View-Independent Action Recognition from Temporal Self-Similarities

Imran Junejo, Emilie Dexter, Ivan Laptev, Patrick Pérez

View-Independent Action Recognition from Temporal Self-Similarities

Imran N. Junejo, *Member, IEEE,* Emilie Dexter, Ivan Laptev, and Patrick Pérez

***Abstract*— This paper addresses recognition of human actions under view changes. We explore self-similarities of action sequences over time and observe the striking stability of such measures across views. Building upon this key observation, we develop an action descriptor that captures the structure of temporal similarities and dissimilarities within an action sequence. Despite this temporal self-similarity descriptor not being strictly view-invariant, we provide intuition and experimental validation demonstrating its high stability under view changes. Self-similarity descriptors are also shown stable under performance variations within a class of actions, when individual speed fluctuations are ignored. If required, such fluctuations between two different instances of the same action class can be explicitly recovered with dynamic time warping, as will be demonstrated, to achieve cross-view action synchronization. More central to present work, temporal ordering of local self-similarity descriptors can simply be ignored within a bag-of-features type of approach. Sufficient action discrimination is still retained this way to build a view-independent action recognition system. Interestingly, self-similarities computed from different image features possess similar properties and can be used in a complementary fashion. Our method is simple and requires neither structure recovery nor multi-view correspondence estimation. Instead, it relies on weak geometric properties and combines them with machine learning for efficient cross-view action recognition. The method is validated on three public datasets. It has similar or superior performance compared to related methods and it performs well even in extreme conditions such as when recognizing actions from top views while using side views only for training.**

***Index Terms*— Human Action Recognition, Human Action Synchronization, View Invariance, Temporal Self-Similarities, Local Temporal Descriptors**

I. Introduction

Visual recognition and understanding of human actions have attracted much attention over the past three decades [1], [2] and remain an active research area of computer vision. A good solution to the problem holds a yet unexplored potential for many applications such as the search and the structuring of large video archives, video surveillance, human-computer interaction, gesture recognition and video editing. Recent work has demonstrated the difficulty of the problem associated with the large variation of human action data due to the individual variations of people in expression, posture, motion and clothing; perspective effects and camera motions; illumination variations; occlusions and disocclusions; and distracting effects of scenes surroundings. Also, actions frequently involve and depend on manipulated objects, which adds another layer of variability. As a consequence, current methods often resort to restricted and simplified scenarios with simple backgrounds, simpler kinematic action classes, static cameras or limited view variations.

Various approaches using different constructs have been proposed over the years for action recognition. These approaches can be roughly categorized on the basis of representation used by the authors. Time evolution of human silhouettes was frequently used as action description. For example, [3] proposed to capture the history of shape changes using temporal templates and [4] extends these 2D templates to 3D action templates. Similarly, the notions of *action cylinders* [5], and *space-time shapes* [6]–[8] have been introduced based on silhouettes. Recently, space-time approaches analyzing the structure of local 3D patches in the video have been shown promising in [9]–[13]. Using space-time or other types of local features, the modeling and recognition of human motion have been addressed with a variety of machine learning techniques such as Support Vector Machines (SVM) [14], [15], Hidden Markov Models (HMM) [16]–[18] and Conditional Random Fields (CRF) [19]–[23].

Most of the current methods for action recognition are designed for limited view variations. A reliable and a generic action recognition system, however, has to be robust to camera parameters and different view points while observing an action sequence. View variations originate from the changing and frequently unknown positions of the camera. Similar to the multi-view appearance of static objects, the appearance of actions may drastically vary from one viewpoint to another. Differently to the static case, however, the appearance of actions may also be affected by the dynamic view changes of the moving camera.

Multi-view variations of actions have been previously addressed using epipolar geometry such as in [5], [24]–[28], by learning poses seen from different viewpoints [29]–[33] or by a full 3D reconstruction [34], [35]. Such methods rely either on existing point correspondences between image sequences or/and on many videos representing actions in multiple views. Both of these assumptions, however, are limiting in practice due to (i) the difficulty of estimating non-rigid correspondences in videos and (ii) the difficulty of obtaining sufficient video data spanning view variations for many action classes.

In this work we address multi-view action recognition from a different perspective and avoid many assumptions of previous methods. In contrast to the geometry-based methods above we require neither the identification of body parts nor the estimation of corresponding points between video sequences. Differently to the previous view-based methods we do not assume multi-view action samples neither for training nor for testing.

Our approach builds upon self-similarities of action sequences over time. For a given action sequence and a given type of low level features, we compute distances between extracted features

• *Imran N. Junejo is with the Department of Computer Sciences, University of Sharjah, U.A.E. E-mail: ijunejo@sharjah.ac.ae*

• *Emilie Dexter is with INRIA Rennes - Bretagne Atlantique, Campus Universitaire de Beaulieu, France. E-mail: emilie.dexter@inria.fr*

• *Ivan Laptev is with INRIA Paris - Rocquencourt / ENS, France. E-mail: ivan.laptev@inria.fr*

• *Patrick Pérez is with Thomson Corporate Research, France. E-mail: Patrick.Perez@thomson.net*

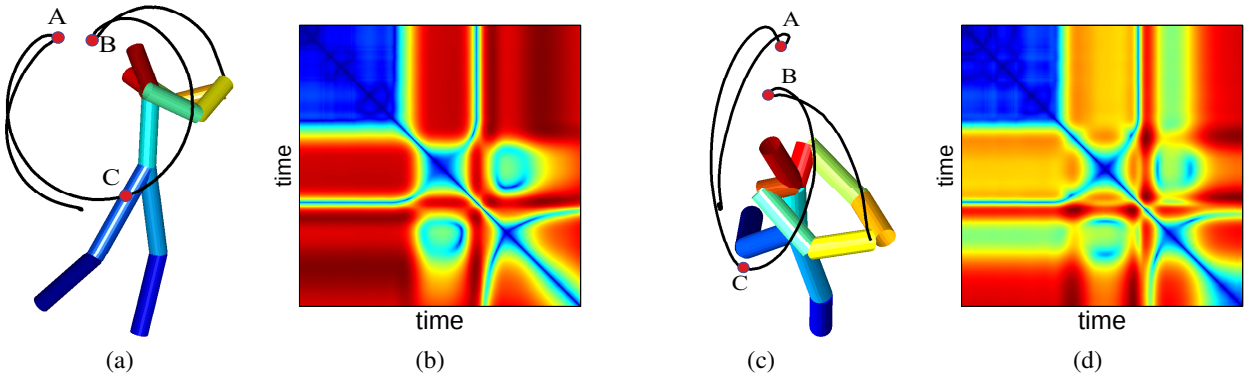


Fig. 1. **Cross-view stability of trajectory-based self-similarity matrices (SSMs) on a simple example**. (a) and (c) demonstrate, based on motion capture (MOCAP) data, a golf swing action seen from two different views. (b) and (d) represent their respective SMMs for the trajectory of one hand projected in corresponding view. Even though the two views are different, the structures or the patterns of the computed SSMs are very similar.

for all pairs of time frames and store results in a Self-Similarity Matrix (SSM). We claim SSMs to be stable under view changes of an action. Fig. 1 illustrates our idea with an example of a golf swing action seen from two different views. For this example we compute SSMs as pair-wise distances between all 2D points on the projected hand trajectories illustrated in Fig. 1(a,c). Despite the view variation, close trajectory points **A** and **B** remain close in both views while the distant trajectory points **A** and **C** have large distances in both projections. The visualizations of SSMs computed for both sequences in Fig. 1(b,d) have a striking similarity despite the different projections of the action. More generally, if body poses of an action are similar at moments t_1, t_2, the value of SSM(t_1, t_2) i.e., the distance between some action descriptors at t_1, t_2 will be low for any view of an action. On the contrary, if the body poses are different at t_1, t_2, the value of SSM(t_1, t_2) is likely to be large for most of the views and non-trivial action descriptors.

In the rest of the paper we operationalize SSMs for human action sequences and deploy them for view-independent action recognition. In particular, we observe similar properties of SSMs computed for different image features and use such SSMs in a complementary fashion.

The paper is organized as follows. In the next section we review related work, with special emphasis on the relationship between SSMs and so-called Recurrence Plots (RPs) which SSM can be seen as an extension of. Section III gives a formal definition of SSM using alternative image features and reports first experiments on mocap data demonstrating its structural stability across views. Section IV describes the proposed representation of action sequences based on local temporal SSM descriptors and demonstrate how this representation is at the same time precise, specific to a class of action and largely view-independent, by using it to synchronize (align temporally) different performances of similar actions. In Section V, we introduce our view-independent action recognition system based on such descriptions and test it on three public datasets. These experiments demonstrate the practicality and the potential of the proposed method. Section VI concludes the paper.

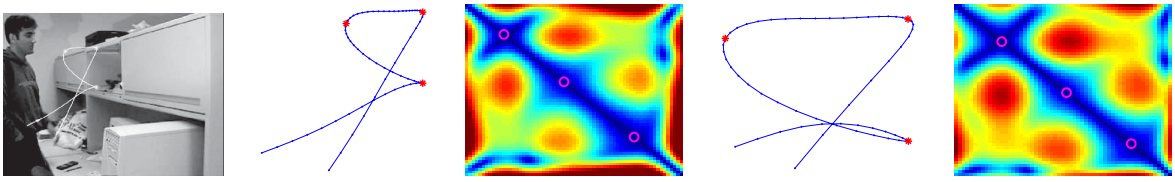
II. Related Work

This paper concerns view-independent action recognition, a topic which has received a considerable attention from researchers recently. To address this problem, [5], [24], [25] employ epipolar geometry. Point correspondences between actions are assumed to be known for imposing fundamental matrix constraints and performing view-invariant action recognition. Rao *et al*. [26] show that the maxima in space-time curvature of a 3D trajectory are preserved in 2D image trajectories, and are also view-invariant. [28] proposes a quasi view-invariant approach, requiring at least 5 body points lying on a 3D plane or that the limbs trace a planar area during the course of an action. Recently [27] showed that for a moving plane the fundamental ratios, i.e. the ratios among the elements in the upper left 2×2 submatrix of the fundamental matrix, are invariant to the camera parameters as well as its orientation and can be used for action recognition. However, obtaining automatic and reliable point correspondences for daily videos with natural human actions is a very challenging and currently unsolved problem, which limits the application of above mentioned methods in practice.

One alternative to the geometric approach is to represent actions by samples recorded for different views. [29]–[32] create a database of poses seen from multiple viewpoints. Extracted silhouettes from a test action are matched to this database to recognize the action being performed. The drawback of these methods is that each action needs to be represented by many training samples recorded for a large and representative set of views. Other methods [35] and [34] perform a full 3D reconstruction from silhouettes seen from multiple deployed cameras. This approach requires a setup of multiple cameras or training on poses obtained from multiple views, which again restricts the applicability of methods in practice.

The approach in [33] exploits transfer learning for constructing view-stable and discriminative features for view-independent action recognition. For a pair of given views, the features are learned from a separate set of actions observed in both views. Given a new action class observed and learned in one view only, transfer learning enables recognition of instances of that class in the second view. While the use of transfer learning in [33] is interesting, the method is limited to a set of pre-defined views and requires training to be done separately for each pair of views. It also requires (non-target) actions to be observed and view-tagged for multiple views. Our method avoids these limitations. We compare our results with [33] on the common benchmark in Section V-C.

The methods most closely related to our approach are that of [36]–[39]. For image and video matching, [36] recently explored



(a) (b) (c) (d) (e)

Fig. 2. **Relationship between proposed SSM representation and *dynamic instances* introduced in [26]**. Two actors perform the action of opening a cabinet door, where the hand trajectory is shown in (b) and (d). The SSMs computed for these two actions based only on one hand trajectory are shown in (c) and (e), respectively. The "dynamic instances" (as proposed by [26]), marked in red stars in (b) and (d), represent valleys in the corresponding SSM, depicted by magenta circles in (c) and (e), respectively. The spread of each valley depends on the peak-width of the corresponding dynamic instance.

local self-similarity descriptors. The descriptors are constructed by correlating the image (or video) patch centered at a pixel to its surrounding area by the sum of squared differences. The correlation surface is transformed into a binned log-polar representation to form a local descriptor used for image and video matching. Differently to this method, we explore the structure of similarities between *all* pairs of time-frames in a sequence. The main focus of our work is on the use of self-similarities for view-invariant action recognition which was not addressed in [36].

Our approach has a close relation to the notion of video self-similarity used by [37], [38]. In the domain of periodic motion detection, Cutler and Davis [38] track moving objects and extract silhouettes (or their bounding boxes). This is followed by building a 2D matrix for the given video sequence, where each entry of the matrix contains the absolute correlation score between the two frames i and j. Their observation is that for a periodic motion, this similarity matrix will also be periodic. To detect and characterize the periodic motion, they resort to Time-Frequency analysis. Following this, [37] uses the same construct of the self-similarity matrix for gait recognition in videos of walking people. The periodicity of the gait creates diagonals in the matrix and the temporal symmetry of the gait cycles are represented by the cross-diagonals. In order to compare sequences of different length, the self-similarity matrix is subdivided into small units. Both of these works focus primarily on videos of walking people for periodic motion detection and gait analysis. The method in [39] also concerns gait recognition using temporal similarities between frames of different image sequences. None of the methods above explores the notion of self-similarity for view-invariant action recognition.

SSM as a Recurrence Plot: Recurrence is a fundamental phenomenon of many dynamical systems. The study of such systems is typically based on recorded data time-series, $\{\mathbf{x}_t,\ t = 1 \cdots T\}$, from which one wants to learn as much information about observed system as possible. Traditional techniques for understanding a dynamical system involve embedding this time-series into an E-dimensional reconstruction phase space using delay coordinates [16]–[18]. This process involves estimation of two parameters, i.e., (i) the embedding dimension E and (ii) the delay, which is a difficult task.

In order to *visualize* the geometry of a dynamical system's behavior, Eckman *et al.* [40] first proposed the *Recurrence Plot* (RP), defined as:

$$RP(i,j) = \Theta(\varepsilon - \|\mathbf{x}_i - \mathbf{x}_j\|_2) \tag{1}$$

where $\Theta(.)$ is the Heaviside function. Once a suitable threshold ε is determined, the RP is then a binary image displaying a black dot where the values are within the *threshold corridor*. As we shall see, the proposed self-similarity matrix is a variant of the RP: instead of capturing the system's behavior using *dots* and *lines* by thresholding, we aim for plots with richer textures, in terms of distinct peaks and valleys, which are hopefully distinctive for different dynamical systems. These patterns on the RPs, and *a fortiori* on SSMs, contain a wealth of information about the dynamics of a system and capture specific behaviors of the system. Researchers have attempted to classify dynamic systems into different categories based on these *textures*. Part of this categorization [41] is reproduced in Table I.

(a) (b) (c) (d) (e)

Fig. 3. **Earlier example of SSM for motion periodicity analysis.** (a)-(d) are frames from a sequence of a walking person [38]. (e) represents the SSM obtained for this sequence by [38] using the absolute correlation score between frames of the sequence. Time-Frequency analysis is performed on this matrix to detect periodicity in a motion sequence.

McGuire *et al.* [42] have shown that RPs not only preserve invariants of a dynamical system (such as the Lyapunov exponents [43]) but are also to some extent independent of the embedding dimension [42], which naturally raises the question of whether embedding is necessary at all for understanding the underlying dynamics of a system [44]. In addition, [42] have shown that RPs for different systems are identical as long as the transformation is *isometric*. This conclusion also apply to proposed SSMs. As we shall see, SSMs are not strictly invariant under *projective* or *affine* transformations, but are experimentally found stable under 3D view changes.

III. Self-Similarity Matrix (SSM)

Self-similarity matrices have already appeared in the past under various specific forms, including binary recurrence plots associated to time series, as mentioned above. In this section we define such matrices for different image features, with examples for several action classes, and start investigating their stability across views.

For a sequence of images $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_T\}$ in discrete (x, y, t)-space, a SSM of $\mathcal{I}$ is a square symmetric matrix of size

TABLE I
TYPICAL PATTERNS OF RECURRENCE PLOTS AND THEIR MEANING (REPRODUCED FROM [41])

	SSM Pattern	Meaning
(1)	Homogeneity	The process is stationary
(2)	Fading in the corners	Non-stationary data; the process contains a trend or a drift
(3)	Periodic/quasi-periodic patterns	Cyclicities in the process; the time distance between periodic patterns (e.g. lines) corresponds to the period
(4)	Single isolated points (or structures)	Strong fluctuation in the process; if only single isolated points occur, the process may be an uncorrelated random or even anti-correlated process
(5)	Diagonal lines (parallel to the main diagonal)	The evolution of states is similar at different epochs; the process could be deterministic; if these diagonal lines occur beside single isolated points, the process could be chaotic (if these diagonal lines are periodic, unstable periodic orbits can be observed)
(6)	Diagonal lines (orthogonal to the main diagonal)	The evolution of states is similar at different times but with reverse time; sometimes this is an indication for an insufficient embedding
(7)	Long bowed line structures	The evolution of states is similar at different epochs but with different velocity; the dynamics of the system could be changing

$T \times T$,

$$\left[d_{ij}\right]_{i,j=1,2,\ldots,T} = \begin{bmatrix} 0 & d_{12} & d_{13} & \ldots & d_{1T} \\ d_{21} & 0 & d_{23} & \ldots & d_{2T} \\ \vdots & \vdots & \vdots & & \vdots \\ d_{T1} & d_{T2} & d_{T3} & \ldots & 0 \end{bmatrix} \quad (2)$$

where d_{ij} is the distance between certain low-level features extracted in frames $\mathcal{I}_i$ and $\mathcal{I}_j$ respectively. The diagonal corresponds to comparing a frame to itself (no dissimilarity), hence is composed of zeros. The exact structures or the patterns of this matrix depend on the features and the distance measure used for computing the entries d_{ij}. For example, after tracking walking people in a video sequence, [37] and [38] compute a particular instance of SSM where d_{ij} is the absolute correlation between two frames, as depicted in Fig. 3. The computed matrix patterns (cf. Fig. 3(e)) have a significant meaning for their application - the diagonals in the matrix indicate periodicity of the motion.

In this work, we define d_{ij} as the Euclidean distance between the different features that we extract from an action sequence. This form of SSM is known in the literature as the Euclidean Distance Matrix (EDM) [45].

To get a fist insight into the representation power of SSMs, a comparison with the notion of "dynamic instances" proposed by Rao *et al*. [26] is illustrated in Fig. 2. The authors of [26] argue that continuities and discontinuities in position, velocity and acceleration of a 3D trajectory of an object are preserved under 2D projections. For an action of opening a cabinet door, performed by two different actors from considerably different viewpoints, these points are depicted in Fig. 2. Fig. 2(c)(e) shows the SSMs computed for these two actions based only on one hand trajectory, where red color indicates higher values and dark blue color indicates lower values. The dynamic instances, red stars in Fig. 2(b)(d), correspond to valleys of different area/spread in our plot of SSM (cf. Fig. 2(c)(e)), marked by magenta circles along the diagonal of the matrix. The exact spread of these valleys depend on the width of the peaks in the spatio-temporal curvature of the actions, as shown in Fig. 2(b)(d). However, whereas [26] capture only the local discontinuities in the spatio-temporal curvature, the SSM captures more information about other dynamics of the actions present in the off-diagonal parts of the matrix. Note also that the proposed notion of self-similarity, unlike [5] or [26], does not require estimation of point correspondences or time-alignment between different actions.

A. Trajectory-based Self-Similarities

If a set of M points $\mathbf{x}^m,\ m = 1 \cdots M$, distributed over a person is "tracked" (in sense to be specified later) over the duration of an action performance, the mean Euclidean distance between each of the k pairs of corresponding points at any two instants i and j of the sequence can be computed as

$$d_{ij} = \frac{1}{M} \sum_{m=1}^{M} \|\mathbf{x}_i^m - \mathbf{x}_j^m\|_2 \quad (3)$$

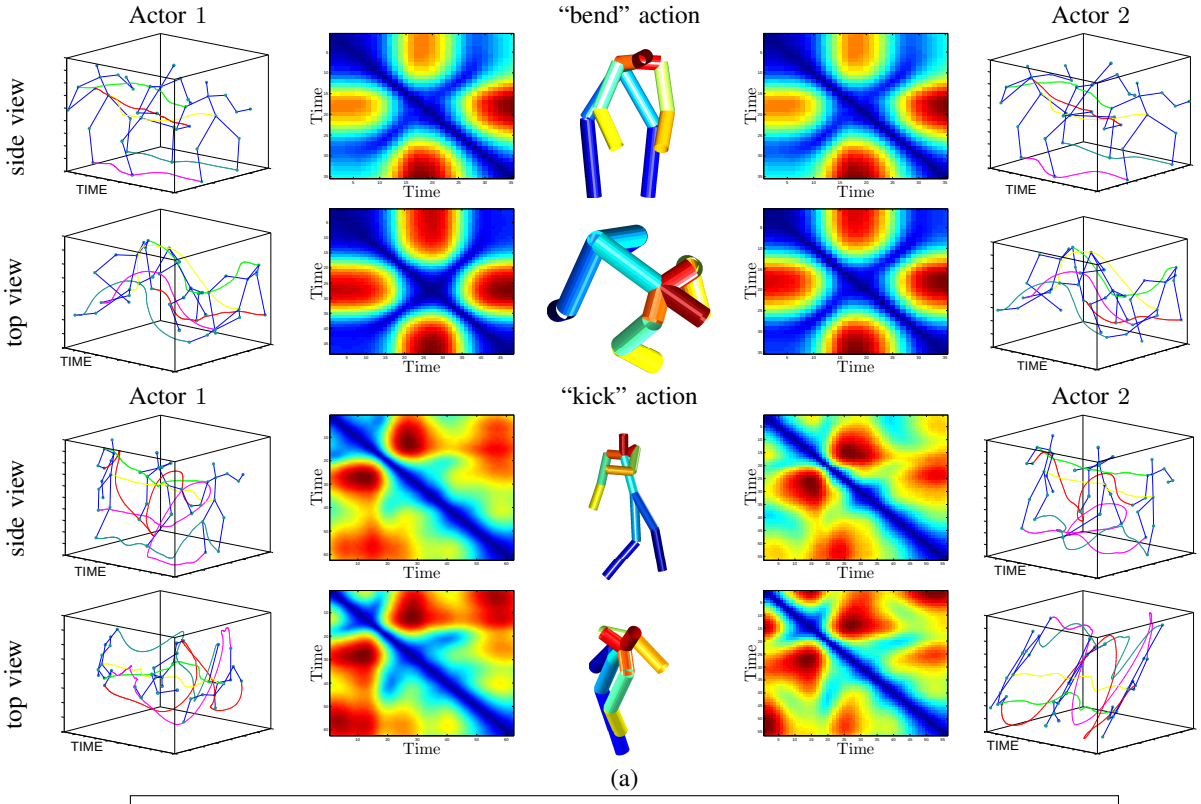
where $\mathbf{x}_i^k$ and $\mathbf{x}_j^k$ indicate positions of points on the track k at time instants i and j. We denote the self-similarity matrix computed from (3) by SSM-pos.

In a first set of experiments aimed at investigating SSM properties in a controlled set-up, such point trajectories are directly obtained via motion capture, rather than from video sequences. In this case, a "view" corresponds to the projection of 3D point tracks onto a given 2D plane. In these experiments, we track $M = 13$ joints on a person performing different actions [16], as shown in the Fig. 4(a). In order to remove the effect of global person translation, without loss of generality, the points are centered to their centroid so that their first moment is zero.

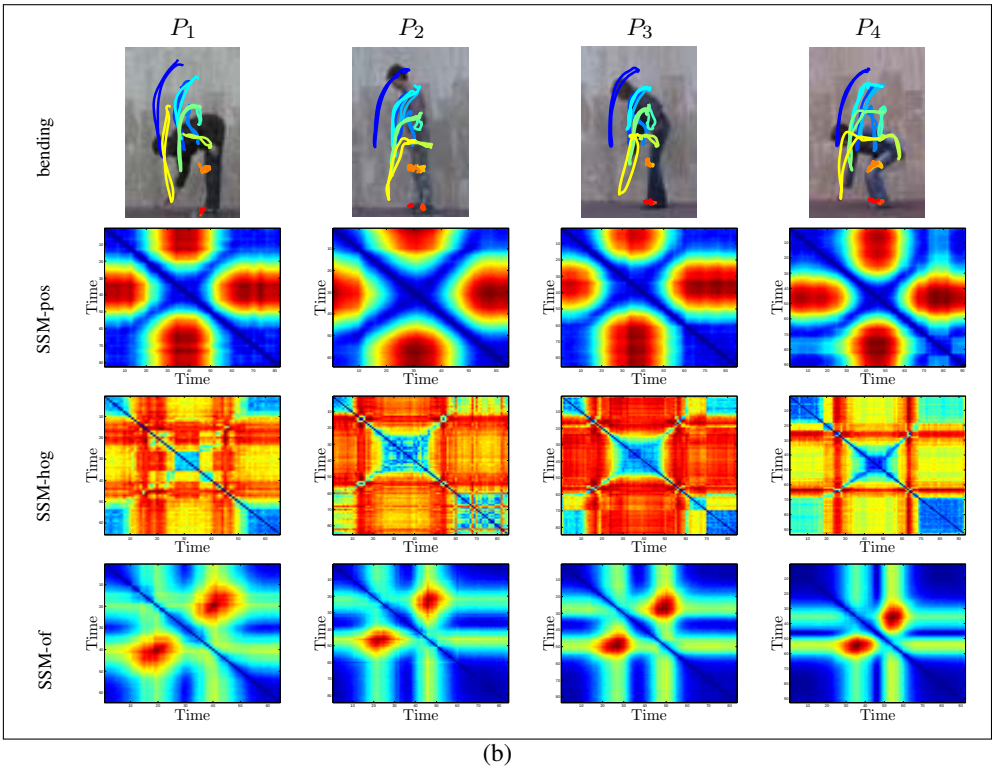
The overall goal of proposed work being the recognition of actions in videos irrespective to view points, the actual computation of SSM-pos requires that points are extracted and tracked in the input video. We assume that this task is handled automatically by an external module such as KLT [46] point tracker. Note that our method is not restricted to any particular subset of points as far as the points are distributed over moving body parts. The definition of SSM-pos in 3 needs however to be adapted to a set of tracks with arbitrary length and starting time:

$$d_{ij} = \frac{1}{|S_{ij}|} \sum_{m \in S_{ij}} \|\mathbf{x}_i^m - \mathbf{x}_j^m\|_2, \quad (4)$$

where $S_{ij} \subset \{1, \cdots, M\}$ is the set with indices of point trajectories that are alive between frames i and j.



(a)



(b)

Fig. 4. **Examples of SSMs for different types of features and for different actions**. (a) Examples from CMU mocap dataset. Columns 1 and 5 represent two actors while columns 2 and 4 represent corresponding SSM-pos computed with 13 projected point trajectories, respectively. Different rows represent different actions and viewing angles. Note the stability of SSMs over different views and persons performing the same action. (b) Examples from Weizman video dataset [7]. Row 1: four bending actions along with manually extracted point trajectories used for computing SSM-pos; Rows 2, 3, 4 represent SSM-pos, SSM-hog and SSM-of respectively for these four bending actions. Note the similarity column-wise.

In addition to the SSM-pos, we also compute similarities based on the first and the second-order derivatives of the 2D positions, i.e., velocities and accelerations. Similarities computed base on these features are denoted by SSM-vel and SSM-acc, respectively.

B. Image-based Self-Similarities

Beside point trajectories, alternative image features can be used to construct other SSMs for the same image sequence. To describe spatial appearance of a person at each image frame, we compute Histograms of Oriented Gradients (HoG) features [47]. This descriptor, originally used to perform human detection, characterizes the local shape by capturing the gradient structure. In our implementation, we use 4 bin histograms for each of 5×7 blocks defined on a bounding box around the person in each frame. Feature distance d_{ij} between time instants i and j is then computed as the Euclidean distance between two HoG vectors extracted from frames $\mathcal{I}_i$ and $\mathcal{I}_j$. We denote SSMs computed using HoG features by SSM-hog.

In addition to HoG features, we also test the proposed method by considering optical flow vectors as another input feature. The corresponding SSMs are denoted by SSM-of. More precisely, we assume, as for point trajectories, that optical flow is provided by another module, e.g., Lucas and Kanade algorithm [48] based on two consecutive frames. We concatenate the components of optical flow vectors computed for all n pixels in a bounding box around a person into a flow vector of size $2n$. Entry d_{ij} of SSM-of matrix then amounts to the Euclidean distance between the flow vectors corresponding to the two frames $\mathcal{I}_i$ and $\mathcal{I}_j$. In practice, we enlarge and resize bounding boxes in order to avoid border effects on the flow computation and to ensure the same size of the flow vectors along an action sequence. We resize the height to a value equal to 150 pixels and the width is set to the greatest value for the considered sequence.

Examples of SSMs computed for different image features are shown in Fig. 4. Fig. 4(a) contains example actions from the CMU motion capture (mocap) dataset projected onto different views. Column 1 and 5 of Fig. 4(a) represent two different actors while columns 2 and 4 represent their computed SSM-pos, respectively. The first two rows represent a bending action performed by two actors and projected onto two considerably different views. The last two rows, similarly, represent a football kick action for two actors and two different views. Note the similarity of SSMs computed for actions of the same class despite the changes of the actor and the considerable changes of views. Note also the visual difference of SSMs between two action classes. Computing SSMs on real image features instead of mocap data leads to similar conclusions. Fig. 4(b) illustrates SSMs obtained for the bending action from the video dataset [7]. Row 2 shows SSM-pos computed using point tracks overlaid on images in first row. Rows 3 and 4 show SSM-hog and SSM-of for the same sequences respectively. For a given type of features, note the similarity of SSMs over the different instances of the same action class. SSMs for different feature types do not look similar since different features capture different properties of the action. This suggests the use of SSMs computed for different features in a complementary manner.

C. Structural Stability of SSM across Views

As noted above, the patterns of proposed SSMs are promisingly stable through changes of viewpoints. In order to assess more thoroughly this stability we conducted the following experiments using the CMU mocap dataset. We deployed a total of $K = 684$ synthetic affine cameras (at distinct latitudes and longitudes) on a sphere surrounding the person performing an action, as shown in Fig. 5(a). For each of these cameras, we compute the SSM matrix, as described in Section III-A, and aim to assess qualitatively and quantitatively the stability of the patterns contained in these SSMs. To this end, we consider SSMs as being discrete "images" of size $T \times T$ which allows us to resort to classic tools for image structure analysis. We consider in particular orientation of bi-dimensional gradient as it is known to capture image structures independently of various changes, including changes in the dynamics of intensity levels. We will further rely on this philosophy when building SSM descriptors in next section. For the time being we consider a simpler structure analysis based on so-called circular statistics [49].

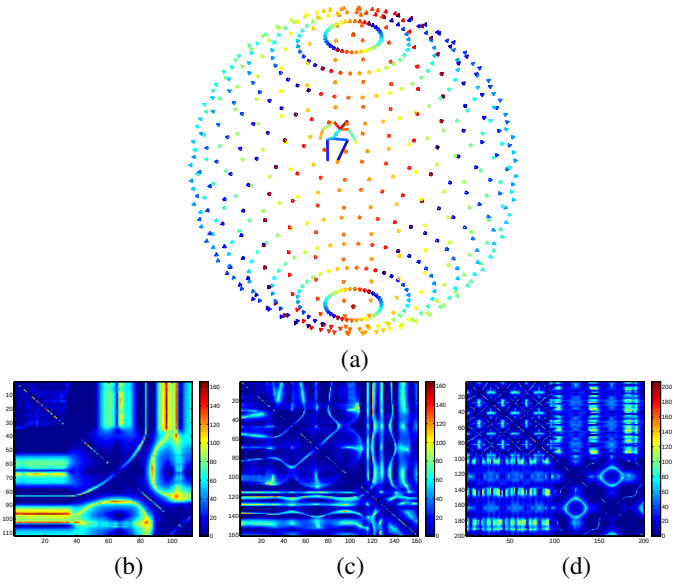


Fig. 5. **Stability of SSM-pos structures across viewpoints for mocap data sequence** (a) Synthetic cameras around a person performing an action. Self-Similarity matrices (SSMs) are generated for each of these synthetic cameras and for each of these computed SSMs, a gradient angle is computed at each matrix point. From these orientations, circular standard deviations are computed for a *golf swing* (b), *kick* (c) and a *jumping jack* (d) action sequence (code provide by [27] for (a)).

At each "pixel" (i, j) of the SSM associated to k-th view, we compute the orientation $\theta_{ij}^{(k)}$ of the bi-dimensional gradient vector. In order to ascertain the effect of different viewing directions on the computed SSMs, we then compute at this point the circular mean and standard deviation, $\overline{\theta}_{ij}$ and $\overline{\sigma}_{ij}$ of the orientation over the $K = 684$ SSMs. Let $\overline{\mathbf{r}}_{ij} = \begin{bmatrix}\overline{c}_{ij} & \overline{s}_{ij}\end{bmatrix}'$, where

$$\overline{c}_{ij} = \Sigma_{k=1}^{K} \cos\theta_{ij}^{(k)}/K \quad \overline{s}_{ij} = \Sigma_{k=1}^{K} \sin\theta_{ij}^{(k)}/K. \tag{5}$$

Then the circular mean direction is given as:

$$\overline{\theta}_{ij} = \begin{cases} \arctan(\overline{s_{ij}}/\overline{c_{ij}}) & \text{if } \overline{c_{ij}} \geq 0 \\ \arctan(\overline{s_{ij}}/\overline{c_{ij}}) + \pi \text{sign}(\overline{s_{ij}}) & \text{if } \overline{c_{ij}} < 0 \end{cases}$$

The mean resultant length, $\overline{r}_{ij} = \sqrt{\overline{c}_{ij}^2 + \overline{s}_{ij}^2}$, is used to compute the *circular standard deviation* as $\overline{\sigma}_{ij} = \sqrt{-2\ln \overline{r}_{ij}}$. We computed $[\overline{\sigma}_{ij}]_{i,j=1\cdots T}$ for some sample action sequences from *golf swing, kick* and *jumping jack* action classes, as shown in Fig.

5(b)(c)(d), respectively. One can notice that, for each action, the standard deviations are low over most parts of the SSM support, which is a good indicator of SSM structure stability across views. Highest values delineate what can be seen as the strong contours of the average SSM structure for concerned action.

IV. SSM-BASED ACTION DESCRIPTION AND ALIGNMENT

As discussed in the previous section, SSMs have view-stable and action-specific structure. Here we aim to capture this structure and to construct SSM-based descriptors for subsequent view independent action analysis such as alignment and recognition. We note the following properties of SSM: (i) absolute values of SSM may depend on the varying properties of the data such as the projected size of a person in the case of SSM-pos; (ii) fluctuations in the individual performances of a type of actions and temporal de-synchronization of the views may effect the global structure of SSM; (iii) the uncertainty of values in SSM increases with the distance from the diagonal due to the increasing difficulty of measuring self-similarity over long time intervals. These properties led us to the SSM description that follows.

As already mentioned in previous section, we avoid dependency to varying absolute SSM values by resorting to gradient orientations computed from neighbouring elements of the matrix seen as an image. We also avoid global descriptors and, in a manner reminiscent to popular local image descriptors used for object detection and recognition, we accumulate histograms of gradient orientations in local patches. These patches however are only centered on the diagonal of SSM. Our patch descriptor has a log-polar block structure as illustrated in Fig. 6. The diameter of the circular regions under consideration should be seen as temporal window extent. For log-polar block a at time i, we compute the normalized 8-bin histogram $\mathbf{h}_i^a = [h_{i,b}^a]'_{b=1:8}$ of SSM gradient orientations within the block. We then concatenate the histograms of the 11 blocks of the analysis support into a descriptor vector $\mathbf{h}_i = [\mathbf{h}_i^{a\prime}]'_{a=1:11}$. For descriptors at boundaries with blocks falling outside SSM we set $\mathbf{h}_i^a$ to a zero vector.

Choosing a temporal extent of the descriptor involves a trade-off between the amount of captured temporal information and its variability, which is delicate to tune. In addition, using a single descriptor size may be suboptimal when representing events of varying lengths and with irregularly changing speed. We address this issue by considering local SSM descriptors of multiple sizes and demonstrate the impact of this approach on action recognition in Section V-C.

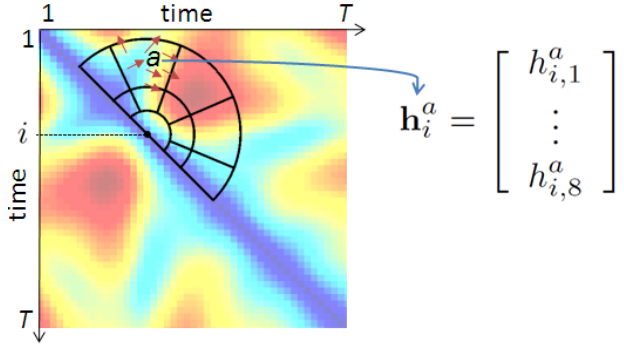


Fig. 6. **Local descriptors for SSM**: each individual descriptor is centered at a diagonal point $i \in \{1 \cdots T\}$ of the SSM and has a log-polar block structure. Histograms $\mathbf{h}_i^a$ of 8 gradient directions are computed separately for each of the 11 blocks of the analysis support and are concatenated into a descriptor vector $\mathbf{h}_i$.

When constructing a joint local descriptor for multiple SSMs computed for F different features, we concatenate F corresponding local descriptors $\mathbf{h}_i^f$ from each SSM into a single vector $\mathbf{h}_i = [\mathbf{h}_i^{f\prime}]'_{f=1:F}$. In such a way we obtain for instance SSM-hog-of descriptors by concatenating image-based SSM-hog and SSM-of descriptors. When temporal ordering is required, the representation for a video sequence can finally be defined by the sequence of local descriptors $H(\mathcal{I}) = (\mathbf{h}_1, ..., \mathbf{h}_T)$ computed over all diagonal elements of SSMs associated to all feature types.

Temporal cross-view action synchronization: Before moving to action recognition based on representation previously defined, we first test this representation on the problem of temporal alignment, or synchronization, of video sequences representing the *same* action from different viewpoints. The problem amounts to finding the monotonic mapping between the time-line of the first sequence and the time-line of the second one. Consider for instance two videos $\mathcal{I}^1$ and $\mathcal{I}^2$ recorded simultaneously for the side and the top views of a person in action, as shown in Fig. 7(a). To further challenge the alignment, we apply a nonlinear time transformation to one of the sequences. To solve the alignment, we (i) compute optical flow based SSM-of for both image sequences, (ii) represent both videos by a sequence of local SSM descriptors, $H(\mathcal{I}^1)$ and $H(\mathcal{I}^2)$ respectively, computed for a single temporal scale as described above and (iii) align the two descriptor sequences using Dynamic Programming (DP). The estimated time transformation is illustrated by the red curve in Fig. 7(b) which closely follows the ground truth transformation (blue curve) despite the drastic change of viewpoint between sequences.

Using the same method, we next address alignment of *different* instances of similar actions. Fig. 8 demonstrates alignment of pairs of videos representing actions *throwing a ball, drinking* and *smoking* performed by different people in varying views. The automatically estimated alignment recovers the manual alignment at key-frames as illustrated in Fig. 8(right) despite large variations in appearance and viewpoints across videos. Note, that in all alignment experiments we have used known person bounding boxes for computing SSM-of descriptors.

The successful alignment of actions illustrated above indicates the strength of SSM-based descriptors and their ability to cope with video variations in terms of viewpoints, subject appearance and movement speed. This suggests that SSM-based descriptors can be used for action recognition as will be investigated in the next section.

V. SSM-BASED ACTION RECOGNITION

In this section we evaluate SSM-based video descriptors for the task of view-invariant action recognition. To recognize action sequences we follow recently successful bag-of-features (BoF) approaches [12], [50], [51] and represent each video as a set of quantized local SSM descriptors with their temporal positioning in the sequence being discarded. Taking this view that global temporal ordering is not taken into action (as opposed to its use for synchronization where it is a crucial information) permits to filter out fluctuations between actions from the same class while retaining sufficient action discrimination to build a view-independent action recognition system, as demonstrated below.

Fig. 7. **Temporal alignment of same action performances in videos from different viewpoints and with synthetic de-synchronization**. (a): Two de-synchronized sequences with the side and the top views of the same action are represented with a set of matching key-frames. The second sequence has been time warped according to $t' = a\cos(bt)$ transformation. (b) Distance matrix between sequences $H(\mathcal{I}^1)$ and $H(\mathcal{I}^2)$ of SSM-pos descriptors (bright colors represent large distance values). Dynamic Programming (red curve) finds the minimum cost monotonic path from $(0, 0)$ to (T, T) in this matrix. This path coincides almost perfectly with the original warping (blue curve) despite drastic view variations.

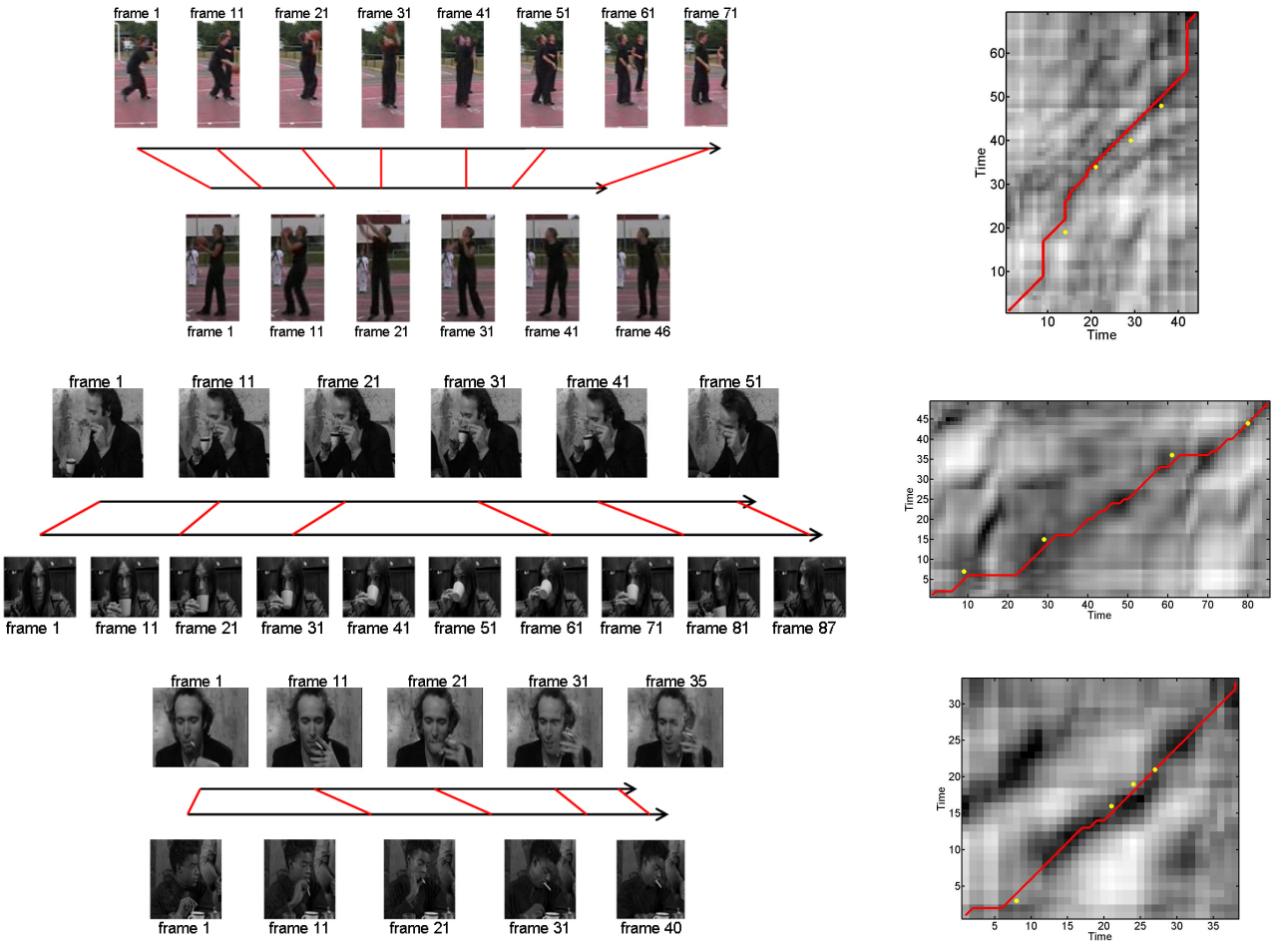
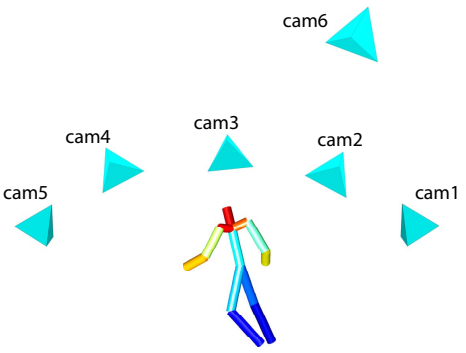


Fig. 8. **Temporal alignment of video sequences representing different performances of actions *throwing a ball, drinking* and *smoking***. (Left): Pairs of aligned video sequences are illustrated with a few frames and the links between corresponding frames estimated by our algorithm. (Right): Distance matrices between sequential descriptors of both videos used as input for aligning video sequences by Dynamic Programming. The estimated temporal alignment is illustrated by red curves. The successful alignment achieved by our method on these sequences is confirmed when comparing red curves with yellow dots illustrating sparse manual alignment for a few key-frames of videos (best viewed in color).

As in classic BoF approach, local SSM descriptors are quantized based on a visual "vocabulary" learned off-line: by k-means clustering of 10,000 random local SSM descriptors from training sequences, 1000 clusters are defined, with their centers being the words of this vocabulary. In subsequent classifier training and testing, each feature is then assigned to the closest (we use Euclidean distance) vocabulary word. This way, each image sequence $\mathcal{I}$ is now described by a normalized histogram $\mathcal{H}(\mathcal{I})$ of visual words. These histograms are the input data used for recognition.



(a)

training views \ test	cam1	cam2	cam3	cam4	cam5	cam6	All
cam1	**92.7**	93.3	86.0	84.8	87.8	86.0	88.4
cam2	92.1	**92.7**	89.0	86.0	86.6	90.2	89.4
cam3	86.0	87.2	**95.7**	93.9	82.9	79.9	87.6
cam4	82.3	84.8	93.3	**95.7**	87.2	79.3	87.1
cam5	85.4	86.0	84.8	87.8	**91.5**	87.8	87.2
cam6	89.6	89.6	84.1	86.6	88.4	**93.9**	88.7
All	93.9	91.5	95.7	96.3	93.3	92.1	**93.8**

(b)

Fig. 9. **SSM-based cross-view action recognition on CMU mocap data**. (a) A person figure animated from the motion capture data and six virtual cameras used to simulate projections in our experiments. (b) Accuracy of the cross-view action recognition using SSM-pos-vel-acc descriptors to build the bag of features used by nearest-neighbor classifier.

In the following we consider two different types of classifiers: the Nearest Neighbour Classifier (NNC) and a Support Vector Machine (SVM) classifier. In the case of NNC, we simply assign to test sequence $\mathcal{H}(\mathcal{I})$ the action label of the training sequence $\mathcal{I}^*$ which minimizes distance $D_{NN}(\mathcal{H}(\mathcal{I}), \mathcal{H}(\mathcal{I}^*))$ over all training sequences. The distance D_{NN} is defined by the greedy matching of local descriptors described in [51]. We apply NNC only to datasets with a limited number of samples. For SVM classification, we train non-linear SVMs using χ^2 kernel and adopt one-against-all approach for multi-class classification.

We evaluate SSM-based action recognition on three public datasets. For all recognition experiments we report results for n-fold cross-validation and make sure the actions of the same person do not appear in the training and in the test sets simultaneously. In Section V-A we validate the approach in controlled multi-view settings using motion capture data. In Section V-B we demonstrate and compare the discriminative power of our method on a standard single-view action dataset [7]. We finally evaluate the performance of the method on a comprehensive multi-view action dataset [35] in Section V-C. We demonstrate the advantage of combining SSM descriptors computed for different types of image features and multiple temporal scales. Multi-view recognition results are compared with results of other methods on the same datasets.

A. Experiments with CMU MoCap dataset

To simulate multiple and controlled view settings we have used 3D motion capture data from CMU dataset (*http://mocap.cs.cmu.edu*). Trajectories of 13 points on the human body were projected to six cameras with pre-defined orientations with respect to the human body (see Fig. 9(a)). We have used 164 sequences in total corresponding to 12 action classes *(bend, cartwheel, drink, fjump, flystroke, golf, jjack, jump, kick, run, walk, walkturn)*. To simulate potential failures of the visual tracker, we distracted trajectories by randomly breaking them into parts with the average length of 2 seconds. Fig. 9(b) demonstrates results of NNC action recognition when training and testing on different views using SSM-pos, SSM-vel and SSM-acc. As observed from the diagonal, the recognition accuracy is the highest when training and testing on the same views while the best accuracy (95.7%) is achieved for cam5 (frontal view). Interestingly, the recognition accuracy degrades only moderately with substantial view changes and remains still high across top view (camera 6) and side views (camera 1 to 5). When training and testing on all views, the average accuracy is 90.5%.

B. Experiments with Weizman actions dataset

To asses the discriminative power of our method on real video sequences, we apply it to a standard single-view video dataset with nine classes of human actions performed by nine subjects [7](see Fig. 10(top)). On this dataset we compute NNC recognition accuracy when using either image-based or trajectory-based self-similarity descriptors according to Section III. Given the low resolution of image sequences in this dataset, the trajectories were acquired by [16] via semi-automatic tracking of body joints. Recognition accuracy achieved by our method for optical flow-based and trajectory-based self-similarities is 94.6% and 95.3% respectively and the corresponding confusion matrices are illustrated in Fig. 10(a)-(b). The recognition results are high for both types of self-similarity descriptors and outperforms the accuracy of 92.6% achieved by a recent trajectory-based method [16]. Whereas higher recognition rates on this *single-view* dataset have been reported, e.g., in [52], the main strength of our method will be demonstrated for action recognition across *multiple views*, as described in the next section.

C. Experiments with IXMAS dataset

IXMAS dataset is publicly available and numerous researchers have reported their results on this dataset. Without resorting to engineering a different experimental setup to test view invariance, using this dataset allows for a quick and a fair comparison of our method to the other methods. Thus we present results for IXMAS video dataset [35] with 11 classes of actions performed three times by each of 10 actors and recorded simultaneously from 5 different views. Sample frames for all cameras and four action classes are illustrated in Fig. 11. Here we use SVM classifier in combination with image-based self-similarity descriptors in terms

	bend	jack	jump	pjump	run	side	skip	walk	wave
bend	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	**77.8**	0.0	0.0	0.0	11.1	11.1	0.0
pjump	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0
skip	0.0	0.0	20.0	0.0	10.0	0.0	**70.0**	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
wave	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**

(a)

	bend	jack	jump	pjump	run	side	skip	walk	wave
bend	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	**66.7**	11.1	0.0	11.1	11.1	0.0	0.0
pjump	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0	0.0	0.0
skip	0.0	0.0	12.5	0.0	0.0	0.0	**87.5**	0.0	0.0
walk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
wave	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**

(b)

Fig. 10. **SSM-based action recognition on Weizman single-view action dataset [7].** (Top) Example frames for nine classes of actions. (Bottom) Confusion matrices corresponding to NNC action recognition using image-based self-similarities SSM-of (a) and trajectory-based self-similarities SSM-pos (b).

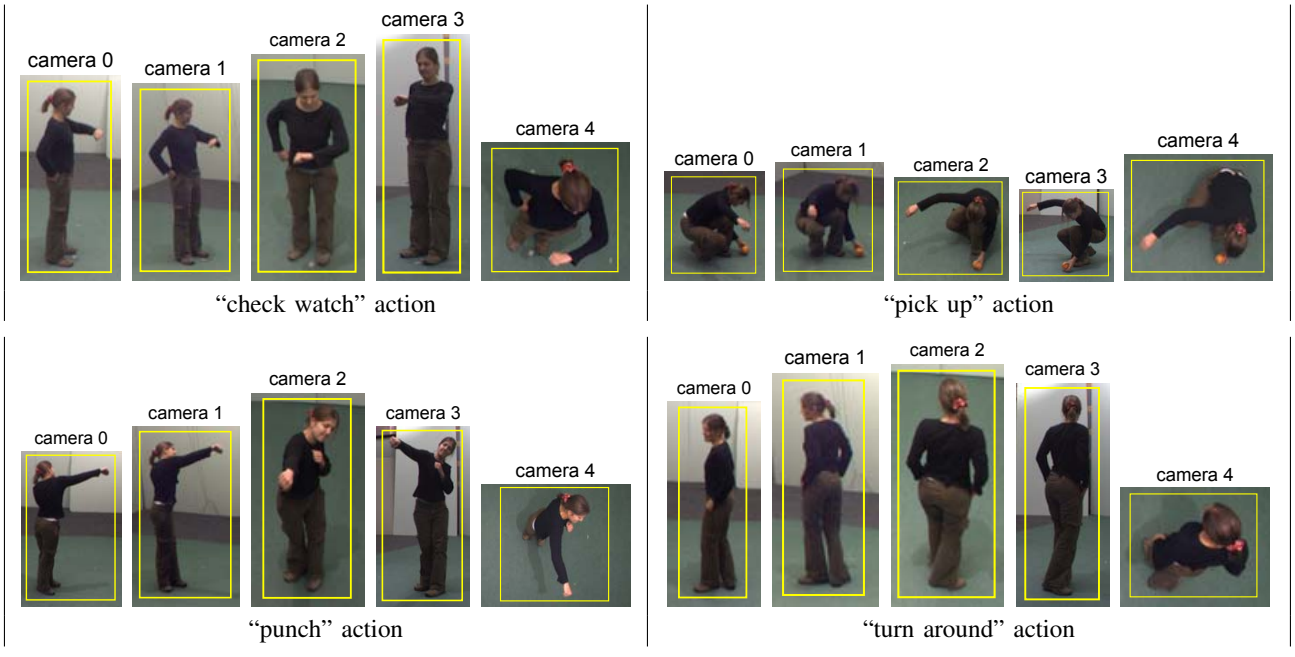


Fig. 11. **Example frames from IXMAS multiview action dataset**: for four classes of action, the five views at a given instant, of one performance of the action is shown.

of SSM-hog, SSM-of and their combination SSM-hog-of. We also consider local SSM descriptors computed at *multiple temporal scales*. For each SSM diagonal point, three local descriptors are computed corresponding to three different diameters for the log-polar domain (respectively 28, 42 and 56 frames in diameter). The number of descriptors assigned to a given sequence is thus multiplied accordingly. All descriptors are quantized independently of their scale using a single visual vocabulary and are used to compute a single histogram associated to the sequence.

Fig. 12(a-c) illustrate recognition accuracy of cross-views action recognition for different combinations of training and test cameras and for different types of SSMs. The results are averaged over all classes and test subjects. Similar to results on CMU dataset in Section V-A, here we observe high stability

	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	77.0	75.2	69.7	71.8	49.4	68.6
Train Cam1	78.5	77.3	67.9	71.5	48.0	68.6
Train Cam2	70.0	73.0	75.8	68.5	55.2	68.5
Train Cam3	73.6	72.4	67.3	71.2	45.9	66.1
Train Cam4	44.5	41.5	55.2	37.9	68.8	49.6
Train All	77.0	78.8	80.0	73.9	63.3	74.6

cross−camera training/testing same camera training/testing

(a): *Recognition results for SSM-hog-of multi-scale features*

	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	66.4	73.3	63.9	60.3	41.5	61.1
Train Cam1	67.3	70.6	62.1	62.4	41.5	60.8
Train Cam2	63.9	65.8	71.8	57.6	52.1	62.2
Train Cam3	62.1	68.2	62.4	61.2	35.9	58.0
Train Cam4	34.8	36.4	55.8	33.3	63.0	44.7
Train All	67.9	73.6	70.6	66.4	59.1	67.5

cross−camera training/testing same camera training/testing

(b): *Recognition results for SSM-of multi-scale features*

	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	67.6	66.1	54.8	57.3	44.4	58.0
Train Cam1	73.6	63.6	57.9	59.5	45.3	60.0
Train Cam2	58.2	54.5	63.3	54.2	49.4	55.9
Train Cam3	60.0	58.2	55.8	60.6	42.1	55.3
Train Cam4	46.7	44.2	51.5	43.9	60.0	49.3
Train All	69.7	63.3	64.8	62.7	52.7	62.7

cross−camera training/testing same camera training/testing

(c): *Recognition results for SSM-hog multi-scale features*

	Test Cam0	Test Cam1	Test Cam2	Test Cam3	Test Cam4	Test All
Train Cam0	80.0	75.9	42.3	55.6	21.8	55.6
Train Cam1	74.8	83.9	36.5	58.3	23.6	56.0
Train Cam2	43.6	46.1	80.5	64.7	34.2	53.7
Train Cam3	47.0	50.0	45.8	85.5	18.8	49.5
Train Cam4	19.7	19.4	43.5	26.1	73.3	36.0
Train All	80.3	84.5	79.4	84.8	68.5	79.6

cross−camera training/testing same camera training/testing

(d): *Recognition results for STIP-hog-hof multi-scale features*

Fig. 12. **Comparative action recognition results for IXMAS multiview action dataset**: results are averaged over 11 action classes and 10 subjects. Results in (a)-(c) are shown for different types of SSMs and the same bag-of-features SVM classification method. Results in (d) are obtained with the same bag-of-feature SVM approach, but using quantized descriptors of spatio-temporal interest points (STIP) instead of quantized local SSM descriptors. Recognition scores are illustrated for different combination of training and test cameras.

	check−watch	cross−arms	get−up	kick	pick−up	punch	scratch−head	sit−down	turn−around	walk	wave
check−watch	**73.7**	8.1	0.0	1.1	1.5	1.7	11.3	0.5	0.1	0.0	1.9
cross−arms	3.8	**72.6**	1.0	1.8	0.4	0.2	15.7	0.6	0.3	0.0	3.6
get−up	0.5	0.6	**72.8**	3.6	4.1	1.4	0.2	8.8	7.7	0.4	0.0
kick	1.7	1.3	3.9	**57.7**	1.0	15.4	0.9	1.4	14.4	0.9	1.4
pick−up	0.9	0.1	1.7	0.4	**84.5**	1.9	0.7	6.5	1.1	2.0	0.3
punch	3.3	1.0	0.9	15.1	2.2	**70.5**	0.0	1.4	2.6	0.0	3.0
scratch−head	13.5	11.9	1.2	0.6	0.4	0.8	**61.1**	0.3	0.1	0.7	9.3
sit−down	0.6	0.1	9.6	1.1	2.3	1.2	0.1	**81.1**	3.5	0.2	0.0
turn−around	0.0	0.1	4.0	5.2	0.8	0.8	0.0	1.8	**73.2**	14.0	0.2
walk	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
wave	3.3	1.6	0.6	1.4	0.1	1.6	8.2	0.0	1.1	0.0	**82.1**

Fig. 13. **Class-confusion matrix for action recognition in IXMAS dataset**: this confusion matrix is obtained using SSM-hog-of multi-scale SSM local descriptors. It corresponds to the average confusion computed for all *cross-camera* recognition setups in Fig. 12(a).

of action recognition over view changes, now using visual data only. The method achieves reasonable recognition accuracy even for extreme variations in views such as for testing on top views (Test Cam4) when using side-views only for training. Also, these tables indicate that using jointly HoG-based and optical flow-based SSMs yields better recognition than using either of the two types of feature individually. The class confusion matrix in Fig. 13, computed using SSM-hog-of, illustrates good per-class recognition performance for all classes when averaged over all cross camera setups in Fig. 12(a), i.e., using camera-X for training and camera-Y for testing for $X \neq Y$.

Comparison to alternative methods: We compare recognition performance of SSM-based features to space-time interest points (STIPs) [9], [15] representing videos by sets of descriptors computed from local space-time patches. STIP descriptors have been recently demonstrated to achieve competitive performance on several action recognition benchmarks [53]. STIP features, however, are not designed to handle large view variations. The recognition performance of STIP features on IXMAS dataset using the same classification method as for SSM-based features is illustrated in Fig. 12(d). It is interesting to observe that STIP features outperform SSM-based features in recognition setups where the same or similar views are used for training and testing. For large variation between training and test views, however, SSM-based descriptors considerably outperform STIP features, especially when testing on top views after learning on side views, and vice-versa. This behavior is consistent with the intuition that SSM-based descriptors gain view independence at the cost of somewhat reduced discriminative power. The comparison of SSM-based and STIP features is summarized in Table II for different recognition setups.

We also compare our approach to the two alternative methods

	cross camera	same camera	any-to-any
SSM-hog-of	**61.8**	74.0	**64.3**
SSM-of	55.0	66.6	57.4
SSM-hog	53.9	63.0	55.7
STIP-hog-hof	42.4	**80.6**	50.0
Farhadi [33]	58.1	68.8	60.3
Weinland [35]	—	57.9	—

TABLE II
COMPARISON OF RECOGNITION RESULTS ON IXMAS DATASET BY ALTERNATIVE METHODS. RESULTS ARE PRESENTED FOR DIFFERENT COMBINATIONS OF TRAINING CAMERA-X AND TEST CAMERA-Y SETUPS WHERE "CROSS CAMERA" INDICATES SETUPS WITH $X \neq Y$, "SAME CAMERA" INDICATES SETUPS WITH $X = Y$ AND "ANY-TO-ANY" INDICATES ALL COMBINATIONS OF X AND Y.

	cross camera	same camera	any-to-any
multi-scale	**61.8**	**74.0**	**64.3**
56 frames	59.9	70.9	61.8
42 frames	59.3	69.4	61.6
28 frames	54.0	65.3	56.2

TABLE III
COMPARISON OF THE IMPACT OF SSM DESCRIPTOR SIZE ON THE RECOGNITION PERFORMANCE IN IXMAS DATASET.

in the literature that were evaluated on the same dataset. Action recognition in IXMAS dataset is addressed by means of 3D reconstruction in [35]. Results of this method reported for the same training/test camera setup are lower compared to our SSM-based recognition scheme as illustrated in Table II. Our SSM-based descriptors also outperform results of the transfer-learning approach reported in [33] both for cross-camera and same-camera setups (cf. Table II). Apart from the superior recognition performance, our method does not require any knowledge about actions in test views which is not the case for [33], [35].

Impact of multiple temporal scales: Table III presents recognition results for SSM-hog-of descriptors computed at multiple and single temporal scales. Comparing single-scale descriptors, we observe that accuracy increases with the temporal extent of the descriptor tested for descriptor sizes 28 frames (1,1sec.), 42 frames (1,9sec.) and 56 frames (2,2sec.). Combining different scales, however, results in the considerable increase of performance compared to single-scale results for all camera setups.

VI. CONCLUSION

We propose a self-similarity based descriptor for view-independent video analysis, with human action recognition as a central application. Self-similarity being possibly defined over a variety of image features, either static (histograms of intensity gradient directions) or dynamic (optical flows or point trajectories), these descriptors can take different form and can be combined for increased descriptive power. Experimental validation on action recognition, as well as for the different problem of action synchronization, clearly confirms the stability of this type of description with respect to view variations. Results on public multi-view action recognition datasets demonstrate superior performance of our method compared to alternative methods in the literature.

Such encouraging results are simply obtained by exploiting the stability across views of SSM patterns, with no need to rely on the delicate recovery of 3D structures nor on the estimation of correspondences across views. Our method only makes mild assumptions about the rough localization of a person in the frame. This lack of strong assumptions is likely to make this approach applicable to action recognition beyond controlled datasets when combined with modern techniques for person detection and tracking.

ACKNOWLEDGEMENTS.

This work was partially funded by the QUAERO project and the MSR/INRIA joint laboratory.

REFERENCES

[1] T. Moeslund, A. Hilton, and V. Krüger, "A survey of advances in vision-based human motion capture and analysis," *CVIU*, vol. 103, no. 2-3, pp. 90–126, November 2006.
[2] L. Wang, W. Hu, and T. Tan, "Recent developments in human motion analysis," *Pattern Recognition*, vol. 36, no. 3, pp. 585–601, March 2003.
[3] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *PAMI*, vol. 23, no. 3, pp. 257–267, March 2001.
[4] D. Weinland, R. Ronfard, and E. Boyer, "Free viewpoint action recognition using motion history volumes," *CVIU*, vol. 103, no. 2-3, pp. 249–257, November 2006.
[5] T. Syeda-Mahmood, M. Vasilescu, and S. Sethi, "Recognizing action events from multiple viewpoints," in *Proc. EventVideo*, 2001, pp. 64–72.
[6] A. Yilmaz and M. Shah, "Actions sketch: A novel action representation," in *Proc. CVPR*, 2005, pp. I:984–989.
[7] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *PAMI*, vol. 29, no. 12, pp. 2247–2253, December 2007.
[8] M. Grundmann, F. Meier, and I. Essa, "3d shape context and distance transform for action recognition," in *Proc. ICPR*, 2008, pp. 1–4.
[9] I. Laptev, "On space-time interest points," *IJCV*, vol. 64, no. 2/3, pp. 107–123, 2005.
[10] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *Proc. CVPR*, 2005, pp. I:405–412.
[11] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005, pp. 65–72.
[12] J. Niebles, H. Wang, and F. Li, "Unsupervised learning of human action categories using spatial-temporal words," in *Proc. BMVC*, 2006.
[13] A. Gilbert, J. Illingworth, and R. Bowden, "Scale invariant action recognition using compound features mined from dense spatio-temporal corners," in *Proc. ECCV*, 2008, pp. I: 222–233.
[14] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *Proc. ICPR*, 2004, pp. III:32–36.
[15] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. CVPR*, 2008.
[16] S. Ali, A. Basharat, and M. Shah, "Chaotic invariants for human action recognition," in *Proc. ICCV*, 2007.
[17] D. Weinland and E. Boyer, "Action recognition using exemplar-based embedding," in *Proc. CVPR*, 2008.
[18] K. Jia and D.-Y. Yeung, "Human action recognition using local spatio-temporal discriminant embedding," in *Proc. CVPR*, 2008.
[19] C. Sminchisescu, A. Kanaujia, Z. Li, and D. Metaxas, "Conditional models for contextual human motion recognition," in *Proc. CVPR*, 2005.
[20] L.-P. Morency, A. Quattoni, and T. Darrell, "Latent-dynamic discriminative models for continuous gesture recognition," in *Proc. CVPR*, 2007.
[21] S. B. Wang, A. Quattoni, L.-P. Morency, D. Demirdjian, and T. Darrell, "Hidden conditional random fields for gesture recognition," in *Proc. CVPR*, 2006.
[22] L. Wang and D. Suter, "Recognizing human activities from silhouettes: Motion subspace and factorial discriminative graphical model," in *Proc. CVPR*, 2007.
[23] P. Natarajan and R. Nevatia, "View and scale invariant action recognition using multiview shape-flow models," in *Proc. CVPR*, 2008, pp. 1–8.
[24] A. Yilmaz and M. Shah, "Recognizing human actions in videos acquired by uncalibrated moving cameras," in *Proc. ICCV*, 2005, pp. I:150–157.
[25] S. Carlsson, "Recognizing walking people," *I. J. Robotic Res.*, vol. 22, no. 6, pp. 359–370, 2003.
[26] C. Rao, A. Yilmaz, and M. Shah, "View-invariant representation and recognition of actions," *IJCV*, vol. 50, no. 2, pp. 203–226, November 2002.

[27] Y. Shen and H. Foroosh, "View invariant action recognition using fundamental ratios," in *Proc. CVPR*, 2008.
[28] V. Parameswaran and R. Chellappa, "View invariance for human action recognition," *IJCV*, vol. 66, no. 1, pp. 83–101, January 2006.
[29] A. Ogale, A. Karapurkar, and Y. Aloimonos, "View-invariant modeling and recognition of human actions using grammars," in *Proc. Workshop on Dynamic Vision*, 2006, pp. 115–126.
[30] M. Ahmad and S. Lee, "HMM-based human action recognition using multiview image sequences," in *Proc. ICPR*, 2006, pp. I:263–266.
[31] R. Li, T. Tian, and S. Sclaroff, "Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series," in *Proc. ICCV*, 2007.
[32] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and viterbi path searching," in *Proc. CVPR*, 2007.
[33] A. Farhadi and M. Tabrizi, "Learning to recognize activities from the wrong view point," in *Proc. ECCV*, 2008, pp. I: 154–166.
[34] P. Yan, S. M. Khan, and M. Shah, "Learning 4d action feature models for arbitrary view action recognition," in *Proc. CVPR*, 2008.
[35] D. Weinland, E. Boyer, and R. Ronfard, "Action recognition from arbitrary views using 3D exemplars," in *Proc. ICCV*, 2007.
[36] E. Shechtman and M. Irani, "Matching local self-similarities across images and videos," in *Proc. CVPR*, 2007.
[37] C. Benabdelkader, R. Cutler, and L. Davis, "Gait recognition using image self-similarity," *EURASIP J. Appl. Signal Process.*, vol. 2004, no. 1, pp. 572–585, January 2004.
[38] R. Cutler and L. Davis, "Robust real-time periodic motion detection, analysis, and applications," *PAMI*, vol. 22, no. 8, pp. 781–796, 2000.
[39] S. Carlsson, "Recognizing walking people," in *Proc. ECCV*, 2000, pp. I:472–486.
[40] J. Eckmann, S. Kamphorst, and D. Ruelle, "Recurrence plots of dynamical systems," *Europhysics Letters*, no. 4, pp. 973–977, 1987.
[41] N. Marwan, M. C. Romanoa, M. Thiela, and J. Kurthsa, "Recurrence plots for the analysis of complex systems," *Physics Letters A*, vol. 438, no. 5-6, pp. 237–329, 2007.
[42] G. McGuire, N. B. Azar, and M. Shelhamer, "Recurrence matrices and the preservation of dynamical properties," *Physics Letters A*, vol. 237, no. 1-2, pp. 43–47, 1997.
[43] E. Bradley and R. Mantilla, "Recurrence plots and unstable periodic orbits," *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 12, no. 3, pp. 596–600, 2002.
[44] J. S. Iwanski and E. Bradley, "Recurrence plots of experimental data: To embed or not to embed?" *Chaos: An Interdisciplinary Journal of Nonlinear Science*, vol. 8, no. 4, pp. 861–871, 1998.
[45] S. Lele, "Euclidean distance matrix analysis (EDMA): Estimation of mean form and mean form difference," *Mathematical Geology*, vol. 25, no. 5, pp. 573–602, 1993.
[46] C. Tomasi and J. Shi, "Good features to track," in *Proc. CVPR*, 1994.
[47] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. CVPR*, 2005, pp. I:886–893.
[48] B. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Image Understanding Workshop*, 1981, pp. 121–130.
[49] J. P. M. de Sa, *Applied Statistics Using SPSS, STATISTICA, MATLAB and R*. Springer Berlin Heidelberg, 2007.
[50] M. Marszałek, C. Schmid, H. Harzallah, and J. van de Weijer, "Learning object representations for visual object class recognition," 2007, the PASCAL VOC'07 Challenge Workshop, in conjunction with ICCV.
[51] I. Laptev, B. Caputo, C. Schüldt, and T. Lindeberg, "Local velocity-adapted motion events for spatio-temporal recognition," *CVIU*, vol. 108, no. 3, pp. 207–229, 2007.
[52] N. Ikizler and P. Duygulu, "Human action recognition using distribution of oriented rectangular patches," in *Workshop on Human Motion*, 2007, pp. 271–284.
[53] H. Wang, M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2009.

Imran N. Junejo received his Ph.D. in computer science from University of Central Florida, U.S.A in 2007. After a one year post-doc at INRIA-Rennes, he joined Department of Computer Sciences, University of Sharjah where he is currently working as an Assistant Professor. His current focus of research is Human Action Recognition from arbitrary views. Other areas of research interests include: Camera Calibration, Metrology, Path Modeling, Video Surveillance, Scene Understanding and Event Detection.

Emilie Dexter received the M.E. degree in Signal and Image Processing from Ecole Nationale Suprieure dElectronique, Informatique et Radiocommunications of Bordeaux, France and the M.Sc. degree in Signal and Image Processing from the University of Bordeaux 1, France, in 2005. She received the Ph.D. degree from the from University of Rennes in 2009. Her current research focuses primarily on event recognition/detection and image sequence synchronization.

Ivan Laptev is a full time researcher at the French National Institute for Research in Computer Science and Control, INRIA Paris. He received his PhD in Computer Science from the Royal Institute of Technology (KTH) in 2004 and his Master of Science degree from the same institute in 1997. He worked as a research assistant at the Technical University of Munich (TUM) during 1997-1999. He joined VISTA research team at INRIA Rennes in 2004 and moved to WILLOW research team at INRIA/ENS in 2009. His research areas include action, scene and object recognition from video and still images. Ivan has published over 30 papers at international conferences and journals of computer vision, he serves as Associate Editor for Image and Vision Computing Journal, he is a regular member of the program committee of major international conferences on computer vision.

Patrick Pérez received the engineering degree from École Centrale Paris in 1990, and the Ph.D. degree from University of Rennes in 1993. After one year as a postdoc in the Dpt of Applied Mathematics at Brown University (USA), he joined Inria (France) in 1994 as a full time researcher. From March 2000 to February 2004, he was with Microsoft Research (Cambridge, UK). He then returned to Inria as a senior researcher and took, in 2007, the direction of Vista research team of the Inria Rennes Center where present work was conducted. In October 2009, Patrick Pérez joined Thomson Corporate Research (France) as a senior researcher. His research focuses on models and algorithms for understanding, analyzing, and manipulating still and moving images. He is currently an Associate Editor for the IEEE Transactions on Pattern Intelligence and Pattern Analysis and member of the Editorial Board of the International Journal of Computer Vision.