Action-based multi-camera synchronization

Luca Zini, Andrea Cavallaro, Francesca Odone

***Abstract*—We propose a video alignment method based on observing the actions of a set of articulated objects. Given object association information, the proposed video synchronization method is applicable to general and unconstrained scenarios in a way that is not feasible with current state-of-the-art approaches: the proposed method does not impose constraints on the relative pose or motion of the cameras, on the structure of the time warping between the videos and on the amount of overlap among the fields of view. The proposed method uses a high-level video analysis (object actions) and models the alignment as a frame association problem (as opposed to the traditional continuous time warping). We present a qualitative and quantitative analysis of the results in real-world complex scenarios, showing the robustness of the method and higher accuracy compared to the only approach from the literature that works under similar conditions.**

***Index Terms*—Multi-view synchronization; object matching; motion description; video alignment.**

I. Introduction

Given the widespread availability of amateur video cameras in different types of handheld devices, the synchronization of different recorded videos is important when the same dynamic event is captured by different devices from different viewpoints. The objective of video synchronization is to estimate the correspondence between frames of different videos using geometrical or dynamical constraints.

The relative pose type among cameras can be classified into one of three groups: paired, frontal or generic (Fig. 1). Cameras are *paired* when their relative rotation is negligible [1], [2], [3] and they differ for translation or focal length [4], [5]. Cameras are *frontal* when they are framing each other and their relative rotation is approximatively 180 degrees [4], [1]. Cameras have a *generic* relative pose when they have non-negligible relative rotations and translations that reduce the common Field of View (FOV) or pose under which they capture objects [6], [7].

Existing methods for video alignment are generally based on strong assumptions on their relative camera location and pose (i.e. the geometry), on the structure of the time misalignment [8], [3] or on the combination of audio and visual features [9]. Assumptions include the explicit knowledge of the scene geometry [6], [10] or that geometry can be retrieved accurately by feature matching [4], [5]. The pose is an important variable both in the estimation of the geometry and in the alignment, and restrictions on the configurations allow one to use specific algorithms and heuristics (e.g. [4]). State-of-the-art methods lack the capability of aligning a set of videos without implicit or explicit requirements on the geometry and on the time misalignment. Our objective is therefore to extend video analysis to data acquired by different cameras in arbitrary poses (generic configuration) and with non-linear and non-smooth time-warping functions.

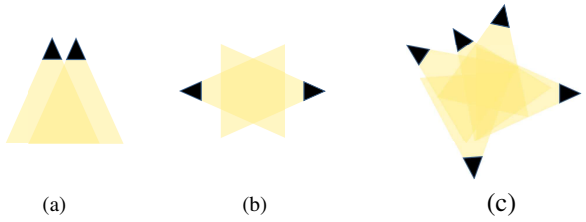


Fig. 1. Relative camera poses: (a) paired, (b) frontal, (c) generic. Implicit or explicit assumptions in state-of-the-art algorithms limit the use of methods to a subset of configurations only. The proposed approach works instead with generic camera configurations.

In this paper we present a method to compute a frame-level video alignment with the assumption that cameras view articulated objects and that objects association information is given as input. From the observation of objects actions, we compute a robust estimate of the alignment by exploiting generic or periodic actions, such as walking or running, and isolated or anomalous events, such as a jump or a fall. The key insight is that, even if the same action appears differently from different viewpoints (Fig. 2), repetitions of the same pattern are approximatively view invariant. Unlike existing works, we use an alignment algorithm that models synchronization as a frame association problem, instead of a continuous time warping. The use of a view-invariant description of objects actions allows us to align videos *independently of restrictions on the geometry of the observed scene* using a multiscale representation of the actions over time to compare each instant being invariant w.r.t. time misalignment. The software of the proposed method is available at *http://www.eecs.qmul.ac.uk/~andrea/software.htm*.

The reminder of the paper is organized as follows. Section II overviews the state of the art on video alignment and analyses the strength and weakness of existing algorithms. In Section III we describe the proposed method. Section IV discusses the experimental results and comparisons, whereas Section V concludes the paper.

II. Prior work

Video alignment methods differ on the assumptions they require on the positioning and pose of the cameras (i.e. their

L. Zini and F. Odone are with the Dipartimento di Informatica Bioingegneria Robotica e Ingegneria dei Sistemi (DIBRIS) Università degli Studi di Genova, Genova, Italy {luca.zini,francesca.odone}@unige.it

A. Cavallaro is with the Centre for Intelligent Sensing, Queen Mary University of London, UK. andrea.cavallaro@eecs.qmul.ac.uk

This work was done when the first author was visiting Queen Mary University of London.

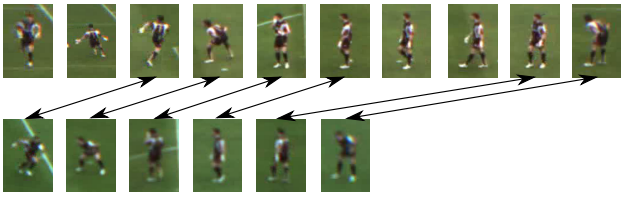


Fig. 2. Example of an action captured by two frontal cameras with different frame rates. Even if the setting is simple the appearance of each frame differs significantly.

geometry) and on the structure of the time misalignment among the recorded videos.

Algorithms might require an *explicit geometry* estimate as input to compute the alignment [6], [10]. Starting from a tracked object and an estimate of the fundamental matrix, one can look for time alignments that result in geometrically consistent tracks. Epipolar geometry is exploited directly looking for intersections of tracks and epipolar lines in [6]; whereas the trajectory is warped to make it a valid set of points for the computation of the fundamental matrix in [10]. Other methods use *implicit geometry* information based on the extraction of high-level features [11], [5], [1], [4], [8]. To derive alignment information, visual changes among two sequences can be correlated [11], or spatio-temporal corners can be matched using the local jet [5]. The timeline of each video can be represented by a descriptor of the repetitions of the same configuration of a tracked point, warping the videos to minimize the distance between descriptions [1]. This work shares some similarities with the proposed approach, as discussed in Sec. III-D. Assuming similar Points of View (POV) of the cameras, it is possible to use methods that maximize visual similarity [3], [8]. Finally, methods exist that compute directly the alignment *without any geometric information* by matching positions and trajectories with robust statistics and then estimating the alignment [12] or both geometry and alignment [7], [4]. The computation of the solution requires a filtered set of matches, which can be obtained either with assumptions on the geometry (e.g. planar [7]), or by using heuristics (e.g. [4]), which restrict the type of camera configurations whose videos can be aligned.

As for the time misalignment, algorithms might assume a *constant shift* [7], [11], [8], [12], [13], an *affine warping* [4], [6], [14], [15], or just a *monotonic* relationship among recordings [1], [10], [3], [16]. All methods making the weak monotonic assumption use Dynamic Time Warping (DTW) to obtain the alignment [1], [10], [17], [3]. DTW [18] is a similarity measure between sequences that assigns *each* frame of a video to at least one frame of the other video, thus assuming a continuous transformation of the timeline. This is in general a very strong assumption as it re a complete association that is not available in case of stream interruptions or frame drops.

Table I summarizes the main characteristics and the assumptions of state-of-the-art alignment algorithms. In addition to geometry independence and the assumption on the temporal misalignment, the table compares the capability of methods to work with only partially overlapped Fields of View (FOV) and with moving cameras.

TABLE I
COMPARISON OF CHARACTERISTICS AND ASSUMPTIONS OF ALIGNMENT ALGORITHMS. WHEN AN ALGORITHM HAS NO EXPLICIT LIMITATIONS OR PROOF FOR A FEATURE, THE SYMBOL "-" IS USED. (KEY. H: HOMOGRAPHY, F: FUNDAMENTAL MATRIX; FOV: FIELD OF VIEW).

Method	Moving cameras support	Geometry independence	Partially overlapped FOV	Temporal warping
[4]	no	F	yes	affine
[1]	yes	yes	no	monotonic
[6]	no	F	yes	affine
[10]	no	F	-	monotonic
[19]	no	F	-	affine
[5]	-	-	yes	affine
[7]	no	H	-	constant
[11]	yes	-	-	constant
[17]	no	H	-	monotonic
[8]	yes	-	-	constant
[20]	no	H	-	constant
[14]	no	yes	yes	affine
[21]	no	yes	-	constant
[3]	no	no	no	monotonic
[15]	yes	yes	yes	affine
[22]	no	H	yes	monotonic
[12]	no	yes	yes	constant
[13]	no	F	yes	affine
[16]	no	H	yes	monotonic
proposed	yes	yes	yes	monotonic

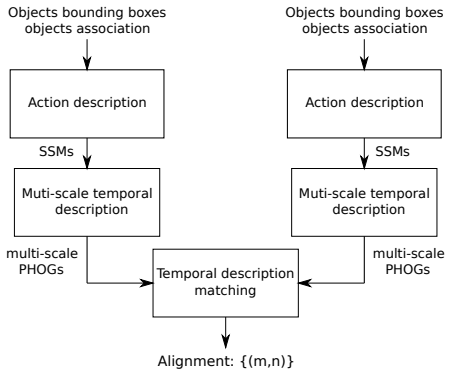


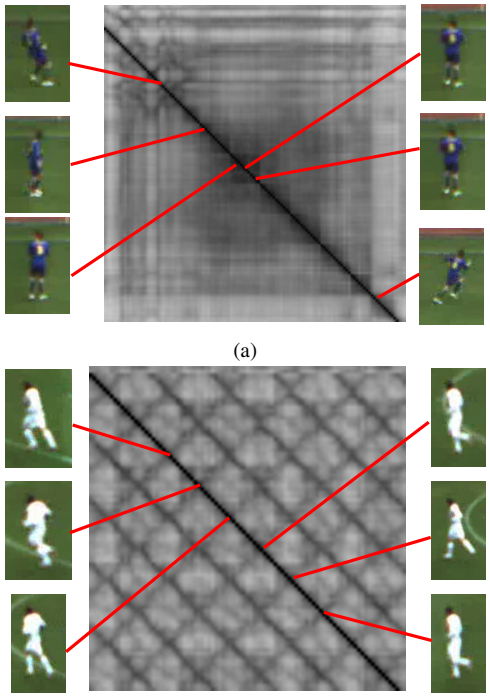
Fig. 3. Block diagram of the proposed approach. Given the segmented and associated objects, we compute a set of SSM-based action descriptions from which we extract a temporal representation that is used to match frames for the alignment

III. PROPOSED METHOD

The proposed algorithm is a two-step approach to video synchronization. In the first step we extract from each camera independently a description of the action of moving objects. In the second step we fuse and compare the data from all the cameras to produce the video alignment (Fig. 3). These two steps are detailed below.

A. Action description

Let $V_1 = \{f_1^i \ : \ i = 1, 2, ..., N_1\}$ and $V_2 = \{f_2^j \ : \ j = 1, 2, ..., N_2\}$ be two views of the same scene, where f_1^i and f_2^j are their frames whose total number is N_1 and N_2. Let V_i^k be a sequence of observations of object k in V_i (i.e. the region inside the bounding boxes of the tracked object)



(a) (b)

Fig. 4. Examples of structures induced on a Self-Similarity Matrix by Histogram-of-Oriented-Gradients description of moving articulated objects. (a) A person standing and producing limited movements generates an uniform block; (b) a walking person generates a regular grid.

and let $|V_i^k|$ be the duration, in frames, of V_i^k. Given object association information, we first extract a description of the action of the objects detected and tracked in each camera and we encode their appearance variations. To this end, we describe the appearance of objects within each bounding box as a sequence of Histogram of Oriented Gradients (HOG) [23] in each view. Then we compute a $|V_i^k| \times |V_i^k|$ Self-Similarity Matrix (SSM), S_i^k, as [1]

$$S_i^k(y,x) = 1 - ||\phi(V_i^k, x) - \phi(V_i^k, y)||_2, \qquad (1)$$

where $\phi(V_i^k, j)$ is the HOG description of the area containing object k in frame j of V_i and x and y are frame indexes.

An example of structures induced on the SSM by common actions of articulated objects is shown in Fig. 4. A person standing generating small movements and a person walking produce very different structures that we will aim to match between the two views.

B. Multi-scale temporal description and matching

Our objective is to define the set $A_{1,2} = \{(f_1^m, f_2^n) \ : \ m \leq N_1 \ , \ n \leq N_2 \ \}$ of frame pairs that were acquired at the same time instant. Let $T_i(f_i^j) = t_i^j$ be a function that computes the timestamp for each frame of V_i. When the frame rate is not fixed (due for example to bandwidth limitations or frame dropping), we can only assume that $T_i(\cdot)$ and its inverse are monotonic. When the frame rate can be modelled as constant, this leads to an affine relation between the frame indexes of the two videos: $T_1(i) = T_2(\alpha i + \beta)$, where α models the frame rate difference and β is the offset between the first frame acquired by each camera (i.e. the offset between V_1 and V_2). Finally, when the frame rate is known or assumed to be constant and identical in both videos, we can relate the two functions $T_i(\cdot)$ with a constant shift only (Fig. 5).

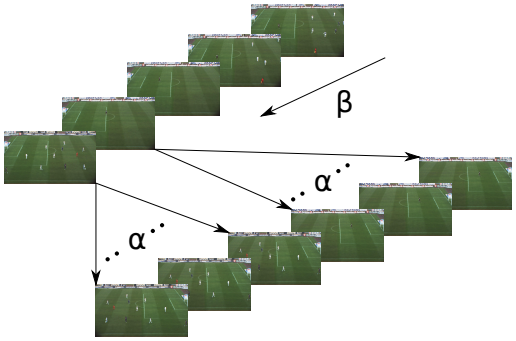


Fig. 5. Example of two videos warped with an affine transformation. The first frame is delayed by β frames and their frame rates differ by a factor α.

We convert the description of the action of each object in a structure invariant to time misalignment that describes how the object appearance changes over time. The key idea we pursue is to create a multiscale description of $S_i^k(.)$ for each video V_i and each object k with two aims: to obtain a more representative description and to match precisely structures with different scales on the SSM (i.e. different video frame rates).

To this end, we extract a description from each point of the diagonal of the matrix, aiming to capture the structure of the SSM (i.e., the structure of the repetition) in a time interval centred on the frame under analysis. To be invariant with respect to time misalignment means that the description extracted from the matrix must be independent of the warping of the matrix itself. We assume that, locally, the time misalignment can be modelled with a linear function, whose effect is a resizing of the SSM (Fig. 7).

We use a Polar HOG (PHOG) structure that has been shown effective in [1], [24]. We devise an alternative grid (Fig. 6) to better deal with the SSMs borders and to avoid the use of small cells, whose histograms may be less stable with a few samples (i.e. small radii of the description). Also, instead of estimating the radius of the PHOG from the maximum of the Laplacian [25] as proposed in [1], we compute a multiscale description that embeds information of different time extents (radii). More in details, in frame f_i^j for each object k we extract a multiscale PHOG description $P_i^k(j)$ from $S_i^k(.)$ centred in (j,j)

$$P_i^k(j) = \{p(S_i^k, j, r_l) \ \ \forall r_l \in R\}, \qquad (2)$$

where $p(S_i^k, j, r_k)$ is the PHOG description with radius r_l centred on pixel (j,j) and R is the set of radii parametrized by the minimum radius r_{min}, the maximum radius r_{max} and a constant b. The radius of the level l of the multiscale representation has a radius $r_l = r_{min} b^l$.

We now want to compare two descriptors, $P_1^k(m)$ and $P_2^k(n)$, using an invariant measure $D(\cdot)$. These descriptors contain a subset of levels that are in common and are shifted in the representation (see Fig. 8). The distance $D(\cdot)$ is computed as the distance between the optimal alignment of the two

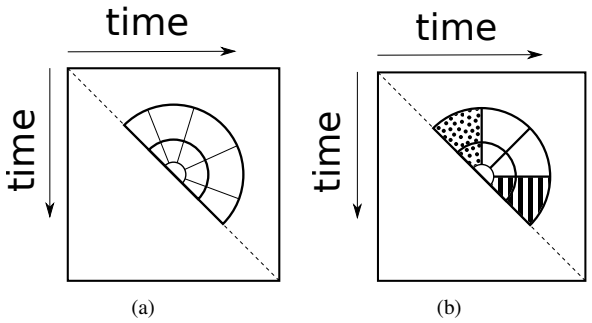


Fig. 6. Comparison between structures used to compute the PHOG description. (a) Structure used in [1]; (b) structure used in the proposed method. The stripped area contains all the possible comparisons between the considered instant (i.e. the center of the support of the description) and the past, the dotted one compares it with the future. In the corners of the SSM at least one of them can be computed and it contains all the available information. Moreover all the cells of (b) have the same size.

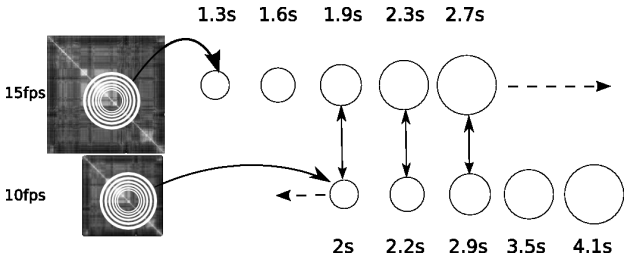


Fig. 7. A comparison of two SSMs with different frame rates. The distance between their multiscale description is computed by shifting one description until a minimum distance is reached. The correct scale for the alignment is shown with black arrows and is the one that gives the best match between the temporal scale (in seconds) of the representation. The overall distance will be the mean of the distances of the descriptors connected by the lines.

descriptions:

$$D(P_1^k(m), P_2^k(n)) = \min_s \frac{1}{L_s} \sum_{l=0}^{|R|} ||P_1^k(m)_{l+s} - P_2^k(n)_l||, \tag{3}$$

where the difference is zero if $l+s \leq 0$ or if $l+s > |P_i^k|$ and L_s is the number of the levels common to the two multi-scale representations after an appropriate shifting (see Fig. 7). The functional is minimized with an exhaustive search over the parameter s. To compute the alignment, we extract the list of distances $D(P_1^k(m), P_2^k(n))$ from all possible pairs (m, n) for each object k in the scene. The result is used to derive the sequence of the desired paired frames

$$A_{1,2} = \{(f_1^m, f_2^n) \ : \ m \leq N_1 \, , \ n \leq N_2 \ \} \tag{4}$$

The alignment algorithm *should* be able to manage explicitly situations where a description has not a correspondence in the other video (due to different fields of view, occlusions, or time misalignment) with a predictable action. This is not feasible with DTW, as it searches for *at least one* correspondence for each frame. To overcome this problem, we propose to use the Needleman-Wunsch string alignment algorithm [26] that can be optimized with dynamic programming. The notion of a gap in a string can be transferred to a frame that has no correspondence in the other video. To compute the alignment,

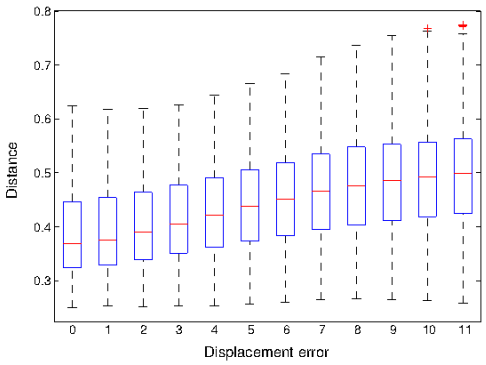


Fig. 8. Distances between descriptions computed by varying their relative misplacement (horizontal axis) from the correct value (0 displacement) on a test football sequence.

we first create the $N_1 \times N_2$ matrix C_{NW}:

$$\begin{aligned} &C_{NW}(m,n) = \max((1 - d(m,n)) + \\ &C_{NW}(m-1,n-1), C_{NW}(m-1,n) + G, \\ &C_{NW}(m,n-1) + G), \end{aligned} \tag{5}$$

where $d(\cdot)$ is the distance function between frame m of V_1 and frame n of V_2, and G is a constant, defined in the range $[0, 1]$, that controls the similarity of a frame without association. The alignment is found by looking at all the coordinate pairs that give the maximum-score path starting from the lower-right corner of matrix C_{NW} and moving toward the upper-left corner (see Fig. 9). The difference between NW and DTW is in that the latter looks for the minimum-score path on the matrix:

$$\begin{aligned} &C_{DTW}(m,n) = d(m,n) + \\ &\min(C_{DTW}(m-1,n-1), \\ &C_{DTW}(m-1,n), C_{DTW}(m,n-1)), \end{aligned} \tag{6}$$

i.e. the cost of an unpaired frame in NW is fixed. The use of the parameter G gives also the possibility to tune the tradeoff between trusting the data (and the noise they contain) and having a smooth solution that strongly penalizes frame dropping. As with DTW, in case of strong noise or very weak signal, the solution will be biased toward the diagonal of the matrix: in case of DTW this is true as the diagonal of C is the shortest path between the two corners and, in the presence of uniform similarities, it will be the path that accumulates the smallest cost. In the case of NW, this depends on the value of G that, if it is set too small, will block the algorithm from discarding frames.

We compute the distance $d(\cdot)$ between two frames in NW as

$$d(m,n) = \mathcal{M}_{k \in B}(D(P_1^k(m), P_2^k(n))), \tag{7}$$

where $\mathcal{M}$ is the median that filters out the noise on the tracking data and errors due to occlusions between objects, and $B = O_{V_1(m)} \cap O_{V_2(n)}$, with $O_{V_1(m)}$ ($O_{V_2(n)}$) being the list of objects that appear in $V_1(m)$ ($V_2(n)$).

C. Computational complexity

First, the algorithm extracts the HOG description from each bounding box and updates the SSM by comparing each HOG

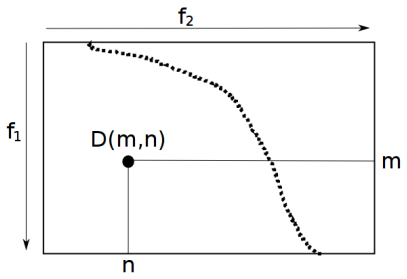


Fig. 9. Given the matrix containing all the distances $d(m,n)$ between the frames f_1^m of the first video and f_2^n of the second video, the alignment algorithm looks for the path from the lower-right corner (the end of both the videos) to the upper-left corner (the beginning of the videos) with the aim of passing through the coordinates that correspond to the indexes of instants framing the same action.

against the last $2r_{max}$ descriptions. Since the information needed to compute a PHOG description is in the lower right part of the SSM of size (r_{max}, r_{max}), the rest can be discarded. Therefore the computational cost of these operations is *constant* with time and *linear* with the number of objects in the scene. Second, the algorithm extracts the PHOG description of each instant and for each track. In this case the computational cost is *constant* for each object and for each instant and is *linear* with the number of frames and the number of objects.

The object descriptions are then used within the NW algorithm. The cost of each comparison is *linear* with the number of objects and *constant* with all the other variables, hence the cost of this step and of the whole algorithm is dominated by the cost needed to fill the matrix C_{NW} that is equal to the product of the number of frames used for the alignment.

For long video streams the quadratic complexity of the algorithm can be bounded by limiting the size of the matrix C_{NW} obtaining a constant computational cost at each step to the price of bounding the maximum misalignment that it is possible to recover to a fixed amount of frames. All the steps until the computation of the alignment depend only on the number of objects. Since no information sharing is needed these may be computed in a distributed way directly across cameras.

With respect to the algorithm proposed in [1], where the description is composed by one level only, we have a penalty on the comparison of two frames that is linear with the number of levels used. However, since the levels are fixed, it is only a constant factor that does not influence the final complexity of the algorithm. Moreover, we do not need to compute the Laplacian for all the possible radii of the PHOG description as in [1], and, since all the parameters are fixed, all the steps in analysing an object terminate always in the same fixed time. Bounding the size of C_{NW} an optimized implementation of the algorithm can run in real-time.

D. Discussion

The general idea of using the SSM to generate a view-invariant representation was used in [24] for action recognition and in [1] for video alignment. Unlike [1], where the analysis is performed at feature level (the input is the tracking of a set of points extracted from the observed videos), we use explicitly the appearance of multiple objects and exploit implicitly and simultaneously information from the pose of the object and its motion. Differently from [24], [1], we extract a multi-scale descriptions from multiple SSMs whose structure is explicitly stored and exploited in Eq. 3 to take into account the misalignment. Moreover the alignment algorithm differs both from [24] and [1] and allows to handle discontinuous time warping. As the proposed approach aligns video pairs, multiple video streams can be aligned pairwise or by selecting a common reference video to align the others.

Unlike methods based on assumptions on geometry [6], [4], we remove constraints such as planar scene, known geometry, restrictions on the time misalignment. Moving cameras are supported when the motion does not change significantly the viewpoint of the observed object within the duration of the description window (r_{max} frames). The cost for relaxing the assumptions on the geometry, the camera configurations and the restrictions on the temporal misalignment is that we reach a frame-level accuracy, instead of a sub-frame accuracy [6], [4].

A source of error for the proposed algorithm is obviously the absence of relevant actions in the time window observed for the alignment. Moreover, when there are just a few objects with similar actions in different time intervals, if the relative order between actions is preserved there can be a mismatch in the alignment (see as example Fig. 10). Overall, the alignment of videos with very different viewpoint and frequent occlusions may need a considerable temporal overlap: as an example the most complex videos tested can be aligned with an error bound to ten frames when the temporal overlap is at least 2/3 of the timeline. Simpler videos can be aligned even if they have larger misalignments. This characteristic is shared by DTW and NW: if the signal is not strong enough with respect to the shift, the cost of moving away from the diagonal of C_{NW} could be dominant in the computation of the solution.

Detailed quantitative discussions and comparisons are presented in the next section.

IV. Experiments

A. Experimental setup

To evaluate and compare the alignment results of the proposed algorithm in real-world scenarios we use two public datasets, namely the ISSIA football dataset [27] and the APIDIS basketball dataset[1]. The ISSIA dataset is captured by six cameras, has 2700 annotated frames acquired from frontal cameras and shows the same configuration complexity on all sequence with an uniform background. The APIDIS dataset is acquired by seven cameras positioned with different orientations and has annotations for 1500 frames with a more complex background and frequent occlusions. From the original dataset we disregard two fisheye cameras and one camera whose FOV is not overlapped with the others. We split the APIDIS dataset in a set composed of sequences with paired viewpoints and a set containing all the other combinations of cameras, which have generic viewpoints (Fig. 11).

[1]http://www.apidis.org/Dataset/

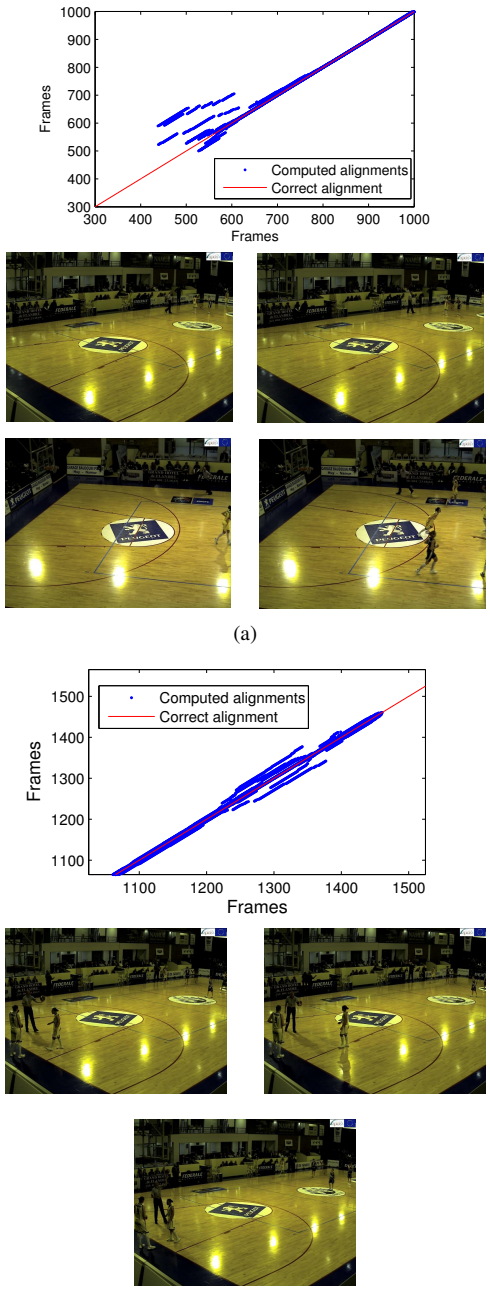


Fig. 10. Example of configurations where the accuracy is reduced in presence of complex warping or noise due to ambiguous data in the scene. (a) Similar actions and only one object in common between the views. (b) In a scene that is static for a long period all the frames are similar.

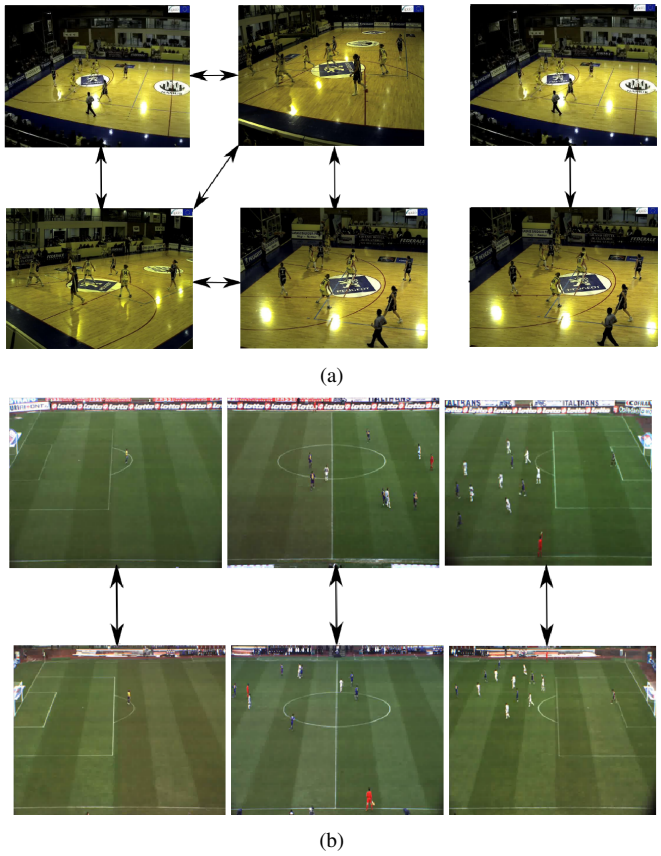


Fig. 11. Datasets used in the experiments. Sample frame from each POV. (a) APIDIS cameras. (b) ISSIA football dataset.

To control and quantify the results, the data to align are created by misaligning the videos according to $t_w = \alpha t - \beta$, with $\alpha \in \{1, 1.1, 1.2, 1.4\}$ and β a constant shifting the frames by reducing the overlap up to 2/3 of the duration of the sequences. The model used is equivalent to a random frame dropping model. All the tests have been repeated removing up to 200 frames by the end of the warped sequence in order to have a solution that is not in correspondence to the diagonal of C_{NW} and to obtain an unbiased estimation of the performances of the algorithm. We consider the starting point of the video in correspondence of the first object. For this reason, the APIDIS videos start with up to 500 frames of difference: setting β varies this displacement.

The following algorithmic parameters are the same for all the experiments. The HOGs have blocks of 2 cell of 16 pixels and 9 bins. The multiscale parameters and G have been chosen to minimize the error on the sequences of three players of the ISSIA dataset (i.e. a subset of a single video): $r_{min} = 20$, $r_{max} = 75$, $b = 1.2$ and $G = 0.2$. In principle r_{min}, r_{max} should be chosen so that their ratio is larger than the expected ratio between the mean frame rates of the two videos. b rules the tradeoff between the precision that can be achieved in the alignment of the descriptions in Eq. 3 and the final computational cost. Instead G acts as a regularization parameter: smaller values promote solutions closer to the identity, larger values encourage frame dropping.

Errors will be reported as the median error (in frames) with respect to the correct timeline. We use as reference the errors reported in [1] that, using *frontal* cameras *shifted and not warped* (i.e. $\alpha = 1$) reports an error of 7.29 frames. Our aim is to work with different FOVs and general viewing angles with an error bounded by the same order of magnitude. In the next section we discuss the results of the comparison between our proposed approach and [1], which is the only method in the literature that shares similar hypotheses to ours (i.e. non-parametric time misalignment, moving cameras support, geometry free model) and thus the only meaningful comparison.

TABLE II
MEAN ERRORS OBTAINED WITH DIFFERENT WARPING PARAMETERS α.

	α			
	1	**1.1**	**1.2**	**1.4**
APIDIS (camera 1 and 7)	1.00	1.00	1.50	1.50
APIDIS (all other cameras)	4.00	4.49	3.83	4.33
ISSIA	0.66	1.16	2.00	2.16

B. Comparative analysis

Because in [1] the input used to compute the SSM makes the algorithm not suitable to work with different FOVs, in order to conduct a fair comparison of the individual steps we use their pipeline with the HOG description in input. In order not to modify the original pipeline we use only one object (player) for each sequence. The comparative analysis is carried out considering all the objects with a long continuous track in the football dataset, and the four objects with the longest continuous track from camera 1, 2, and 7 in the basketball dataset (combination used: camera 1 and 2; and camera 1 and 7).

The improvement of our pipeline compared to that of [1] are computed as relative difference:

$$E = \frac{1}{N}\sum_{i}^{N}\frac{E_l(i) - E_m(i)}{min(E_m(i), E_l(i))}, \tag{8}$$

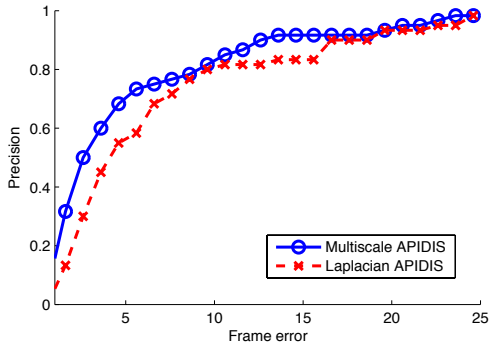
where N is the number of experiments, E_l is the set of the errors (in frames) obtained in the tests with the original method based the maximum of the Laplacian, while E_m contains the corresponding results obtained with our multiscale feature comparison. Based on the experiment described above we obtained a consistent improvement over the original algorithm, with $E = 1.49$ for the basketball dataset and $E = 0.96$ on the football dataset (see also Fig. 12).

The second comparison is between the commonly adopted DTW and the method we propose based on NW (Fig. 13). It is possible to notice a clear superiority of NW, which is mainly due to parts of the videos that have not got correspondence and for the effect of static scenes, on which NW is more robust. In Fig. 14 are visible two examples of two warped timeline of two videos to show how NW is more stable and robust to ambiguous configurations retaining the ability to identify real shifts in the data. In complex configurations where the NW algorithm fail due to lack of sufficient information in the data, DTW usually fails similarly (see Fig. 14 (b)).

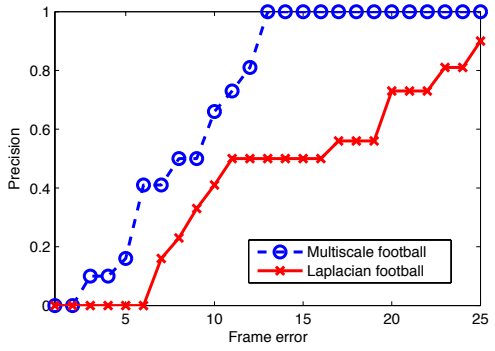
C. Overall evaluation

This section discusses the results obtained on the full dataset and the analysis of the robustness of the proposed approach.

Table II summarizes the results of the proposed algorithm obtained with all the objects in the scene using different values of α; β is not reported explicitly as it has no significant effect. Fixing α and varying β has in fact produced a standard deviation of the error of 0.63 frames. A more critical parameter of our algorithm is the number of objects that are needed to align the video to a certain accuracy. Fig. 15 shows the error for a different number of objects in ten random experiments. The algorithm starts being stable with five objects. This result



(a)



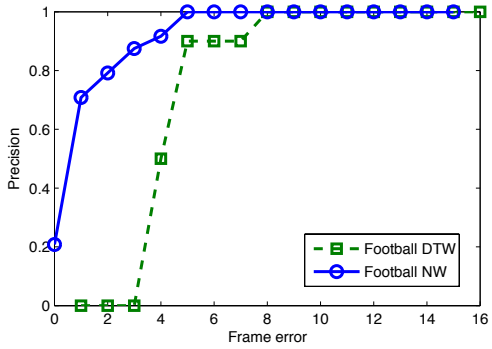
(b)

Fig. 12. Comparison of our pipeline with the multiscale PHOG descriptor with the PHOG-Laplacian descriptor used in [1] on a subset of the (a) ISSIA and (b) APIDIS dataset chosen to be compatible with [1] (see text for details). The gap of performance is more visible with the more complex sequence of the APIDIS dataset.

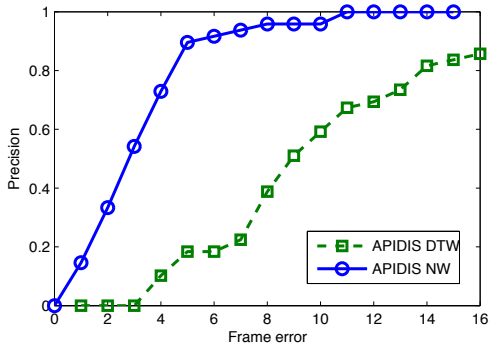
is an upper bound of the error as objects were extracted randomly and could not be always in the scene at the same time. Moreover, the results with few people also suffer from the occlusions by all the other objects that are still present in the video, even if in this experiment they are not considered for the alignment.

To analyse the robustness of the proposed algorithm to work with noisy detections we modify the object bounding boxes by varying their size and position with uniform random noise. To reach significant results we test the noise with $\alpha = 1.4$ using β equal to 1/3 of the video. Fig. 16 shows the results: the noise is the maximum displacement due to a given noise in two consecutive frames, and is expressed as a percentage of the size of each side of the bounding box of an object. The break point of the proposed method in this difficult setting is above 10% of noise (see Fig. 17). Notice that the original annotation data themselves are not always accurate and therefore this amount of noise induces considerable errors that can be usually attenuated by dynamic filters. As a reference an unfiltered Mean-Shift tracker [28] used on a random subsampling of sequences with no id-switch has shown an error comparable to 4% of uniform noise (see Fig.17).

Another source of noise is given by association errors both during the tracking and between the camera views. The experiments in this setting have shown that, thanks to the filtering effect of the median used to compare the frames, with the 20% of wrong associations, the algorithm is able to align

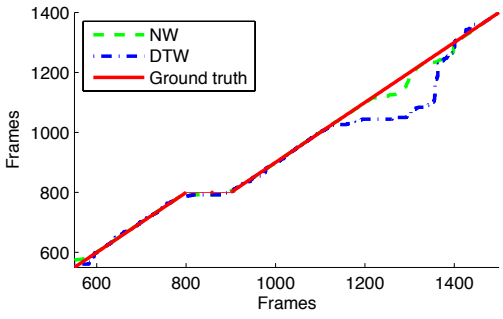


(a)

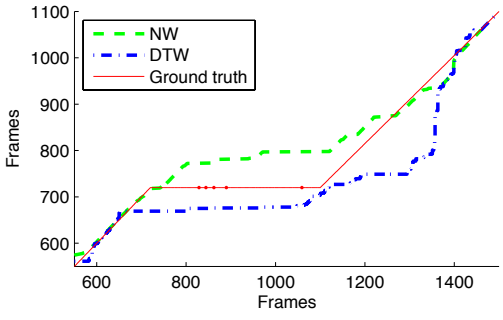


(b)

Fig. 13. Comparison of NW with the commonly used DTW on a subset of the (a) ISSIA and (b) APIDIS dataset.

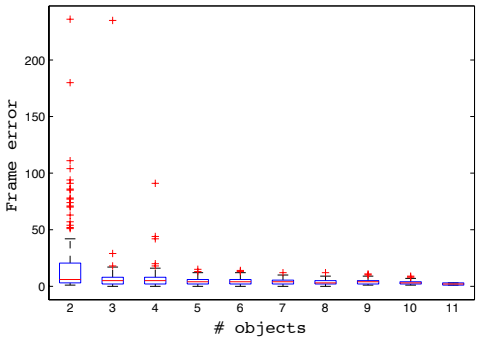


(a)

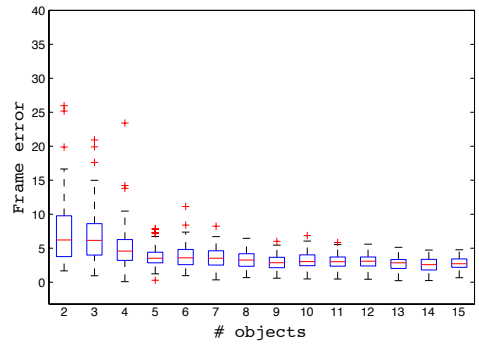


(b)

Fig. 14. Example of two alignments from real noisy data computed with DTW and NW. It is visible (a) how the parameter G affects the action in ambiguous situations by regularizing the solution, but it does not compromise the ability to adapt to useful signal in the data. In more critical scenarios with greater noise and big gaps in the data w.r.t. the length of the timeline both the algorithm may fail in similar ways (b).



(a)



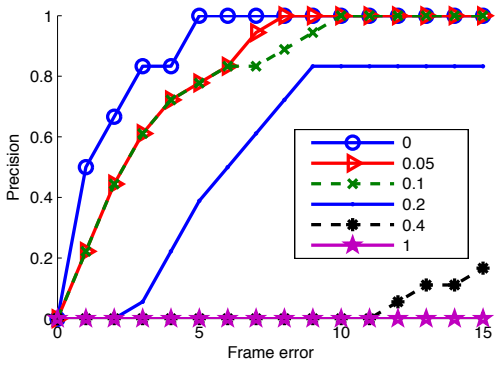
(b)

Fig. 15. Comparison of the performance of the proposed algorithm with a varying number of objects used for the alignment. The errors are generated using randomly selected objects of different cardinality and different misalignment parameters. (a) Results for the APIDIS dataset. (b) Results for the football dataset. The results shown are an upper bound to the error of the algorithm (see text for details). Note that, to include the extreme elements in the visualization, the vertical scales of the two plots are different.

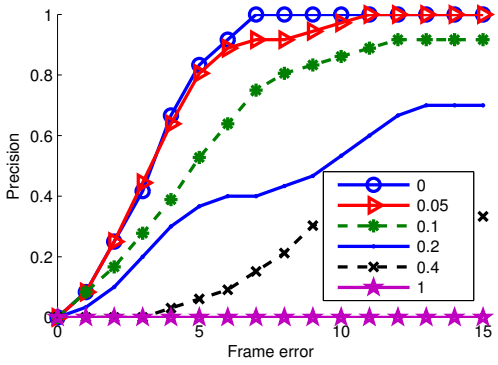
all the videos in the football dataset and the 92% of the difficult sequences of the basketball dataset introducing at most 3 frames of error. This result is significant since our experiments have shown that, by using a brute force approach on subsets of data to minimize the functional of NW, it is possible to ignore the input association obtaining the equivalent of the 15% of errors on the ID switches.

V. Conclusions

We presented a general method for video alignment based on the observation of the actions of multiple objects. Given object association information, the proposed method can work in real-world complex scenarios without assumptions or requirements on the geometry, on the similarity of the points of view and on the structure of the warping functions. The main novelties of the proposed method are the use of an algorithm for the alignment that models the problem as a frame association, instead of a continuous time warping; the use of a multiscale representation of the actions in time to compare each instant being invariant w.r.t. time misalignment; and the combination of HOGs and SSMs in the context of video alignment. The proposed method was validated on two datasets of real sport sequences, reaching an accuracy of a few frames, which is superior to the results obtained by [1]

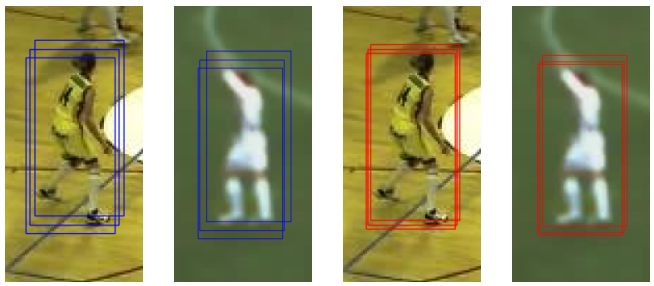


(a)



(b)

Fig. 16. Performance of the proposed alignment algorithm when adding uniform random noise on the position of the bounding box. The range of the random movements is reported in the legend as a percentage of the size of the sides of the bounding box. (a) Errors on the football dataset. (b) Errors on the APIDIS dataset.



(a) (b)

Fig. 17. Examples of shifts from the correct position of the bounding boxes that can be observed in two consecutive frames with two level of noise. (a) 10% of noise. (b) The level of noise registered with a mean-shift tracker (≈4%).

on simpler datasets. The source code of the proposed algorithm will be made available to the research community.

Our future work includes the estimation of a confidence value for the alignment estimate in order to automatically detect which videos cannot be aligned (instead of generating a wrong alignment) and to extend the algorithm to the problem of retrieval, that is the ability of aligning videos even in presence of large displacements. Moreover the method can be implemented on smart cameras to align video streams with a small delay. Thanks to the low constraints on the system, on the scene and the low amount of (meta)data to be transferred for computing the alignment, the proposed approach fits well such scenario.

REFERENCES

[1] E. Dexter, P. Pérez, and I. Laptev, "Multi-view synchronization of human actions and dynamic scenes," in *BMVC*, 2009.
[2] Y. Ukrainitz and M. Irani, "Aligning sequences and actions by maximizing space-time correlations," *ECCV*, pp. 538–550, 2006.
[3] J. Serrat, F. Diego, F. Lumbreras, and J. Álvarez, "Synchronization of video sequences from free-moving cameras," *PRIA*, pp. 620–627, 2007.
[4] Y. Caspi, D. Simakov, and M. Irani, "Feature-based sequence-to-sequence matching," *IJCV*, vol. 68, no. 1, pp. 53–64, 2006.
[5] D. Wedge, D. Huynh, and P. Kovesi, "Using space-time interest points for video sequence synchronization," in *IAPR*, 2007, pp. 190–194.
[6] F. Pádua, R. Carceroni, G. Santos, and K. Kutulakos, "Linear sequence-to-sequence alignment," *IEEE TPAMI*, vol. 32, no. 2, pp. 304–320, 2010.
[7] G. Stein, "Tracking from multiple view points: Self-calibration of space and time," in *CVPR*, vol. 1. IEEE, 1999, pp. 1521–1527.
[8] T. Tuytelaars and L. Van Gool, "Synchronizing video sequences," in *CVPR*, vol. 1. IEEE, 2004, pp. I–762.
[9] P. Shrestha, M. Barbieri, H. Weda, and D. Sekulovski, "Synchronization of multiple camera videos using audio-visual features," *IEEE TMM*, vol. 12, no. 1, pp. 79–92, 2010.
[10] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood, "View-invariant alignment and matching of video sequences," in *ICCV*. IEEE, 2003.
[11] M. Ushizaki, T. Okatani, and K. Deguchi, "Video synchronization based on co-occurrence of appearance changes in video sequences," in *ICPR*. IEEE, 2006.
[12] X. Lin, V. Kitanovski, Q. Zhang, and E. Izquierdo, "Enhanced multi-view dancing videos synchronisation," in *WIAMIS*. IEEE, 2012, pp. 1–4.
[13] A. Elhayek, C. Stoll, K. Kim, H. Seidel, and C. Theobalt, "Feature-based multi-video synchronization with subframe accuracy," *Pattern Recognition*, pp. 266–275, 2012.
[14] Y. Caspi and M. Irani, "Spatio-temporal alignment of sequences," *IEEE TPAMI*, pp. 1409–1424, 2002.
[15] M. Yang, Y. Liu, and Z. You, "Video synchronization based on events alignment," *Pattern Recognition Letters*, pp. 1338–1348, 2012.
[16] R. Li and R. Chellappa, "Aligning spatio-temporal signals on a special manifold," *ECCV 2010*, pp. 547–560, 2010.
[17] C. Lu, M. Singh, I. Cheng, A. Basu, and M. Mandal, "Efficient video sequences alignment using unbiased bidirectional dynamic time warping," *JVCIR*, vol. 22, no. 7, pp. 606–614, 2011.
[18] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Trans. Acoustics Speech and Sig. Proc.*, vol. 26, no. 1, pp. 43–49, 1978.
[19] P. Tresadern and I. Reid, "Video synchronization from human motion using rank constraints," *Computer Vision and Image Understanding*, vol. 113, no. 8, pp. 891–906, 2009.
[20] C. Dai, Y. Zheng, and X. Li, "Accurate video alignment using phase correlation," *Signal Processing Letters, IEEE*, vol. 13, no. 12, pp. 737–740, 2006.
[21] L. Wolf and A. Zomet, "Wide baseline matching between unsynchronized video sequences," *IJCV*, vol. 68, no. 1, pp. 43–52, 2006.
[22] C. Lu and M. Mandal, "An efficient technique for motion-based view-variant video sequences synchronization," in *ICME*. IEEE, 2011, pp. 1–6.
[23] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, vol. 1. IEEE, 2005, pp. 886–893.
[24] I. Junejo, E. Dexter, I. Laptev, and P. Pérez, "View-independent action recognition from temporal self-similarities," *IEEE TPAMI*, vol. 33, no. 1, pp. 172–185, 2011.
[25] T. Lindeberg, *Scale-space theory in computer vision*. Springer, 1993.
[26] S. Needleman and C. Wunsch, "A general method applicable to the search for similarities in the amino acid sequence of two proteins," *Journal of molecular biology*, vol. 48, no. 3, pp. 443–453, 1970.
[27] T. D'Orazio, M. Leo, N. Mosca, P. Spagnolo, and P. Mazzeo, "A semi-automatic system for ground truth generation of soccer video sequences," *IEEE AVSS*, pp. 559–564, 2009.
[28] D. Comaniciu and P. Meer, "Mean shift: A robust approach toward feature space analysis," *IEEE TPAMI*, vol. 24, no. 5, pp. 603–619, 2002.