ActionSnapping: Motion-Based Video Synchronization

Jean-Charles Bazin(✉) and Alexander Sorkine-Hornung

Disney Research, Zurich, Switzerland
{jean-charles.bazin,alexander}@disneyresearch.com

Abstract. Video synchronization is a fundamental step for many applications in computer vision, ranging from video morphing to motion analysis. We present a novel method for synchronizing action videos where a similar action is performed by different people at different times and different locations with different local speed changes, e.g., as in sports like weightlifting, baseball pitch, or dance. Our approach extends the popular "snapping" tool of video editing software and allows users to automatically snap action videos together in a timeline based on their content. Since the action can take place at different locations, existing appearance-based methods are not appropriate. Our approach leverages motion information, and computes a nonlinear synchronization of the input videos to establish frame-to-frame temporal correspondences. We demonstrate our approach can be applied for video synchronization, video annotation, and action snapshots. Our approach has been successfully evaluated with ground truth data and a user study.

1 Introduction

Video synchronization aims to temporally align a set of input videos. It is at the core of a wide range of applications such as 3D reconstruction from multiple cameras [20], video morphing [27], facial performance manipulation [6,10], and spatial compositing [44]. When several cameras are simultaneously used to acquire multiple viewpoint shots of a scene, synchronization can be trivially achieved using timecode information or camera triggers. However this approach is usually only available in professional settings. Alternatively, videos can be synchronized by computing a (fixed) time offset from the recorded audio signals [20]. The videos can also be synchronized by manual alignment, for example by finding video frame correspondences and computing the required time offset. These techniques can be extended to also finding a (fixed) speed factor (linear synchronization), for example when using cameras recording at different frame rates. However, the required manual alignment is usually tedious and time consuming: using video editing software, the user needs to manually drag the videos in the timelines in such a way that the frame correspondences are temporally aligned. The popular "snapping" tool can help the user align videos at pre-specified markers (e.g., frame correspondences) but these markers need to be provided manually and the synchronization is limited to a global time offset.

B. Leibe et al. (Eds.): ECCV 2016, Part V, LNCS 9909, pp. 155–169, 2016.
DOI: 10.1007/978-3-319-46454-1_10

Some techniques have been proposed to automatically compute the synchronization, for example using appearance information such as SIFT features [44]. These methods are appropriate to synchronize videos showing the same scene.

In contrast, our goal is to synchronize videos of a similar action performed by different people at different times and locations (see Fig. 1), i.e., with different appearances. Examples of application include video manipulation [6], sport video analysis (e.g., to compare athletes' performances), video morphing [27] and action recognition. Our approach enhances the "snapping" tool of video editing software and allows the user to automatically snap action videos together in a timeline based on their content. Some methods are dedicated to particular setups (e.g., facial performance [6]) or using other data (e.g., skeleton data from a Kinect [44]). Instead, our approach can be applied to general actions and does not rely on appearance information but rather leverages motion information. We obtain motion information from the input videos by taking advantage of the recent work on dense trajectory extraction [43]. Given a set of input videos, our approach computes a nonlinear synchronization path in a frame-to-frame motion similarity matrix that can handle local speed variations of the recorded actions. This synchronization path can then be used to create a synchronized version of the input videos where the action occurs at the same time in both videos.

Our contributions are the following. First, we propose a novel algorithm that allows the nonlinear synchronization of videos of a similar action performed by different people at different times and different locations. Our approach uses point trajectories [43], does not require challenging silhouette or skeleton extraction, and runs in a fully automatic manner. Second, we develop a multi-temporal scale method to deal with videos of large speed differences. Third, we show the applications of our method for different tasks, including video synchronization, video annotation and action snapshots. Finally, we demonstrate the validity of our approach with qualitative and quantitative evaluation.

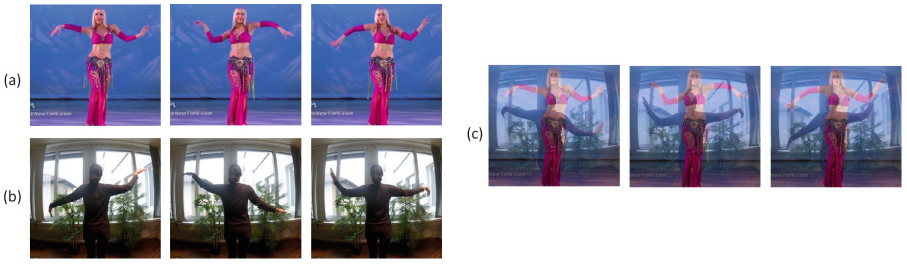

Fig. 1. Given a set of input videos of a similar action (here, snake arms dance in (a) and (b)) performed by different people at different times and different locations, our approach automatically computes a nonlinear synchronization of these videos from motion cues. Our synchronization results are shown overlayed in (c). While the appearance of the scenes looks very different, our approach successfully manages to synchronize the snake arms in (c).

2 Related Work

Our work is related to synchronization, temporal ordering and temporal correspondences. Below we discuss the most related methods in these areas.

Simultaneous acquisition. When a scene is simultaneously captured by several cameras, camera triggers can be used to start the acquisition of all the cameras at the same time, and therefore the captured videos are directly synchronized. In the absence of camera triggers, the cameras might start the acquisition at different times. In such case, typical synchronization solutions are limited to a fixed temporal offset, or a fixed speed scalar for cameras recording at different frame rates. If the acquisition time of the cameras is stored with synchronized timecode, then the time offset can be easily obtained by the time difference of the timecodes. However, such hardware is usually only available in professional settings. An alternative for casual use cases is to use the recorded audio and compute the time offset that temporally aligns the audio signals, for example for 3D reconstruction from multiple cameras [4,20].

In our case, actions are performed at different time instances, e.g., by different people and/or at different places. Therefore these videos cannot be captured simultaneously. As a consequence, camera triggers, timecode information and audio signals cannot be used for synchronization. Moreover, such actions cannot be related by a fixed temporal linear relation, i.e., fixed temporal offset and fixed global speed factor, as the local speed might vary along the action. To cope with the local speed change variation, we need to perform a nonlinear synchronization.

Video-based synchronization. Some techniques exist for synchronizing video sequences acquired at different time instances [9,11,13,33,44]. However, these methods estimate temporal correspondences from appearance-based descriptors such as SIFT. Therefore they work best with camera ego-motion and large-scale scene changes. While these assumptions are reasonable for videos recorded at the same location, they cannot cope with actions performed at different locations with different appearances and/or cannot capture the subtle change of motion since the global appearance might be the same. Note also that some of these methods, such as [9], search for a temporal linear relation which cannot deal with local speed variations.

To represent the status of an action in each video frame, silhouette information is one of the possible options, as for example used by Zhou and De la Torre [49,50] for human action synchronization, and by Xu et al. [46] for animating animal motion from still images. However, in practice, extracting the silhouette in an automatic way is a challenging task. For example it could be performed by background subtraction [49,50], but this requires a foreground-background model and/or a clean plate in a controlled environment. An alternative is manual segmentation [46], for example with GrabCut [32], which can be time consuming. Instead we aim for an automatic approach.

Another technique to represent the status of an action is to extract the human body pose and skeleton, for example from a single image [42,48], video [15,40] or

depth data [18,37]. However despite the recent advances, retrieving an accurate skeleton from a monocular video is still ongoing research due to the difficulty and ambiguity of the task as well as the large variation of human poses. Moreover, while using skeleton information would be appropriate for human actions, we aim for a general method without any priors on shapes, poses and actions, i.e., a method that can be applied on actions of humans and non-humans for example.

Shechtman et al. [35] present a descriptor which captures internal geometric layouts of local self-similarities within images. By applying this technique in space-time, they can detect segments of similar actions in videos. In contrast, our goal is to compute a dense frame-to-frame temporal alignment. In the context of video morphing, Liao et al. [28] establish temporal correspondences between videos using manually given point correspondences. In contrast, we aim for a fully automatic method.

Some existing methods are dedicated to the synchronization of facial performance videos. They are based on facial landmarks [6,10] or the expression coefficients of a morphable face model [47]. While these methods have demonstrated impressive results, they are specifically designed for facial performances.

Additional modalities. Additional modalities can also be used to facilitate the synchronization of actions. For example, Hsu et al. [21] use human motion capture data acquired in a motion capture lab. Zhou and De la Torre [49,50] can align different subjects with different sensors such as video, motion capture and accelerometers. In contrast, our only input data is a set of monocular videos.

Image sequence ordering. Synchronization can also be seen as temporal ordering of a collection of images. Given a set of pictures of a dynamic scene acquired roughly from the same viewpoint and location, Basha et al. [5] compute a temporal order of these pictures using point correspondences. In contrast, we aim for synchronizing videos of actions performed by different people and at different locations.

Images or video frames can also be re-ordered to create new visual contents. For example Bregler et al. [8] re-order mouth sequences from training data to generate a new video sequence according to a given audio track. Kemelmacher-Shlizerman et al. [23] puppeteer a person by finding relevant images from a large image database of the same person and spatially aligning them. Garrido et al. [17] generalize this technique for face reenactment from video sources. Averbuch-Elor et al. [3] compute a spatial ordering (i.e., relative position) of photos of a temporal event. They assume a nearly instantaneous event, such as a particular time instance of a performance. In contrast to these applications, our goal is to synchronize videos, that is establishing dense frame-to-frame temporal correspondences between the input videos.

Action recognition. Since we are working on videos of actions, our work is in part related to the literature of action recognition. Reviewing works of this active research area is beyond the scope of this paper, and we refer the readers to survey papers [1,31]. In contrast to methods for action recognition, our goal is not to

explicitly recognize or classify actions. We assume our input videos represent a similar but arbitrary action and we aim to temporally align these videos. The existing methods for action recognition could be applied in pre-processing in order to automatically identify videos of the same action among a large video collection that can then be processed by our method.

3 Overview of Our Approach

Our goal is to synchronize action videos, that is establishing temporal correspondences between the frames of the action videos. Our synchronization algorithm consists of the following three main steps. First, for each input video, we extract motion information of the action by taking advantage of the recent work on dense trajectory extraction [43]. The motion is obtained in the form of point trajectories by tracking and we then represent these trajectories by multi-temporal scale motion descriptors. Secondly, the difference of motions between each pair of frames of the input videos is computed and stored in a cost matrix. Finally, we obtain a nonlinear synchronization path as the lowest-cost path in this cost matrix. Given two input videos, the synchronization path indicates which frame of a video corresponds to which frame of the other video.

Many of the references cited above also compute synchronization as a low cost path [6,44], or by related techniques like dynamic time warping [10,49,50]. However, as discussed, their features (e.g., silhouette, appearance and facial coefficients) are not applicable for our general video settings (background, pose, humans, animals) with different appearances [44] (e.g., see background of Fig. 1) and might require manual segmentation [46]. In contrast, we use general motion features based on point trajectories and apply a multi-temporal scale approach, which allow us to automatically and robustly synchronize general videos of similar action performed by different people at different locations, even with different appearances and speeds.

4 Proposed Approach

Our input is a set of videos showing a similar action. One of these videos is considered the reference video, and the goal is to synchronize the other videos to this reference video. Without lack of generality, we consider two input videos. In the case of multiple input videos, each video is independently synchronized to the reference video.

Let $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$ be two input videos, and $\boldsymbol{v}_i(j)$ be the j-th frame of video $\boldsymbol{v}_i$. Formally, the synchronization is defined as a mapping $\boldsymbol{p} : \mathbb{R} \rightarrow \mathbb{R}^2$, where $\boldsymbol{p}(t) = (p_1(t), p_2(t))$ associates a global time t with two corresponding video frames $\boldsymbol{v}_1(p_1(t))$ and $\boldsymbol{v}_2(p_2(t))$. As discussed, a linear time mapping is not applicable for our videos due to the local speed change variations. Instead we are searching for a nonlinear temporal mapping that we compute as the low cost path in a cost matrix using Dijkstra's algorithm [12]. In the following, we describe the details of our algorithm.

4.1 Features

Point trajectories. Several techniques to extract and represent motion information from videos have been proposed, especially for camera motion estimation and action recognition. Some of the most popular options include KLT [36], optical flow [7,22], STIP [25] and SIFT flow [30], among many others. We opted for the dense point trajectory extraction technique of Wang et al. [43] since they demonstrated superior results for action recognition. Other point trajectories like [34] could also be used.

Wang et al. sample feature points on a grid spaced by 5 pixels, and each point is tracked to the next frame by median filtering on a dense optical flow field [14]. The camera motion is computed by homography and RANSAC [19], and canceled out from the optical flow. The point tracks consistent with the homography are considered as due to the camera motion and thus removed. We compensate the positions of the remaining tracks by the camera motion, such that these tracks correspond to the actual motion of the action even for videos acquired by hand-held cameras. To avoid tracking drifting, we track points only over $L = 30$ frames. Given a starting frame at time t, the point $P_t = (x_t, y_t)$ is tracked over the next L frames, and the resulting trajectory is composed of the points $(P_t, P_{t+1}, \ldots, P_L)$. Representative examples of trajectories are shown in Fig. 2.

Fig. 2. Representative examples of point trajectories for our video synchronization approach. For a better visualization, only a subset of the trajectories is displayed.

Trajectory representation. To compare the trajectories, we need an appropriate representation. Given a trajectory, a simple concatenation of the points positions would be sensitive to the location of the action in the image. To be location invariant, we instead use the displacement vectors, i.e., the change of x and y coordinates. We obtain a trajectory representation $S = (\Delta P_t, \ldots, \Delta P_{t+L-1})$ where $\Delta P_t = (x_{t+1} - x_t, y_{t+1} - y_t)$. Finally we normalize the trajectory representation vector by its norm. This normalization permits to handle videos where the action is performed at different distances from the camera and videos with different amounts of zoom.

This approach provides satisfying results when the speed of the actions is not too different, typically up to 1.5 times. Beyond this speed ratio, the descriptors of a same trajectory motion but executed at different speeds would be too different to still have a meaningful correlation. To deal with videos where action is performed at different speeds (in our case up to 10 times), we use a multiple

temporal scale approach. For efficiency reason, we do not re-track trajectories over different temporal windows. Instead, we use the trajectories already tracked and compute their multi-temporal scale representation. Concretely, given a trajectory tracked over L frames, we consider the point at mid-time, and compute the trajectory descriptors S over different temporal windows $W = (3, \ldots, 30)$ centered at that mid-time. We write the multi-temporal descriptors of the trajectory as $(S_3, \ldots, S_{30})$.

To handle speed ratios above 10 times, two options could be envisaged. First, a shorter temporal window could be used but the information might get unreliable. Second, points could be tracked over more frames. In practice, capping the speed ratio to 10 worked correctly for all the videos we tested.

4.2 Synchronization

Frame motion representation. From the previous steps, we have the set of trajectories of the input videos and their descriptors. We now would like to represent the motion present in each frame of the input videos from these trajectories. This motion representation will then be used to compare the motion between two different frames of the input videos. First of all, we consider a trajectory is "T-visible" in a frame $\boldsymbol{v}_i(t)$ if at least a part of this trajectory is continuously tracked between $t - T/2$ and $t + T/2$ in $\boldsymbol{v}_i$. To represent the motion of each video frame, we apply a bag-of-features approach [26]. For this, we first compute a codebook of all the trajectory descriptors of the input videos. To compute the vocabulary words, we apply k-means, with $k = 100$ by default, and for efficiency reason, we run it on a subset of $50,000$ randomly selected trajectories. Then given a frame $\boldsymbol{v}_i(t)$, we collect all the trajectories that are T-visible in that frame. Each of these trajectories is assigned to its closest vocabulary word. By doing so for all the trajectories of $\boldsymbol{v}_i(t)$, we obtain a histogram of word occurrences that we use as a motion descriptor of the frame $\boldsymbol{v}_i(t)$. We note $h_i^T(t)$ the resulting histogram of frame $\boldsymbol{v}_i(t)$ over a temporal window T. By applying this procedure for each temporal window $T \in W$, we obtain a multi-temporal descriptor of the motion of the frame $\boldsymbol{v}_i(t)$.

Motion comparison. We now use the above motion representation to compare the motion between two frames. First, since different frames and different videos might have a different number of trajectories, the number of entries in the histogram bins might vary, and therefore it is needed to normalize the histograms. For normalization, we apply RootSIFT [2].

We compare the motions of two frames of the input video pair by measuring the distance $d()$ between their histograms using χ^2 distance [41]. For the multi-temporal histograms, we conduct a temporal scale selection. Concretely, given the multi-temporal histogram of frames $\boldsymbol{v}_1(t)$ and $\boldsymbol{v}_2(t')$, respectively the multiple $h_1^T(t)$ and $h_2^T(t')$ with each $T \in W$, we select the temporal scale pair (T, T') leading to the lowest histogram distance:

$$c(t, t') = \min_{(T,T') \in W \times W} d(h_1^T(t), h_2^{T'}(t')). \tag{1}$$

The cost $c(t, t')$ represents the cost of the motion difference between the frames $\boldsymbol{v}_1(t)$ and $\boldsymbol{v}_2(t')$, taking into account potential speed differences.

Cost matrix and synchronization path. We compute the costs $c(t, t')$ for each pair of frames $\boldsymbol{v}_1(t)$ and $\boldsymbol{v}_2(t')$ and store them in a 2D cost matrix C of size $N_1 \times N_2$ where N_i refers to the number of frames of video $\boldsymbol{v}_i$. Representative results of cost matrices computed by our approach are shown in Fig. 3.

Finally, given the cost matrix C, we compute the synchronization path as the lowest cost path in C using Dijkstra's algorithm [12], see the white paths in Fig. 3. This synchronization path $\boldsymbol{p}$ is a nonlinear mapping that establishes frame-to-frame temporal correspondences between the two videos $\boldsymbol{v}_1$ and $\boldsymbol{v}_2$. Therefore, by navigating along the path, we can create synchronized versions of the input videos.

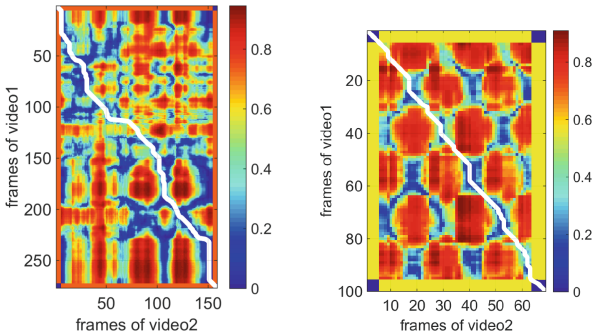

Fig. 3. Representative examples of the computed cost matrix and the low cost nonlinear synchronization path shown in white. The matrix on the left corresponds to a high kick, and the matrix on the right to repetitive bench press motions.

5 Experiments

5.1 Implementation Details

We processed the videos on a desktop computer equipped with an Intel i7 3.2 GHz and 16 GB RAM. The main program is implemented in Matlab and uses some C++ modules for optical flow and point tracking. Given the trajectories [43] obtained in preprocessing, the total execution time (motion descriptors, multi-temporal scale representation, scale selection and path computation) takes around 2–5 min for a pair of typical 20 s full HD videos, depending on the number of trajectories. The entire synchronization pipeline runs in a fully automatic manner, and the synchronized versions of the videos are also generated automatically.

5.2 Results

In the following we show some representative results obtained by our approach. We kindly invite the readers to refer to our project webpage[1] for the full video results, comparisons and additional results. We apply our approach on various videos, for example from the publicly available UCF101 dataset [38] as well as videos that we captured ourselves. A nonlinear synchronization path is computed from the motion cost matrix. This synchronization path is then used to establish temporal correspondences of video frames. Figure 4 shows some examples of synchronized frames obtained from the cost matrix and synchronization path shown in Fig. 3-right.

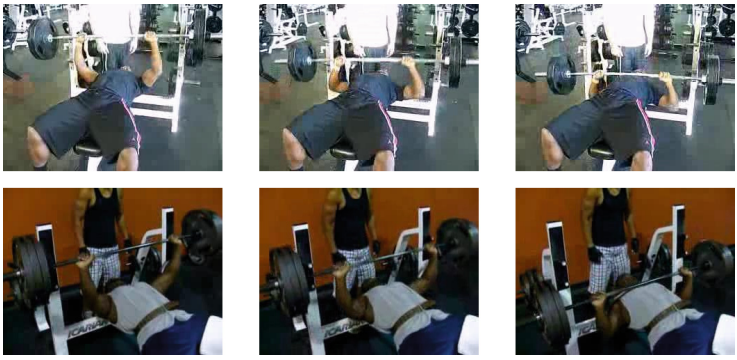

Fig. 4. Examples of frame synchronization obtained by our approach from the cost matrix and the nonlinear synchronization path of Fig. 3-right. One can observe that the barbell is at the same position in the synchronized frames, which indicates that the synchronization result is correct.

Our approach can be applied on multiple input videos. One of these videos is considered the reference video, and the other videos are then synchronized with respect to this reference video. Figure 5 shows a representative synchronization result of multiple input videos of the weightlifting exercise called *clean and jerk*. Our method can be seamlessly applied to scenes of similar (rows 1 and 2) and different appearances (rows 1, 3 and 4). For example, one can note how the backgrounds, athletes morphologies and clothes look different. Moreover the videos at rows 3 and 4 are acquired by a hand-held camera and thus contain camera motion. The videos at rows 1 and 2 contain tilt and zoom changes. The video at row 4 contains overlayed text (see "140k" in the second column). Despite these challenges, our approach successfully manages to synchronize these videos.

In addition to synchronization, we show that our method can be applied for video labelling. Here we manually annotated the steps of the clean and jerk in the reference video (see text in gray boxes at row 1 of Fig. 5). Thanks to our

[1] http://www.disneyresearch.com/publication/ActionSnapping.

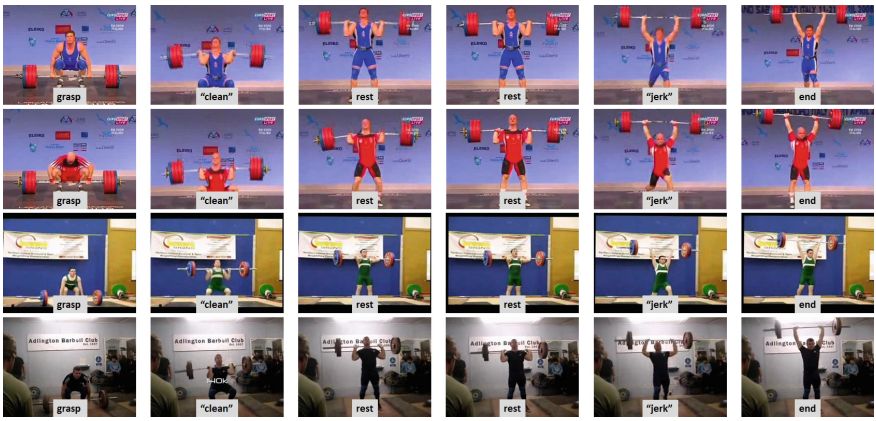

Fig. 5. Automatic synchronization of multiple input videos showing the weightlifting exercise called *clean and jerk*. Our synchronization result can then be used for automatic label propagation for video annotation (see text in gray boxes).

synchronization path, we know which frame of the reference video corresponds to which frame of each other video. Therefore the labels can be automatically propagated from the reference video to the other videos (see gray boxes at rows 2, 3 and 4). Labelling of every single frame of the videos could be achieved manually, but it would be tedious and time consuming, especially for a large set of videos. An alternative strategy could be to explicitly detect the different steps of an action in the context of action recognition [1,31], for example using machine learning techniques and a large training dataset composed of numerous videos with manual annotations. In contrast, our approach only needs one labelled video and then automatically propagates the labels to the other videos.

5.3 Action Snapshots

Our video synchronization approach allows the creation of action snapshots. In contrast to methods using a single video [24,39], we can generate action snapshots from different videos of different people by simply sampling the frames of the synchronized videos. It is also possible to choose the number of frames of each input video. A result from 7 input videos using 1 frame per video is shown in Fig. 6.

Fig. 6. Action snapshots of a baseball pitch from 7 input videos of different people.

5.4 Interaction

Our method aims to synchronize the motion visible in the video. If several motions are visible, then we allow the user to specify which motions or which part of the video he/she wants to synchronize in an interactive and intuitive manner, by simply painting over the action of interest. For example, in Fig. 1, the professional dancer had a hip motion that the beginner dancer did not replicate. To indicate that the arms motions should be used for synchronization and not the hip motion, the user can simply draw one rough rectangle over the location of the arms.

5.5 Evaluation

Comparison to ground truth. To evaluate the accuracy of our synchronization path, we create pairs of videos with ground truth speed difference. Given a video, we create its sped up version. To avoid introducing in-between frame interpolation artifacts, we consider high frame rate videos and create sped up versions by simply skipping frames. If the speed is k times faster, then the ground truth synchronization path should be a line of slope k. We do not constrain the computation of our synchronization path to be a straight line. We compare the ground truth path and our computed path as follows: for each point on our path, we measure the distance to the nearest point on the ground truth path. We apply this procedure on 5 different high frame rate videos acquired by a GoPro camera. For each of them, we generate the video versions with a speed up of each $k \in (1, \ldots, 10)$, and compute the path distance. A comparison of the computed path and the ground truth path for one video pair is shown in Fig. 7. The average error is slightly below 3 frames and the maximum error is 15 frames. While this value might sound relatively high, it has to be noted that the computed path has a "feasibility range". For example, in the extreme case where the input videos have no motion at all, then any synchronization path would lead to visually perfect results. To study this further, we conducted a user study.

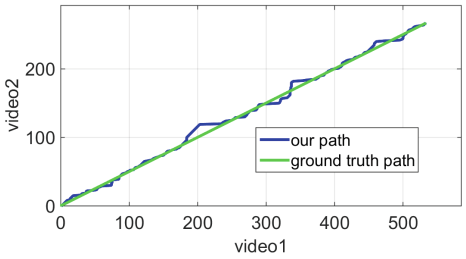

Fig. 7. Evaluation: comparison of our synchronization path to the ground truth path.

User study. We conducted a user study on the results obtained by our synchronization approach on pairs of videos from the UCF101 dataset [38]. We did not consider videos with camera cuts and extreme view point differences (e.g., front

vs. side views). Examples of categories include baseball pitch, bench press, and clean and jerk. Instead of conducting the user study on the videos themselves, we conducted it on a set of frames uniformly sampled from each video pair. The key advantage is that it provides a fine measurement at the frame level. 12 people participated and we tested 15 pairs of videos with 10 uniformly sampled frames per video pair, which represents a total of 1800 votes. The participants were asked to respond to the statement "This pair of images corresponds to the same time instance along the course of the action" by choosing a response from a five-point Likert scale: strongly agree (5), agree (4), neither agree not disagree (3), disagree (2) or strongly disagree (1).

The distribution of the scores is available in Fig. 8. The participants strongly agreed in 66.1 % of the experiments, and a total of 87.2 % had a score equal to or higher than 4 (agree). This demonstrates our approach can achieve an accurate frame-to-frame temporal correspondence.

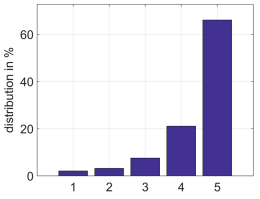

Fig. 8. Distribution of the grades of the user study.

Examples of image pairs with a low score are shown in Fig. 9. In Fig. 9(a), the temporal misalignment is small (around 5 frames, i.e., 0.2 s) but still visible because the baseball pitch of the professional athlete on the right was particularly fast and the video is recorded at 25 fps. A participant graded Fig. 9(b) with 1 because "the athlete on the left is pulling up the bar, while the athlete on the right seems to be at rest". Our method did not manage to synchronize Fig. 9(c) because the motions are horizontally symmetric: the left player threw the ball to the left, while the right player threw the ball to the right. We tested this hypothesis by mirroring one of the videos, and obtained a successful synchronization.

(a) (b) (c)

Fig. 9. Examples of three image pairs with a low score from the user study.

5.6 Limitations

Our approach computes a continuous synchronization path. Therefore it cannot deal with actions performed in a different order, for example walking then jumping in a video, and jumping then walking in another video. This use case would require a method that allows the computation of a non-continuous path, for example reshuffling of video segments.

While our approach has been shown to be effective, it still depends on the availability of reliable motion information. Difficult cases where tracking is unreliable include fast motion, highly blurred images, and uniform textures.

In the case of hand-held cameras, the current implementation compensates the camera motion from the extracted flow information with a homography model, which assumes the background scene is rather planar or relatively far away. For general scenes, more sophisticated camera motion estimation techniques should be employed [45].

In some particular cases, one of the videos might need to be paused for a long duration. For example, in some clean and jerk sequences (see supplementary material), the athlete of the reference video is at rest for a relatively long time before the final push up, while the athlete of the second video completes the full motion quickly. Therefore this second video needs to be paused over the rest duration. While this is not an issue for many applications (such as motion analysis, morphing and video annotation), this pause artifact needs to be avoided for generating realistic retimed videos. An interesting research direction is how to "pause" a video without a freezing effect, for example by adding artificial movement [16] or with optimized frame looping [29].

6 Conclusion

We have presented a novel approach for synchronizing videos of a similar action performed by different people at different times and different locations. Our approach computes a nonlinear synchronization path between the input videos using motion information. Our multi-temporal scale technique allows to handle videos with large speed differences. Our method is general: since it does not assume any priors on motions, shapes or poses, it can be applied to a wide range of videos with cluttered backgrounds, various actions as well as continuous and repetitive motions. Qualitative and quantitative evaluations demonstrated the validity of our approach.

Our approach runs in a fully automatic manner and obtains satisfying results on different kinds of videos. Therefore we believe it will facilitate and enable several applications that can benefit from video synchronization but currently require tedious manual synchronization, such as video morphing, video analysis and video annotation.

Acknowledgements. We are very grateful to World Dance New York for giving us the permission to use their YouTube videos.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human activity analysis: a review. ACM Comput. Surv. **43**, 16 (2011)
2. Arandjelovic, R., Zisserman, A.: Three things everyone should know to improve object retrieval. In: CVPR (2012)
3. Averbuch-Elor, H., Cohen-Or, D.: RingIt: ring-ordering casual photos of a temporal event. TOG **34**, 33 (2015)

4. Ballan, L., Brostow, G.J., Puwein, J., Pollefeys, M.: Unstructured video-based rendering: interactive exploration of casually captured videos. TOG (SIGGRAPH) **29**, 87 (2010)
5. Basha, T.D., Moses, Y., Avidan, S.: Photo sequencing. IJCV **110**(3), 275–289 (2014)
6. Bazin, J.C., Malleson, C., Wang, O., Bradley, D., Beeler, T., Hilton, A., Sorkine-Hornung, A.: FaceDirector: continuous control of facial performance in video. In: ICCV (2015)
7. Beauchemin, S.S., Barron, J.L.: The computation of optical flow. ACM Comput. Surv. **27**, 433–466 (1995)
8. Bregler, C., Covell, M., Slaney, M.: Video rewrite: driving visual speech with audio. In: SIGGRAPH (1997)
9. Caspi, Y., Irani, M.: Spatio-temporal alignment of sequences. TPAMI **24**, 1409–1424 (2002)
10. Dale, K., Sunkavalli, K., Johnson, M.K., Vlasic, D., Matusik, W., Pfister, H.: Video face replacement. TOG (SIGGRAPH Asia) **30**(6) (2011)
11. Diego, F., Serrat, J., López, A.M.: Joint spatio-temporal alignment of sequences. Trans. Multimedia **15**, 1377–1387 (2013)
12. Dijkstra, E.W.: A note on two problems in connexion with graphs. Numer. Math. **1**, 269–271 (1959)
13. Evangelidis, G.D., Bauckhage, C.: Efficient subframe video alignment using short descriptors. TPAMI **35**, 2371–2386 (2013)
14. Farnebäck, G.: Two-frame motion estimation based on polynomial expansion. In: Scandinavian Conference on Image Analysis (2003)
15. Fossati, A., Dimitrijevic, M., Lepetit, V., Fua, P.: From canonical poses to 3D motion capture using a single camera. TPAMI **32**, 1165–1181 (2010)
16. Freeman, W.T., Adelson, E.H., Heeger, D.J.: Motion without movement. In: SIGGRAPH (1991)
17. Garrido, P., Valgaerts, L., Rehmsen, O., Thormaehlen, T., Perez, P., Theobalt, C.: Automatic face reenactment. In: CVPR (2014)
18. Girshick, R.B., Shotton, J., Kohli, P., Criminisi, A., Fitzgibbon, A.W.: Efficient regression of general-activity human poses from depth images. In: ICCV (2011)
19. Hartley, R.I., Zisserman, A.: Multiple View Geometry in Computer Vision. Cambridge University Press, Cambridge (2004)
20. Hasler, N., Rosenhahn, B., Thormählen, T., Wand, M., Gall, J., Seidel, H.: Markerless motion capture with unsynchronized moving cameras. In: CVPR (2009)
21. Hsu, E., Pulli, K., Popovic, J.: Style translation for human motion. TOG (SIGGRAPH) **24**, 1082–1089 (2005)
22. Jain, M., Jegou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR (2013)
23. Kemelmacher-Shlizerman, I., Sankar, A., Shechtman, E., Seitz, S.M.: Being John Malkovich. In: ECCV (2010)
24. Klose, F., Wang, O., Bazin, J.C., Magnor, M.A., Sorkine-Hornung, A.: Sampling based scene-space video processing. TOG (SIGGRAPH) **34**, 67 (2015)
25. Laptev, I.: On space-time interest points. IJCV **64**, 107–123 (2005)
26. Li, F., Perona, P.: A bayesian hierarchical model for learning natural scene categories. In: CVPR (2005)
27. Liao, J., Lima, R.S., Nehab, D., Hoppe, H., Sander, P.V.: Semi-automated video morphing. In: CGF (Eurographics Symposium on Rendering) (2014)
28. Liao, J., Lima, R.S., Nehab, D., Hoppe, H., Sander, P.V., Yu, J.: Automating image morphing using structural similarity on a halfway domain. TOG **33**, 168 (2014)

29. Liao, Z., Joshi, N., Hoppe, H.: Automated video looping with progressive dynamism. TOG (SIGGRAPH) **32**, 4 (2013)
30. Liu, C., Yuen, J., Torralba, A.: SIFT flow: dense correspondence across scenes and its applications. TPAMI (2011)
31. Poppe, R.: A survey on vision-based human action recognition. Image Vis. Comput. **28**, 976–990 (2010)
32. Rother, C., Kolmogorov, V., Blake, A.: "GrabCut": interactive foreground extraction using iterated graph cuts. TOG (SIGGRAPH) **23**, 309–314 (2004)
33. Sand, P., Teller, S.J.: Video matching. TOG (SIGGRAPH) **23**(3), 592–599 (2004)
34. Sand, P., Teller, S.J.: Particle video: Long-range motion estimation using point trajectories. IJCV **80**, 72–91 (2008)
35. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: CVPR (2007)
36. Shi, J., Tomasi, C.: Good features to track. In: CVPR (1994)
37. Shotton, J., Fitzgibbon, A.W., Cook, M., Sharp, T., Finocchio, M., Moore, R., Kipman, A., Blake, A.: Real-time human pose recognition in parts from single depth images. In: CVPR (2011)
38. Soomro, K., Zamir, A.R., Shah, M.: UCF101: A dataset of 101 human actions classes from videos in the wild. Technical Report CRCV-TR-12-01 (2012)
39. Sunkavalli, K., Joshi, N., Kang, S.B., Cohen, M.F., Pfister, H.: Video snapshots: creating high-quality images from video clips. TVCG **18**, 1868–1879 (2012)
40. Urtasun, R., Fleet, D.J., Fua, P.: Temporal motion models for monocular and multiview 3D human body tracking. CVIU **104**, 157–177 (2006)
41. Vedaldi, A., Zisserman, A.: Efficient additive kernels via explicit feature maps. TPAMI **34**, 480–492 (2012)
42. Wang, C., Wang, Y., Lin, Z., Yuille, A.L., Gao, W.: Robust estimation of 3D human poses from a single image. In: CVPR (2014)
43. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
44. Wang, O., Schroers, C., Zimmer, H., Gross, M., Sorkine-Hornung, A.: VideoSnapping: interactive synchronization of multiple videos. TOG (SIGGRAPH) **33**, 77 (2014)
45. Wu, C.: Towards linear-time incremental structure from motion. In: International Conference on 3D Vision (3DV) (2013)
46. Xu, X., Wan, L., Liu, X., Wong, T., Wang, L., Leung, C.: Animating animal motion from still. TOG (SIGGRAPH Asia) **27**, 117 (2008)
47. Yang, F., Bourdev, L.D., Shechtman, E., Wang, J., Metaxas, D.N.: Facial expression editing in video using a temporally-smooth factorization. In: CVPR (2012)
48. Yang, Y., Ramanan, D.: Articulated human detection with flexible mixtures of parts. TPAMI **35**, 2878–2890 (2013)
49. Zhou, F., De la Torre, F.: Canonical time warping for alignment of human behavior. In: NIPS (2009)
50. Zhou, F., De la Torre, F.: Generalized time warping for multi-modal alignment of human motion. In: CVPR (2012)