VIEW-INDEPENDENT VIDEO SYNCHRONIZATION FROM TEMPORAL SELF-SIMILARITIES

Emilie Dexter, Patrick Pérez, Ivan Laptev and Imran N. Junejo
IRISA/INRIA - Rennes Bretagne Atlantique, Campus Universitaire de Beaulieu, 35042 Rennes Cedex, France

Keywords: Video synchronization, Temporal alignment, Self-similarities.

Abstract: This paper deals with the temporal synchronization of videos representing the same dynamic event from different viewpoints. We propose a novel approach to automatically synchronize such videos based on temporal self-similarities of sequences. We explore video descriptors which capture the structure of video similarity over time and remain stable under viewpoint changes. We achieve temporal synchronization of videos by aligning such descriptors by Dynamic Time Warping. Our approach is simple and does not require point correspondences between views while being able to handle strong view changes. The method is validated on two public datasets with controlled view settings as well as on other videos with challenging motions and large view variations.

1 INTRODUCTION

The number and the variety of video recording devices has exploded in recent years from professional cameras towards digital cameras and mobile phones. As one consequence of this development, the simultaneous footage of the same dynamic scenes becomes increasingly common for example for sport events and public performances. The recorded videos of the same event often differ in viewpoints and camera motion, hence, providing information that can be exploited, e.g., for novel view synthesis and reconstruction of dynamic scenes or for search in video archives. Synchronization of such videos is the first challenging and important step to enable such applications.

In the past, video synchronization was mostly addressed under assumptions of stationary cameras and linear time transformations. Some works explore estimations of spatial and temporal transformations between two videos (Stein, 1999; Caspi and Irani, 2002; Ukrainitz and Irani, 2006) while others focus only on temporal alignment (Rao et al., 2003; Carceroni et al., 2004; Wolf and Zomet, 2006; Ushizaki et al., 2006).

In the literature, the majority of approaches exploit spatial correspondences between views either to estimate the fundamental matrix (Caspi and Irani, 2002) or to use rank constraints on observation matrices as in (Wolf and Zomet, 2006). In contrast, other methods try to extract temporal features without correspondences as in (Ushizaki et al., 2006) where authors investigate an image-based temporal feature of image sequence for synchronization. The time-shift is estimated by evaluating the correlation between temporal features.

Finally, a few papers deal with synchronization of moving cameras and to the best of our knowledge, none of these addresses automatic synchronization. For example in (Tuytelaars and Van Gool, 2004), authors choose manually the 5 independently moving points because these points have to be tracked successfully along all the sequences.

In this work, we address automatic synchronization of videos of the same dynamic event without correspondences between views or assumptions on the time-warping function. We explore a novel temporal descriptor of videos, fairly stable under view changes based on temporal self-similarities. Synchronization is achieved by aligning descriptors by dynamic programming.

1.1 Related Work

Our work is most closely related to the methods of (Cutler and Davis, 2000; Benabdelkader et al., 2004; Shechtman and Irani, 2007). The notion of self-similarity is exploited by (Shechtman and Irani, 2007) to match images and videos or to detect actions in

Dexter E., Pérez P., Laptev I. and N. Junejo I. (2009).
VIEW-INDEPENDENT VIDEO SYNCHRONIZATION FROM TEMPORAL SELF-SIMILARITIES.
In *Proceedings of the Fourth International Conference on Computer Vision Theory and Applications*, pages 383-391
DOI: 10.5220/0001561003830391

videos. They compute a local patch descriptor for every pixel by correlating the patch centered at a pixel with its neighborhood. Matching a template image or an action to another is achieved by finding a similar set of descriptors.

The notion of temporal self-similarity we explore in this paper is more related to the works of (Cutler and Davis, 2000; Benabdelkader et al., 2004). The authors construct a similarity matrix where each entry is the absolute correlation score between silhouettes of moving objects for all pairs of frames. This matrix is used respectively for periodic motion detection and gait recognition.

Our method is also related to the approach of (Rao et al., 2003) by the use of dynamic programming. In their work, the authors evaluate temporal alignment by including Dynamic Time Warping (DTW) and rank constraints on observation matrices which allows the use of non-linear time warping functions between time axes of videos. In contrast to this work we do not rely on spatial correspondences between image sequences which are hard to obtain in practice.

1.2 Our Approach

In this paper, we propose a novel approach to automatically synchronize videos of the same dynamic event recorded from substantially different, static viewpoints. In contrast to the majority of existing methods, we do not impose restrictive assumptions as sufficient background information, point correspondences between views or linear modeling of the temporal misalignment.

We explore self-similarity matrices (SSM) as a temporal descriptor of video sequences as recently proposed for action recognition in (Junejo et al., 2008). Although SSMs are not strictly view-invariant, they are fairly stable under view changes as illustrated in Fig. 1(b,d) where SSMs computed for different views of a golf swing action have a striking similarity despite the difference in the projections depicted in Fig. 1(a,c). The intuition behind this claim is the following. If configurations of a dynamic event are similar at moments t_1 and t_2, the value of SSM(t_1, t_2) will be low for any view of that event. On the contrary, if configurations are different at t_1 and t_2 the value of SSM(t_1, t_2) is likely to be large for most of the views. Fig. 1 illustrates this idea. Corresponding SSMs are computed using distances of points on the hand trajectory illustrated in Fig. 1(a,c). We can observe that close trajectory points A, B remain close in both views while the distant trajectory points A and C have large distances in both projections.

As a result, the same dynamic event produces similar self-similarity matrices where time axes can be matched by estimating a time-warping transformation. Furthermore, we suggest aligning sequence descriptors by DTW in order to obtain exhaustive time correspondences between videos.

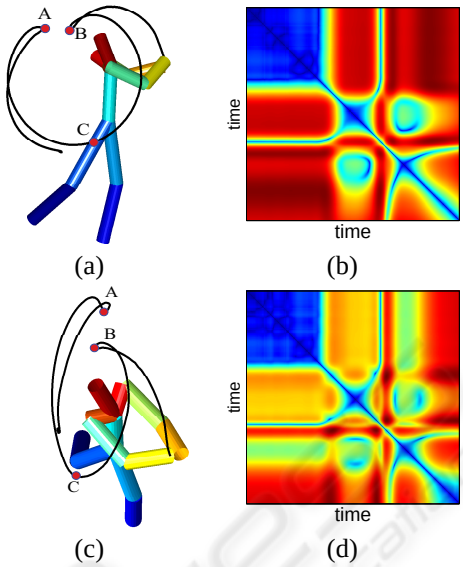


Figure 1: (a) and (c) demonstrate a golf swing action seen from two different views, and (b) and (d) represent their computed self-similarity matrices (SSMs) based on 2D point trajectories. Even though the two views are different, the structures of the patterns of computed SSMs are similar.

The remainder of this paper is organized as follows: Section 2 introduces self-similarity descriptors of videos. Section 3 describes descriptor alignment based on dynamic programming. In Section 4 we demonstrate results of video synchronization for two public datasets as well as for our own challenging videos.

2 VIDEO DESCRIPTORS

In this section, we introduce the temporal description of videos. First, we describe the computation and the properties of self-similarity matrices. Then a local descriptor for SSM is proposed for synchronization.

2.1 Self-similarity Matrices

Our main hypothesis is that similarities and dissimilarities are preserved under view changes. As a result, the same dynamic event recorded from different views should produce similar structures or patterns of self-similarity, enabling subsequent video synchronization.

For a sequence of images $\mathcal{I} = \{I_1, I_2, \dots I_T\}$, lying in discrete (x,y,t)-space, the SSM is the square symmetric distance matrix $\mathcal{D}(\mathcal{I})$ lying in $\mathbb{R}^{T\times T}$ defined as an exhaustive table of distances between image features taken by pair from the set $\mathcal{I}$:

$$\mathcal{D}(\mathcal{I}) = [d_{ij}] = \begin{bmatrix} 0 & d_{12} & \dots & d_{1T} \\ d_{21} & 0 & \dots & d_{2T} \\ \vdots & \vdots & & \vdots \\ d_{T1} & d_{T2} & \dots & 0 \end{bmatrix} \quad (1)$$

where d_{ij} represents a distance between some features extracted from frames I_i and I_j respectively. The diagonal corresponds to comparing an image to itself, hence, is always zero.

The structure or the patterns of the matrix $\mathcal{D}(\mathcal{I})$ is determined by features and distance measure used to compute its entries. Different features produce matrices with different characteristics.

In this work, we use the Euclidean distance to compute the entries d_{ij} of the matrix for features extracted from image sequence. This form of $\mathcal{D}(\mathcal{I})$ is known as the Euclidean Distance Matrix (EDM) (Lele, 1993). We have considered two different types of features to compute $\mathcal{D}(\mathcal{I})$: point trajectories and image-based features.

2.1.1 Trajectory-based Self-similarities

We first consider trajectory-based similarities where we track points of the moving object. Entries d_{ij} are expressed as the Euclidean distance between the positions of the tracked points for a pair of frames. The similarity measure between points tracked in the frames I_i and I_j can be computed as:

$$d_{ij} = \sum_k \|x_i^k - x_j^k\|_2 \quad (2)$$

where k indicates the point being tracked, and i and j indicate the frame numbers in the sequence $\mathcal{I}$. These point trajectory features are used in our experiments on the motion capture (MoCAP) dataset presented in Section 4.1 where the tracked points correspond to joints on the human body. We denote this computed matrix by SSM-pos.

2.1.2 Image-based Self-similarities

In addition to the trajectory-based self-similarities, we also propose to use image-based features. In this regard, we use optical flow vectors or Histogram of Oriented Gradients (HoG) features (Dalal and Triggs, 2005) to estimate $\mathcal{D}(\mathcal{I})$.

In our experiments, the optical flow is calculated using the method proposed by Lucas and Kanade (Lucas and Kanade, 1981) either on bounding box centered around the the foreground object for the public image sequence dataset or on the entire image for realistic videos. The global optical flow vector is obtained by concatenating flows in both directions.

In contrast to optical flow vectors which express motions, HoG features, originally used to perform human detection, characterize the local shape by capturing edge and gradient structures. Our implementation uses 4 bin histograms for each 5 x 7 blocks defined on a bounding box around a foreground object for the public image sequence dataset or on the entire image in each frame of realistic videos.

For both features, d_{ij} is the Euclidean distance between two vectors corresponding to the frames I_i and I_j. The SSMs computed by HoG features and optical flow vectors are respectively denoted SSM-hog and SSM-of.

2.2 Descriptor

As mentioned above, SSM is symmetric positive semidefinite matrix with zero-value diagonal and has view-stable structure. For video synchronization, we need to capture this structure and consequently construct appropriate descriptors.

We opt for a local representation to describe the self-similarity matrices after observing their properties. Indeed, global structures of SSM can be influenced by changes in temporal offsets and time warping. Furthermore the uncertainty of values increases with the distance from the diagonal due to the increasing difficulty of measuring self-similarity over long time intervals.

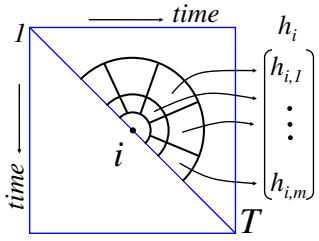


Figure 2: Local descriptors of an SSM are centered at every diagonal point $i = 1...T$ and rely on a log-polar block structure. Histograms of gradient directions are computed separately for each block and concatenated into descriptor vector h_i.

As shown in Fig. 2, we compute at each diagonal point a descriptor based on a log-polar block structure. We construct a 8-bin histogram of gradient directions for each of 11 blocks and concatenate the normalized histograms into a descriptor vector h_i corresponding to the frame number i. Finally the video sequence is represented by the sequence of such descriptors $H = (h_1, ..., h_T)$ computed for all diagonal

elements of the SSM.

3 DESCRIPTOR ALIGNMENT

We aim to align the temporal descriptors extracted from the self-similarity matrices by a classical DTW algorithm. Such an approach, which was introduced for warping two temporal signals in particular for speech recognition (Rabiner et al., 1978), is well-adapted to our problem of descriptor alignment.

Given two image sequences I^1 and I^2 of the same dynamic event seen from different view-points, we compute SSMs and the corresponding global descriptors, H^1 and H^2. We denote h_i^1 the local descriptor of I^1 for the frame i. I^1 and I^2 have respectively N and M frames.

The DTW algorithm aims to estimate the warping function w between time axes of the two videos. The warping between frames i and j of both sequences is expressed as $j = w(i)$.

Given a dissimilarity measure S, where a smaller value of $S(h_i^1, h_j^2)$ indicates greater similarity between h_i^1 and h_j^2, we define the cost matrix $\mathcal{C}$ as

$$\mathcal{C} = [c_{ij}] = \left[S(h_i^1, h_j^2)\right]. \quad (3)$$

Each entry of this matrix measures the cost of alignment between frames i and j of both sequence descriptors. The best temporal alignment is the set of pairs $\{(i, j)\}$ which contributes to the global minimum similarity measure. As a consequence, the optimal warping w must minimize the accumulated cost C_T:

$$C_T = \min_w \sum_{i=1}^{N} S(h_i^1, h_{w(i)}^2) \quad (4)$$

To solve (4) using dynamic programming, we must construct the accumulated cost matrix C_A from the cost matrix $\mathcal{C}$. Considering three possible moves (horizontal, vertical and diagonal) in $\mathcal{C}$ for the warping, we can recursively compute, for each pair of frames (i, j), $C_A(h_i^1, h_j^2)$ by

$$\begin{aligned} C_A(h_i^1, h_j^2) &= c_{ij} + \min[C_A(h_{i-1}^1, h_j^2), \\ &\quad C_A(h_{i-1}^1, h_{j-1}^2), C_A(h_i^1, h_{j-1}^2)] \end{aligned} \quad (5)$$

Vertical and horizontal moves correspond to associating one frame in a sequence to two consecutive frames in the other sequence whereas diagonal one amounts to associating two pairs of consecutive images.

The final solution C_T of (4) is by definition $C_T = C_A(h_N^1, h_M^2)$. The warping function, w, is obtained by tracing back from the pair of frames (N, M) the optimal path in the accumulated cost matrix C_A. Finally, if the pair of frames (i, j) belongs to the path, it means that the i^{th} frame of the first sequence I^1 temporally corresponds to the j^{th} frame of the second sequence I^2.

As mentioned above, DTW algorithm requires a distance measure $S(\cdot, \cdot)$ to evaluate the alignment cost. We try different distances, including the one proposed by (Cha and Srihari, 2002) for histograms. However cost matrices are extremely similar for our descriptors. So, we choose the Euclidean distance to measure the similarity between descriptors H^1 and H^2.

4 SYNCHRONIZATION RESULTS

In this section, we present various results on video synchronization. The first experiments in Section 4.1 and in Section 4.2 aim to validate the method in controlled multi-view settings using: (i) motion capture (MoCAP) datasets, and (ii) a public image sequence dataset (Weinland et al., 2007). We finally demonstrate synchronization results on realistic videos in Section 4.3.

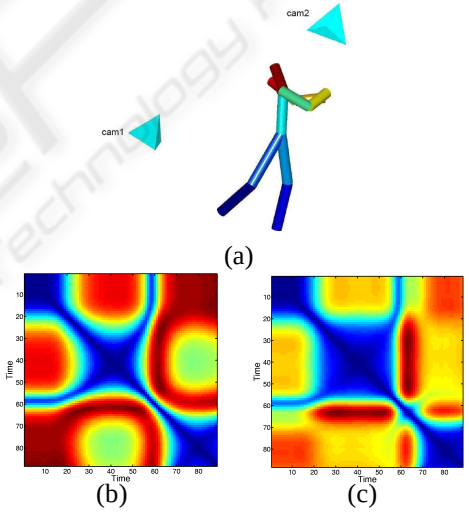


Figure 3: (a) A person figure animated from the CMU motion capture dataset and two virtual cameras used to simulate projections in our experiments. (b) SSM corresponding to cam1. (c) SSM corresponding to cam2.

4.1 Synchronization on CMU MoCAP Dataset

We have used 3D MoCAP data from the CMU dataset (*mocap.cs.cmu.edu*) to simulate multiple and controlled view settings of the same dynamic action. Trajectories of 13 points on the human body were projected to two cameras with pre-defined orientations with respect to the human body as illustrated in Fig. 3(a). We need to remove the effect of translation

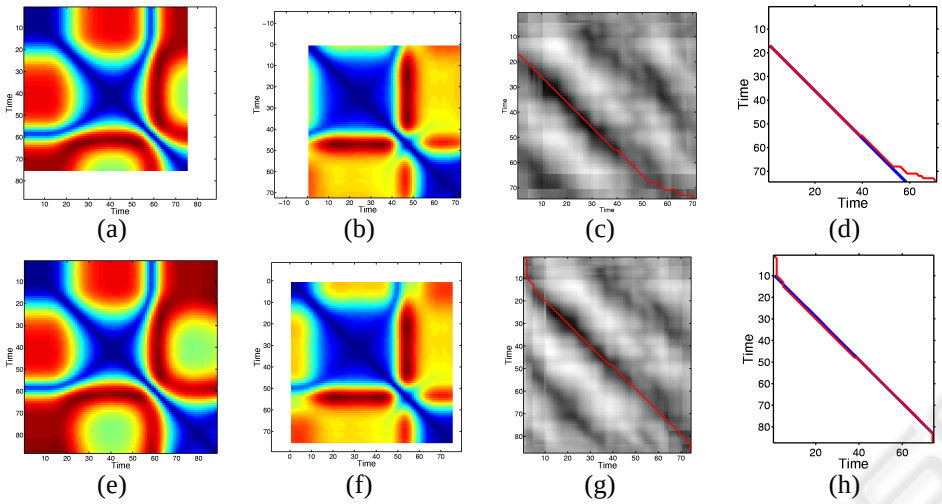


Figure 4: Synchronization of SSMs for simulated time-shift. (a-d) Synchronization for sequences with overlapping time intervals. (a) Truncated SSM for cam1. (b) Truncated SSM for cam2. (c) Cost matrix representation with the time transformation estimation (red curve). (d) The time transformation estimation recovers the ground truth transformation (blue curve). (e-h) Synchronization for sequences with the time interval of one sequence which is contained in the time interval of the second. (e) Original SSM for cam1. (f) Truncated SSM for cam2. (g) Cost matrix representation with the time transformation estimation (red curve). (h) The time transformation estimation recovers the ground truth transformation (blue curve).

and scale such that the points are zero-centered.The points are normalized by $\mathbf{x}_i = \frac{\mathbf{x}'_i}{\|\mathbf{x}'_i\|}$, where $\mathbf{x}'_i$ corresponds to the joints being tracked in frame i and $\mathbf{x}_i$ corresponds to their normalized coordinates. An example of the computed SSMs for these two projections are proposed in Fig. 3(b,c).

For this dataset, trajectory-based SSMs can be computed and synchronized in presence of simulated temporal misalignment. We choose to apply the simplest time transformation: the time-shift. We simply truncate SSMs in order to simulate time-shift. Two cases are possible: time intervals of both sequences overlap or the time interval of one sequence is contained in the time interval of the second. In the first case, one SSM is truncated at the beginning and the second at the end. In the second case, only one SSM is truncated at the beginning and the end.

Fig. 4 illustrates synchronization of both of these cases for the SSMs shown in Fig. 3. We temporally align descriptors of truncated SSMs by estimating the optimal path in the cost matrix with the Dynamic Time Warping algorithm. The optimal path or estimated time transformation, represented by red curves, recovers almost perfectly the ground truth transformation for both proposed examples corresponding to the truncated SSMs in the Fig. 4(d,h). These experiments with controlled view settings validate our framework of video synchronization when time-warping function is a simple time-shift.

4.2 Synchronization on IXMAS Dataset

Experiments were also conducted using real image sequences from the public IXMAS dataset (Weinland et al., 2007). This dataset has 5 synchronized views of 10 different actors performing 11 classes of actions three times. Positions and orientations are freely chosen by actors. An illustration of this dataset is depicted in Fig. 5.

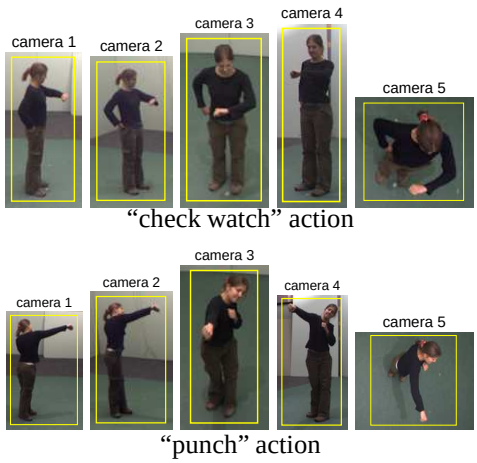


Figure 5: Example frames for two action classes and five views of the IXMAS dataset.

For this dataset, we compute image-based features on bounding boxes around the actors. The boxes are extracted from silhouettes available for each frame of

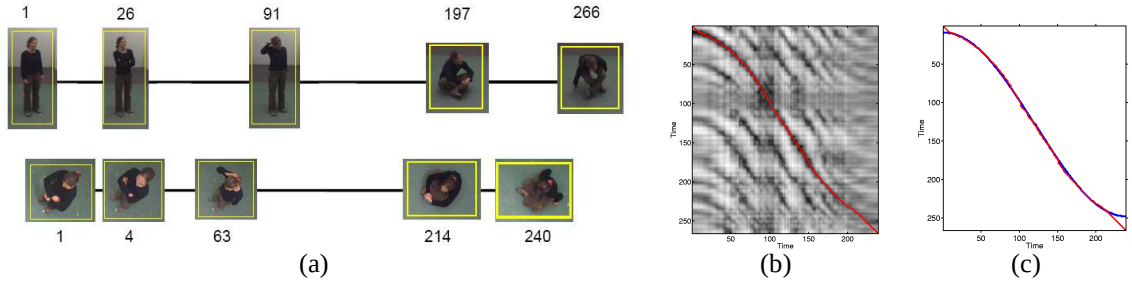


Figure 6: Synchronization of nonlinearly time warped sequences. (a) Sequences with very different view conditions are represented by key-frames. (b) Cost matrix for computed descriptors with the time transformation estimation (red curve).(c) Synchronization result where the time transformation estimation recovers almost completely the ground truth transformation (blue curve).

this dataset. Then, we enlarge and resize bounding boxes in order to avoid border effect in the optical flow computation and to ensure the same size of features along the sequence. We resize the height to a value equal to 150 pixels and the width is set to the largest value for the considered sequence. For HoG features, we use 4 bin histograms for each 5 x 7 block defined on the bounding box.

As mentioned above, sequences of this dataset are synchronized. As a consequence, we must simulate temporal misalignment. Furthermore, sequences of this dataset, originally used to perform action recognition, can be considered either action-by-action or as long sequences composed of several successive actions. In this paper, we propose only experimental results for an example of long sequence. However, for action-by-action sequences, we can apply the same misalignment method as for the MoCAP dataset.

For long image sequences, we can further challenge the synchronization by applying a nonlinear time transformation to one of the sequences in addition to the time-shift. The time of one sequence is warped by $t' = a\cos(bt)$. An example of synchronization for this warping form is depicted in the Fig. 6(c) where the estimated time transformation is illustrated by the red curve and does almost perfectly recover the ground truth transformation (blue curve) despite the drastic view variation between image sequences seen in Fig. 6(a).

We notice that the beginning and the end of the estimated time transformation do not correspond exactly with the ground truth. This is due to the fact that DTW estimation assumes, wrongly, that the admissible paths end at (N,M). Despite the false correspondences that this constraint causes, the algorithm is able to recover a large part of the ground truth. However, these results demonstrate that our approach supports linear and nonlinear time transformations even under drastic view variations between image sequences.

4.3 Synchronization on Natural Videos

We have tested the proposed framework to synchronize realistic videos with moving objects or human activities. For these image sequences, we compute optical flow between consecutive frames and estimate corresponding self-similarity matrices and descriptors.

4.3.1 Sequence with Moving Objects

In the first experiment we have used videos of moving objects as illustrated in Fig. 8(a). The sequences represent two balls bouncing on a table from two different viewpoints: a top view and a side view. An illustration of the scene configuration is proposed in Fig. 7 where green and purple curves represent ball trajectories. Synchronization results for this pair of sequences are presented in Fig. 8(c). The original transformation between the two videos, which is a time-shift, is partially recovered. In fact, at the beginning and at the end of both sequences, there is no motion which leads to misalignment due to the lack of temporal information.

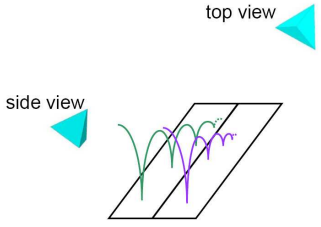
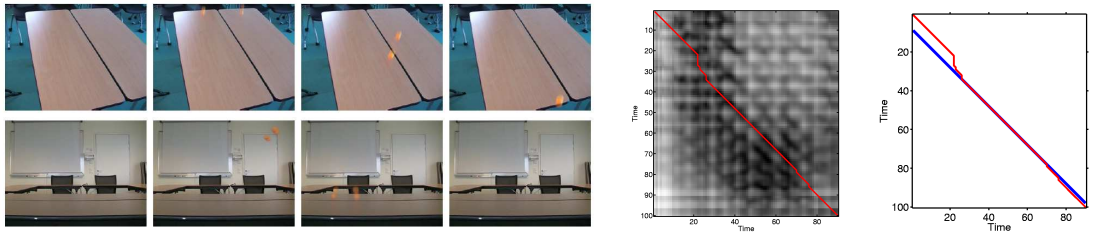


Figure 7: Scene configuration of the videos with two balls bouncing on a table.

However, our approach has difficulties for periodic motion such as walking or running. Indeed,

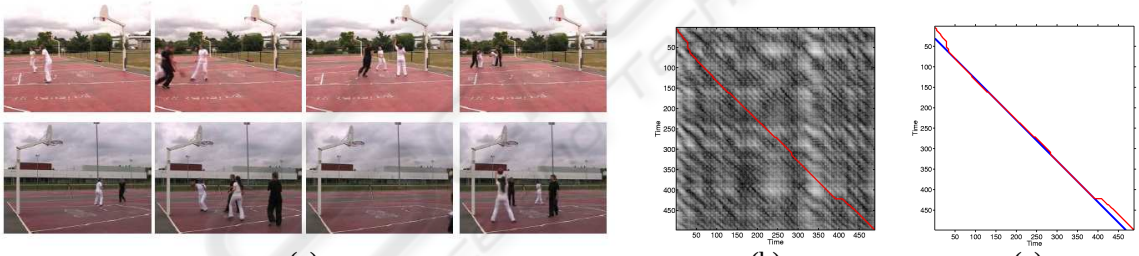


(a) (b) (c)

Figure 8: Synchronization of videos with moving objects. (a) Two balls bounce on a table seen from the top (upper row) and from the side (lower row). (b) Cost matrix with the time transformation estimation (red curve). (c) Synchronization result with the time transformation estimation and the ground truth in blue.

(a) (b) (c)

Figure 9: Synchronization of videos of basketball with two players. (a) The upper row represents the first side view whereas the lower row represents the opposite view. The players always appear in the field of view of cameras. (b) Cost matrix with the time transformation estimation (red curve). (c) DTW estimation recovers the original transformation (blue curve).



(a) (b) (c)

Figure 10: Synchronization of videos of basketball with four players.(a) The upper row represents the first side view whereas the lower row represents the opposite view. The players can appear and disappear along the sequences. (b) Cost matrix with the time transformation estimation red curve). (c) DTW estimation recovers the original transformation (blue curve).

periodic motions induce periodic structures in corresponding self-similarity matrices and cause ambiguities for the DTW algorithm. When motion is almost periodic as in Fig. 8(a), the performance of our approach depends on the length of the time-shift. As in our example the time-shift is short, ambiguities are limited.

4.3.2 Sequences with Human Activities

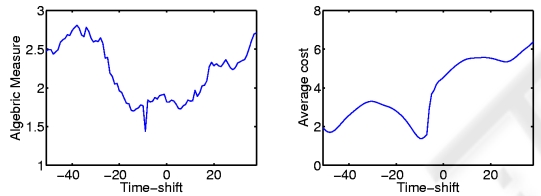
In the second experiment we consider outdoor basketball videos. We present two synchronization results. The first pair of sequences, illustrated in Fig. 9(a), shows two players seen from two cameras with almost opposite viewpoints. In addition to the challenging views, another difficulty of this experiment lies in the large time-shift equal to 76 frames between both considered videos. The second pair of sequences is presented in Fig. 10(a) where four players can be seen in both views. At some instance, some players move out of the field of views of cameras.

For both image sequences, we synchronize descriptors of the computed SSM-of. The time warping functions illustrated by the red curve in Fig. 9(c)

and Fig. 10(c) recover the ground truth transformations (blue curve).The synchronization of the first pair of image sequences demonstrates that the method can handle large time-shift, provided that motion in sequences is not periodic. In addition, we can observe that appearances and disappearances of the players in the second pair of videos do not disturb the time-warping estimation.

4.4 Comparison

In this subsection, we compare our method with the approach proposed by (Wolf and Zomet, 2006)(WZ). Due to the lack of space, we do not describe this method and invite the reader to refer to the paper for details. Their approach can be used to align sequences linked by time-shift transformation. For each possible time-shift value, they evaluate an algebraic measure based on rank constraints of trajectory-based matrices. They retain the time-shift that minimizes this measure. They propose to represent results by a graph of the computed measure versus the time-shift as illustrated in Fig. 11(a).

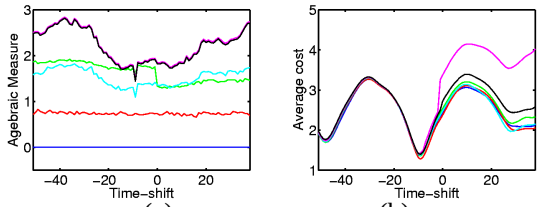


(a) (b)

Figure 11: Results on noise-free projected MoCAP point trajectories (a) WZ result : the algebraic error versus time-shift (b) Our result : the average cost versus time-shift.

In order to have similar result representation, we compute the average cost value in the cost matrix on a path for a given time-shift. We plot this average value versus the time-shift as illustrated in Fig. 11(b). Fig. 11 presents results for both methods on MoCAP dataset for the same example as in Fig. 4(e-h) but using 20 trajectories randomly chosen for each sequence. We re-compute SSMs for these trajectories.

In order to compare robustness of the two approaches, we apply noises with different variances. We can observe on Fig. 12 that for low variance noise (black, magenta and cyan curves) both methods recover the time-shift. However for higher variances, our method can recover the time-shift whereas their approach has difficulties (green, red and blue curves).



(a) (b)

Figure 12: Results for noisy data (a) WZ result : the algebraic error versus time-shift (b) Our result : the average cost versus time-shift.

5 CONCLUSIONS

We have presented a novel approach for video synchronization based on temporal self-similarities of videos. It is characterized by its simplicity and its flexibility: we do not impose restrictive assumptions as sufficient background information, or point correspondences between views. In addition, temporal self-similarities, which are not strictly view-invariant, supply view-independent descriptors for synchronization. Although our method does not provide synchronization with sub-frame accuracy, it can perform video synchronization automatically without temporal misalignment modeling.

We have validated our framework on datasets with controlled view settings and tested its performance on challenging real videos. These videos were captured by static cameras but the method could be applied to moving cameras, which we will investigate in future work. Furthermore, as the self-similarity matrix structures are not only stable under view changes but also specific to actions, the method could address the problem of action synchronization, i.e. the temporal alignment of sequences featuring the same action performed by different people under different viewpoints.

REFERENCES

Benabdelkader, C., Cutler, R. G., and Davis, L. S. (2004). Gait recognition using image self-similarity. *EURASIP J. Appl. Signal Process.*, 2004(1):572–585.

Carceroni, R., Padua, F., Santos, G., and Kutulakos, K. (2004). Linear sequence-to-sequence alignment. In *Proc. Conf. Comp. Vision Pattern Rec.*, pages I: 746–753.

Caspi, Y. and Irani, M. (2002). Spatio-temporal alignment of sequences. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 24(11):1409–1424.

Cha, S. and Srihari, S. (2002). On measuring the dis-

tance between histograms. *Pattern Recognition*, 35(6):1355–1370.

Cutler, R. and Davis, L. (2000). Robust real-time periodic motion detection, analysis, and applications. *PAMI*, 22(8):781–796.

Dalal, N. and Triggs, B. (2005). Histograms of oriented gradients for human detection. In *Proc. Conf. Comp. Vision Pattern Rec*, volume 2, pages 886–893.

Junejo, I., Dexter, E., Laptev, I., and Pérez, P. (2008). Cross-view action recognition from temporal self-similarities. In *Proc. Eur. Conf. Comp. Vision*, pages 293–306.

Lele, S. (1993). Euclidean distance matrix analysis (edma): Estimation of mean form and mean form difference. *Mathematical Geology*, 25(5):573–602.

Lucas, B. and Kanade, T. (1981). An iterative image registration technique with an application to stereo vision. In *Image Understanding Workshop*, pages 121–130.

Rabiner, L., Rosenberg, A., and Levinson, S. (1978). Considerations in dynamic time warping algorithms for discrete word recognition. *IEEE Trans. on Acoustics, Speech and Signal Processing*, 26(6):575– 582.

Rao, C.and Gritai, A., Shah, M., and Syeda Mahmood, T. F. (2003). View-invariant alignment and matching of video sequences. In *Proc. Int. Conf. on Image Processing*, pages 939–945.

Shechtman, E. and Irani, M. (2007). Matching local self-similarities across images and videos. In *Proc. Conf. Comp. Vision Pattern Rec.*

Stein, G. (1999). Tracking from multiple view points: Self-calibration of space and time. In *Proc. Conf. Comp. Vision Pattern Rec.*, volume 1, pages 521–527.

Tuytelaars, T. and Van Gool, L. (2004). Synchronizing video sequences. In *Proc. Conf. Comp. Vision Pattern Rec.*, volume 1, pages 762–768.

Ukrainitz, Y. and Irani, M. (2006). Aligning sequences and actions by minimizing space-time correlations. In *Proc. Europ. Conf. on Computer Vision*.

Ushizaki, M., Okatani, T., and Deguchi, K. (2006). Video synchronization based on co-occurrence of appearance changes in video sequences. In *Int. Conf. on Pattern Recognition*, pages III: 71–74.

Weinland, D., Boyer, E., and Ronfard, R. (2007). Action recognition from arbitrary views using 3d exemplars. In *Proc. Int.Conf. on Computer Vision*, pages 1–7.

Wolf, L. and Zomet, A. (2006). Wide baseline matching between unsynchronized video sequences. *Int. J. of Computer Vision*, 68(1):43–52.