bertinetto, L.; Mueller, R.; Tertikas, K.; Samangooei, S.; and Lord, N. A. 2020. Making Better Mistakes: Leveraging Class Hierarchies with Deep Networks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 12506–12515.

Boizard, N.; El Haddad, K.; Ravet, T.; Cresson, F.; and Dutoit, T. 2023. Deep Learning-Based Stereo Camera Multi-Video Synchronization. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, 1–5.

Chen, C.-H.; and Ramanan, D. 2017. 3D Human Pose Estimation = 2D Pose Estimation + Matching. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7035–7043.

Cheng, Y.; Wang, B.; Yang, B.; and Tan, R. T. 2021. Graph and Temporal Convolutional Networks for 3D Multi-person Pose Estimation in Monocular Videos. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 1157–1165.

Cheng, Y.; Yang, B.; Wang, B.; and Tan, R. T. 2020. 3D Human Pose Estimation Using Spatio-Temporal Networks with Explicit Occlusion Training. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 10631–10638.

Ci, H.; Wang, C.; Ma, X.; and Wang, Y. 2019. Optimizing Network Structure for 3D Human Pose Estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2262–2271.

Dong, J.; Jiang, W.; Huang, Q.; Bao, H.; and Zhou, X. 2019. Fast and Robust Multi-Person 3D Pose Estimation From Multiple Views. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 7792–7801.

Elhayek, A.; Stoll, C.; Kim, K. I.; Seidel, H.-P.; and Theobalt, C. 2012. Feature-Based Multi-video Synchronization with Subframe Accuracy. In *Pattern Recognition*, 266–275.

Fang, H.-S.; Xu, Y.; Wang, W.; Liu, X.; and Zhu, S.-C. 2018. Learning Pose Grammar to Encode Human Body Configuration for 3D Pose Estimation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 6821–6828.

Joo, H.; Liu, H.; Tan, L.; Gui, L.; Nabbe, B.; Matthews, I.; Kanade, T.; Nobuhara, S.; and Sheikh, Y. 2015. Panoptic Studio: A Massively Multiview System for Social Motion Capture. In *Proceedings of the IEEE International Conference on Computer Vision*, 3334–3342.

Li, C.; and Lee, G. H. 2019. Generating Multiple Hypotheses for 3D Human Pose Estimation with Mixture Density Network. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 9887–9895.

Li, S.; Zhang, W.; and Chan, A. B. 2015. Maximum-Margin Structured Learning With Deep Networks for 3D Human Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2848–2856.

Martinez, J.; Hossain, R.; Romero, J.; and Little, J. J. 2017. A Simple yet Effective Baseline for 3D Human Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 2640–2649.

Mitra, R.; Gundavarapu, N. B.; Sharma, A.; and Jain, A. 2020. Multiview-Consistent Semi-Supervised Learning for 3D Human Pose Estimation. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 6907–6916.

Moreno-Noguer, F. 2017. 3D Human Pose Estimation From a Single Image via Distance Matrix Regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2823–2832.

Pavlakos, G.; Zhou, X.; and Daniilidis, K. 2018. Ordinal Depth Supervision for 3D Human Pose Estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7307–7316.

Pavlakos, G.; Zhou, X.; Derpanis, K. G.; and Daniilidis, K. 2017. Coarse-To-Fine Volumetric Prediction for Single-Image 3D Human Pose. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 7025–7034.

Pundik, D.; and Moses, Y. 2010. Video Synchronization Using Temporal Signals from Epipolar Lines. In *Proceedings of the European Conference on Computer Vision*, 15–28.

Rhodin, H.; Salzmann, M.; and Fua, P. 2018. Unsupervised Geometry-Aware Representation for 3D Human Pose Estimation. In *Proceedings of the European Conference on Computer Vision*, 750–767.

Rhodin, H.; Spörri, J.; Katircioglu, I.; Constantin, V.; Meyer, F.; Müller, E.; Salzmann, M.; and Fua, P. 2018. Learning Monocular 3D Human Pose Estimation From Multi-View Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8437–8446.

Shahroudy, A.; Liu, J.; Ng, T.-T.; and Wang, G. 2016. NTU RGB+D: A Large Scale Dataset for 3D Human Activity Analysis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1010–1019.

Shi, X.; Chen, Z.; Wang, H.; Yeung, D.-Y.; Wong, W.-K.; and Woo, W.-c. 2015. Convolutional LSTM Network: A Machine Learning Approach for Precipitation Nowcasting. In *Advances in Neural Information Processing Systems*, 1–9.

Shuai, Q.; Geng, C.; Fang, Q.; Peng, S.; Shen, W.; Zhou, X.; and Bao, H. 2022. Novel View Synthesis of Human Interactions from Sparse Multi-view Videos. In *SIGGRAPH Conference Proceedings*, 1–10.

Silla, C. N.; and Freitas, A. A. 2011. A survey of hierarchical classification across different application domains. *Data Mining and Knowledge Discovery*, 22: 31–72.

Song, H.; Wang, W.; Zhao, S.; Shen, J.; and Lam, K.-M. 2018. Pyramid Dilated Deeper ConvLSTM for Video Salient Object Detection. In *Proceedings of the European Conference on Computer Vision*, 715–731.

Tekin, B.; Márquez-Neila, P.; Salzmann, M.; and Fua, P. 2017. Learning to Fuse 2D and 3D Image Cues for Monocular Body Pose Estimation. In *Proceedings of the IEEE International Conference on Computer Vision*, 3941–3950.

Tu, H.; Wang, C.; and Zeng, W. 2020. VoxelPose: Towards Multi-camera 3D Human Pose Estimation in Wild Environment. In *Proceedings of the European Conference on Computer Vision*, 197–212.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, 1–11.

Wu, X.; Wu, Z.; Zhang, Y.; Ju, L.; and Wang, S. 2019. Multi-Video Temporal Synchronization by Matching Pose Features of Shared Moving Subjects. In *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, 2729–2738.

Yan, S.; Xiong, Y.; and Lin, D. 2018. Spatial Temporal Graph Convolutional Networks for Skeleton-Based Action Recognition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 7444–7452.

Yin, L.; Han, R.; Feng, W.; and Wang, S. 2022. Self-Supervised Human Pose based Multi-Camera Video Synchronization. In *Proceedings of the 30th ACM International Conference on Multimedia*, 1739–1748.

Zhang, J.; Cai, Y.; Yan, S.; Feng, J.; et al. 2021. Direct Multi-view Multi-person 3D Pose Estimation. In *Advances in Neural Information Processing Systems*, 13153–13164.

Zhao, L.; Peng, X.; Tian, Y.; Kapadia, M.; and Metaxas, D. N. 2019. Semantic Graph Convolutional Networks for 3D Human Pose Regression. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 3425–3435.

Zheng, C.; Wu, W.; Chen, C.; Yang, T.; Zhu, S.; Shen, J.; Kehtarnavaz, N.; and Shah, M. 2023. Deep Learning-Based Human Pose Estimation: A Survey. *ACM Computing Surveys*, 111: 1–36.

Zheng, K.; Fan, X.; Lin, Y.; Guo, H.; Yu, H.; Guo, D.; and Wang, S. 2017. Learning View-Invariant Features for Person Identification in Temporally Synchronized Videos Taken by Wearable Cameras. In *Proceedings of the IEEE International Conference on Computer Vision*, 2858–2866.

Zhou, X.; Wang, D.; and Krähenbühl, P. 2019. Objects as Points. *arXiv preprint arXiv:1904.07850*.