Cross-View Action Recognition from Temporal Self-Similarities

Imran Junejo, Emilie Dexter, Ivan Laptev, Patrick Pérez

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES

PUBLICATION
INTERNE
N° 1895

CROSS-VIEW ACTION RECOGNITION FROM TEMPORAL SELF-SIMILARITIES

IMRAN JUNEJO, EMILIE DEXTER, IVAN LAPTEV, AND PATRICK PÉREZ

ISSN 1166-8687

IRISA
CAMPUS UNIVERSITAIRE DE BEAULIEU - 35042 RENNES CEDEX - FRANCE

IRISA

INSTITUT DE RECHERCHE EN INFORMATIQUE ET SYSTÈMES ALÉATOIRES
Campus de Beaulieu – 35042 Rennes Cedex – France
Tél. : (33) 02 99 84 71 00 – Fax : (33) 02 99 84 71 71
http://www.irisa.fr

Cross-View Action Recognition from Temporal Self-Similarities

Imran Junejo, Emilie Dexter, Ivan Laptev, and Patrick Pérez

Systèmes cognitifs
Projets VISTA

Publication interne n ˚ 1895 — June 2008 — 17 pages

Abstract: This paper concerns 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 descriptor not being strictly view-invariant, we provide intuition and experimental validation demonstrating the high stability of self-similarities under view changes. Self-similarity descriptors are also shown stable under action variations within a class as well as discriminative for action recognition. 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 cues captured by self-similarities 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 for training only.

Key-words: Action Recognition, Self-Similarity, Sequence Alignment

(Résumé : tsvp)

Centre National de la Recherche Scientifique
(UMR 6074) Université de Rennes 1 – Insa de Rennes

Institut National de Recherche en Informatique
et en Automatique – unité de recherche de Rennes

Résumé : Ce document traite de la reconnaissance dactions humaines sous des vues différentes. Nous nous intéressons aux auto-similarités temporelles des actions et observons la stabilité de telles mesures quelle que soit la vue considérée. Nous développons autour de cette constatation un descripteur qui reflète la structure des similarités et dissimilarités temporelles au sein dune action. Bien que ce descripteur ne soit pas strictement invariant aux changements de points de vue, nous proposons une validation intuitive et expérimentale démontrant la grande stabilité des auto-similarités pour des points de vue différents. De plus, ces descripteurs sont stables la variabilité des actions au sein dune même classe et discriminants pour la reconnaissance dactions. Il est intéressant de noter que les auto-similarités calculées à partir de caractéristiques différentes possèdent les mêmes propriétés et peuvent être utilisées de manière complémentaire. Notre méthode est simple et ne requiert ni estimation de structures ni mise en correspondances entre vues. Au lieu de cela, elle s'appuie sur les faibles informations géométriques de l'auto-similarité et les combine avec de lapprentissage pour une reconnaissance daction efficace dans un contexte de vues multiples. La méthode a été validée sur trois bases de données, et obtient des performances similaires ou supérieures aux méthodes afférentes. De plus, celle-ci a montré de bonnes performances y compris dans des conditions extrêmes, par exemple lorsque la reconnaissance daction est effectuée pour des vues de dessus alors que la phase dentrainement ne considère que des vues de côtés.

Contents

1 Introduction

There has been a long tradition of research on human action understanding and behavior recognition in the vision community. Even still, determining similarity between human actions stands out to be one of the core problems of computer vision. Recent surveys highlight the immense attention that this problem has attracted [1, 2]. A good solution to this problem, however, holds a tremendous potential for applications to various computer vision problems such as video indexing and archiving, human computer interaction, gesture recognition and video surveillance to name a few. Some of the key issues that have been a bottle-neck for this problem are (i) that a good kinematic tracking has proved to be hard, specially, requiring feature point correspondences between different actions has forced researchers to resort to manual point tracking, and (ii) the models often adopted are overly complex, making the methods computationally expensive and impractical.

We aim to address the problem of action recognition in realistic monocular videos. We strive to make the method simple and flexible, by not imposing overly restrictive assumptions, and yet still be able to perform action recognition from arbitrary views. This is a very difficult problem, as appearance of an action may drastically vary from one viewpoint to another. In addition to this viewpoint change, other factors that make the problem even more challenging are the perspective or affine distortions (depending on the model used), anthropometric variations, or the speed at which the action is performed. Therefore, to make the problem more tractable, various simplifications or restricted special cases have been considered over the years [3, 4, 5, 6, 7, 8, 9]. We aim at alleviating such constraints.

One common restrictive assumption is to rely on multiple cameras. [10, 5] employ the use of epipolar geometry. Point correspondences between actions are used to estimate the fundamental matrices to perform view-invariant action recognition. However, obtaining correct point correspondences between actions from low resolution real videos is still a challenging problem. [9, 11] 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. [7, 12] perform full 3D reconstruction from extracting silhouettes seen from multiple deployed cameras. These methods require a setup of multiple cameras or training on poses obtained from multiple views, which restricts the applicability of these methods and can also be quite expensive.

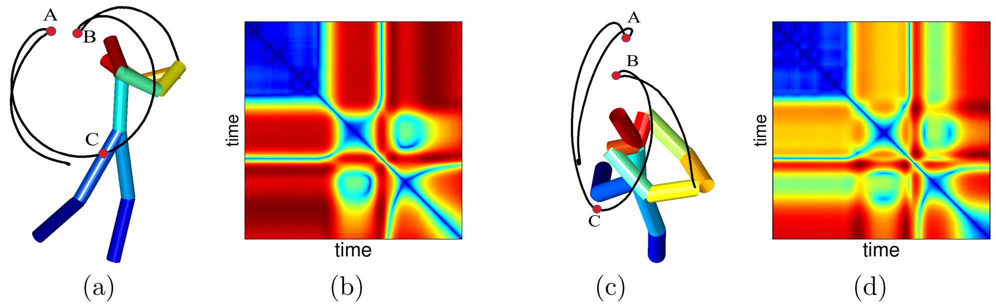
In contrast, we want to perform action analysis in monocular sequences associated to arbitrary viewpoints. To this end, one has to come up with (approximate) view-invariants. Rao et al. [3] show, for instance, that the maxima in space-time curvature of a 3D trajectory gets preserved in 2D image trajectories. However, these maxima (or dynamic instances) might not exist for an action, and also another limitation of this representation is that these instances might not always be preserved under the projection model. [4] propose 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. [13, 14] introduce space-time shapes, which are built by stacking together silhouettes of a tracked object, and various space-time features are then extracted from them to perform action recognition.

In contrast to these approaches, we propose to exploit a simple and yet a powerful tool based on “self-similarities”. Self-similarity of an action sequence, although not strictly view-invariant, is indeed surprisingly stable across views, irrespective of the precise features chosen to compute it. Its use allows us to devise a view-invariant action recognition system that requires neither structure recovery nor multi-view matching.

1.1 Related Work

The methods most closely related to our approach are that of [15, 16, 17]. Recently for image and video matching, based on the notion of self-similarity, [15] compute a *local* patch descriptor for every pixel. This is done by correlating the image patch centered at a pixel to its surrounding area by the simple Sum of Squared Differences (SSD). The descriptor (or the correlation surface) is transformed into a binned log-polar representation. Matching a template image to another image corresponds to finding a similar ensemble of descriptors in both images.

However, our approach is more related to the notion of video self-similarity as presented by [16, 17]. In the domain of periodic motion detection, Cutler and Davis [17] 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 this 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 use the Time-Frequency analysis. Following this, [16] use the same construct of the self-similarity matrix for gait recognition in videos of walking people. The periodicity



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

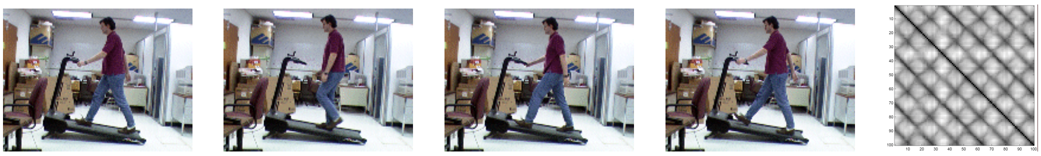
Figure 1: (a) and (c) demonstrate a golf swing action seen from two different views, and (b) and (d) represent their computed self-similarity matrices (SSM), respectively. Even though the two views are different, the structure or the patterns of the computed SSM are very similar.

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. However, while working only on image intensities, both of these works focus primarily on videos of walking people for periodic motion detection and gait analysis.

1.2 Our Approach: Overview

We propose a novel approach to action recognition by computing self-similarities of an action sequence. The proposed method is intuitive and flexible, in the sense that it can accommodate various features. We assume an Affine camera model, which is a fair assumption for real videos [3, 4]. We impose no restrictions on the pose or the camera viewpoint. The proposed self-similarity matrix (SSM) for an action sequence captures both the static and dynamic properties of the action. Using SSM, we contend that the similarities and dissimilarities of an action sequence are preserved under view variations. For example, Fig. 1 shows an action of a golf swing seen from two different views, Fig. 1(a) and Fig. 1(c), and their corresponding computed SSMs in Fig. 1(b) and Fig. 1(d), respectively. In both views, the points **A** and **B** are close to each other, i.e. the distance between **A** and **B** is low, while the distance between **A** and **C** is higher in both the views. Such qualitative similarities might explain the noticeable similarity in the "patterns" of the two SSMs. Our contention is that for different actions (from considerably different viewpoints), these SSM patterns are distinctive enough to be learned on a per-action basis or simply to synchronize multiple views of the same action.

The rest of the paper is organized as follows: Section 2 introduces the SSM and the various features that we use to compute it. Section 3 describes the learning based on these



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

Figure 2: (a)-(d) are four images from a sequence of a walking person. (e) represents the SSM obtained for this sequence by [17].

SSMs to perform action recognition. In Section 4, we test the method on three public datasets to demonstrate the practicality and the potential of the proposed method.

2 Self-Similarity Matrix (SSM)

The main contribution of the paper is the introduction of the self-similarity matrix (SSM), with the rationale that different action sequences produce SSMs of different patterns or structures, thus allowing us to perform action recognition.

For a sequence of images $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_T\}$, lying in discreet (x, y, t)-space, the square symmetric distance matrix $\mathcal{D}(\mathcal{I})$ lying in $\mathbb{R}^{T \times T}$ is defined as an exhaustive table of *distances* between image features taken by pair from the set $\mathcal{I}$:

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

where d_{ij} represents a distance between the frames $\mathcal{I}_i$ and $\mathcal{I}_j$. The diagonal corresponds to comparing an image to itself, hence, always zero. The exact structure or the patterns of $\mathcal{D}(\mathcal{I})$ depends on the features and the distance measure used for computing the entries d_{ij}. For example, after tracking walking people in a video sequence, [16, 17] compute d_{ij} as the absolute correlation between two frames, an example of which is shown in Fig. 2. The computed matrix patterns (cf. Fig. 2(e)) have a significant meaning for their application - the diagonals in the matrix indicate periodicity of the motion.

In this work, to compute d_{ij}, we use the Euclidean distance to measure the distance between the different features that we extract from an action sequence. This form of $\mathcal{D}(\mathcal{I})$ is then known in the literature as the Euclidean Distance Matrix (EDM)[18].

Before describing the features that we use, some word about the importance of matrix $\mathcal{D}$ is in order. From morphometrics and isometric reconstruction to non-linear dimensionality reduction, this matrix has proven to be a very useful tool for a variety of applications,

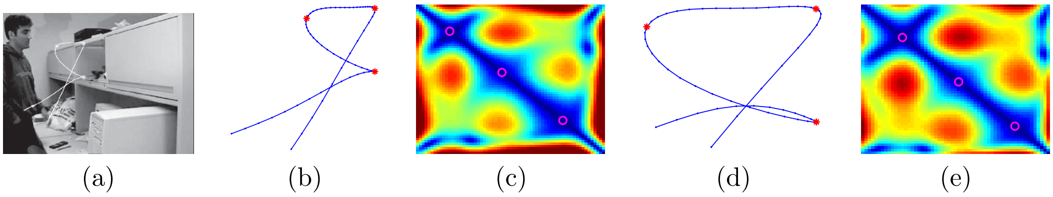


Figure 3: Comparison of the proposed SSM with [3]: Two actors perform the action of opening a cabinet door from different viewpoints, where the hand trajectory is shown in (b) and (d). The computed SSM for these two actions are shown in (c) and (e), respectively. The dynamic instances (as proposed by [3]), marked in red '*' in (b) and (d), represent valleys in the corresponding SSM, depicted by magenta circle in (c) and (e), respectively. The spread of each valley depends on the peak-width of the corresponding dynamic instance.

although never before for action recognition. For example, Isomap [19], a popular non-linear dimensionality reduction method, starts by computing distances between all pairs of images. These computed distances represent an adjacency matrix where each image represents a node in the graph. Note also that this notion of self-similarity, unlike [5] or [3], does not require point correspondences or time-alignment between *different actions* to perform action recognition.

To get an intuitive understanding of the proposed method, a comparison of SSM with the notion of "dynamic instances", as proposed by Rao et al.[3], is shown in Fig. 3. [3] 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. 3. Fig. 3(c)(e) shows the SSMs computed for these two actions, where red color indicates higher values and dark blue color indicates lower values. The dynamic instances, red '*' in Fig. 3(b)(d), correspond to valleys of different area/spread in our plot of SSM (cf. Fig. 3(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. 3(b)(d). However, whereas [3] captures 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. Our observation is that for different action (from considerably different viewpoints), these patterns of SSM are discriminative.

SSMs are fairly robust, handles missing (or noisy) data robustly, and is fairly easy to compute [18]. In a way, SSM can be visualized as measuring how *unlikely* is it that two images or two poses of an action are the same. Although, the patterns depicted by $\mathcal{D}(\mathcal{I})$ have no direct physical meaning, intuitively, *they capture both the static and dynamic properties of the 3D (space-time) action shape* [16]. The computation of SSM is also fairly flexible, in the sense that we can choose from a variety of different features, depending on the available data. Below we describe some of the features we use to compute the SSM:

2.1 Trajectory-based Self-Similarities

If the joints of a subject are tracked over some time, the Euclidean distance between the position of these tracked joints for any two frames of the sequence can be computed as:

$$d_{ij} = \sum_{k} \|x_i^k - x_j^k\|_2 \tag{2}$$

where the superscript k indicates the joint being tracked, and the subscript indicates the frame number of the sequence $\mathcal{I}$ for which $\mathcal{D}(\mathcal{I})$ is being computed. We denote this computed matrix by SSM-pos. In our experiments with motion capture dataset, we track 13 joints on a person performing different actions [8], as shown in the Fig. 4(a). In order to remove the effect of translation, without loss of generality, the points are centered to their centroid so that their first moments are zero. The remaining scale normalization is achieved by $\mathbf{x}_i = \frac{\mathbf{x}'_i}{\|\mathbf{x}'_i\|}$, where $\mathbf{x}'_i$ represent the joints being tracked in frame i and $\mathbf{x}_i$ represent their normalized coordinates.

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

It is beyond the scope of the current work to describe a joint or a point tracking algorithm, and like [10, 8, 4], we assume the points are correctly tracked when using these position based features. As mentioned above, we do not estimate any entity (like the fundamental matrix or the space-time shape [14]) that requires point correspondences between actions, rather, we only compute the SSM by using the features that capture the dynamics of an action sequence.

2.2 Image-based Self-Similarities

For video sequences where positional feature tracking fails, other features could be used to compute $\mathcal{D}(\mathcal{I})$. A simple choice is the Euclidean distance between Histograms of Oriented Gradients (HoG) features [20]. This descriptor, originally used to perform human detection, characterizes the local shape because it captures edge and gradient structure. In our implementation, we use 4 bin histograms for each 5 block defined on a bounding box around a foreground object in each frame. d_{ij} is then the Euclidean distance between two HoG vectors corresponding to the frames $\mathcal{I}_i$ and $\mathcal{I}_j$. The SSM computed by using HoG features is denoted by SSM-hog.

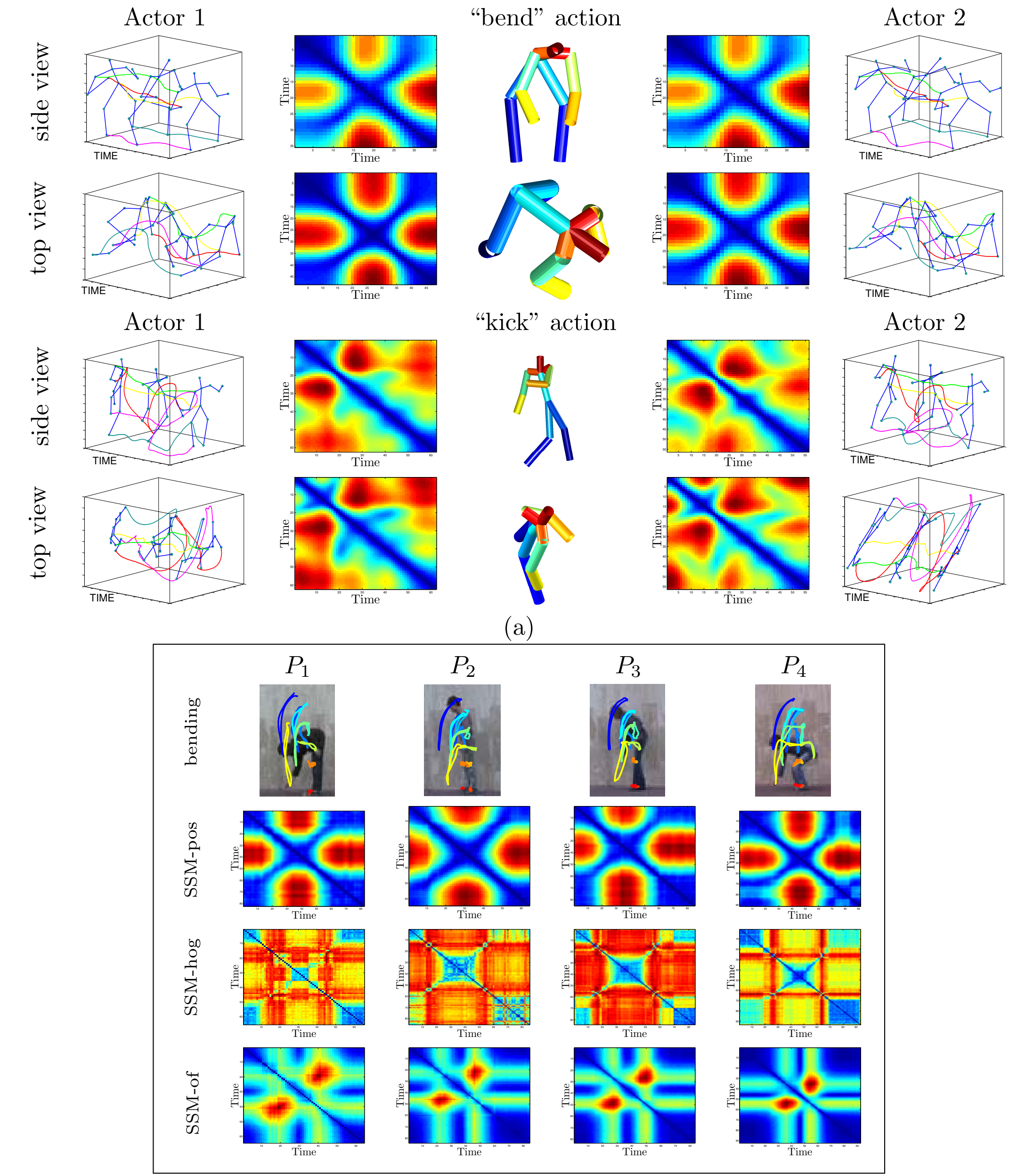
In addition to the HoG features, we also test the proposed method by considering the estimated optical flow vector as an input feature. The optical flow is calculated using the method described by Lucas and Kanade [21] on bounding boxes centered on the foreground object between two consecutive frames. Considering both components of the optical flow, three different SSMs are computed based on: x-direction flow vector (SSM-ofx), y-direction flow vector (SSM-ofy) and global vector concatenating both directions (SSM-of). Here also d_{ij} is measured as 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 ensure 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.

An example of the SSM, computed by using different features is shown in Fig. 4. Fig. 4(a) is an example from the CMU motion capture (mocap) database, projected in different views. Column 1 and 5 of Fig. 4(a) represent two different actors, and column 2 and 4 represent their computed SSMs, respectively. The first two rows represent an action of bending, where row 1 is almost a front view of an actor and row 2 is a viewpoint from behind an actor at a considerable height. Similarly, rows 3 and 4 represent an action of a football kick. Notice that even though there is a large variation in the actions performed and the different views in which the action is projected, the SSMs have a very similar form. Fig. 4(b) shows the SSMs obtained from a real dataset [13] for the action of bending. Row 2 depicts SSM-pos computed by tracking object movements (cf. [8]). Rows 3 and 4 show the computed matrix based on HoG and Optic Flow vectors, respectively. Rows $2, 3$ and 4 may not look similar, but notice the similarity column-wise. This is primarily due to the fact that each similarity feature captures different characteristics of the action.

3 SSM-based action description and recognition

As argued 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 action recognition. We pay attention to the following properties of SSM: (i) absolute values of SSM may depend on the variant properties of the data such as the projected size of a person in the case of SSM-pos; (ii) changes in temporal offsets and time warping 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; (iv) SSM is a symmetric positive semidefinite matrix with zero-valued diagonal.

Due to (ii)-(iii) we choose a local representation and compute patch-based descriptors centered at each diagonal element of SSM. Our patch descriptor has a log-polar block structure as illustrated in Fig. 5. For each of the 11 descriptor blocks we compute 8-bin histogram of SSM gradient directions within a block and concatenate the normalized histograms into a descriptor vector h_i. A joint local descriptor for m SSMs is constructed by concatenating corresponding local descriptors of each SSM into a single vector $h_i = (h_i^1, ..., h_i^m)$. The representation for a video sequence is finally defined by the sequence of local descriptors $H = (h_1, ..., h_n)$ computed for all diagonal elements of the corresponding SSM.



(a)

(b)

Figure 4: (a) contains an example from CMU mocap dataset. Column 1 and 5 represent two actors, and column 2 and 4 represent their computed SSM, respectively. The first two rows are for the action of bending, (each row represents a different 2D projection of the 3D action), while last two rows are for a football kick action. Middle column shows the approximate viewing angle of the synthetic camera. (b) contains some results of the computed SSMs for a real dataset [13]. Row 2 is the computed matrices based on point tracking (i.e. trajectory based) by using the SSM-pos. Rows 3 and 4 are computed based on the HoG and Optic Flow (OF) features, respectively. Note the similarity column-wise. See text for more details.

3.1 Temporal multi-view sequence alignment

Before addressing action recognition, we validate our representation on the problem of multi-view sequence alignment. We consider two videos recorded simultaneously for the side and the top views of a person in action as shown in Fig. 6(a). To further challenge the alignment estimation, we apply a nonlinear time transformation to one of the sequences. To solve alignment, we (i) compute SSM-of for both image sequences, (ii) represent videos by the sequences of local SSM descriptors H^1, H^2 as described above, (iii) and finally align sequences H^1 and H^2 by Dynamic Programming. The estimated time transformation is illustrated by the red curve in Fig. 6(b) and does almost perfectly recover the ground truth transformation (blue curve) despite the drastic view variation between image sequences.

3.2 Action recognition

To recognize action sequences we follow recently successful bag-of-features approaches [22, 23] and represent each video as a bag of local SSM descriptors H. We then apply either Nearest Neighbour Classifier (NNC) or Support Vector Mahcines (SVM) to train and classify

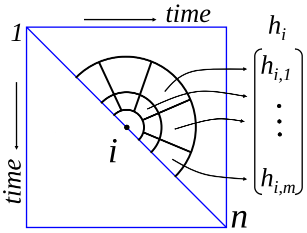
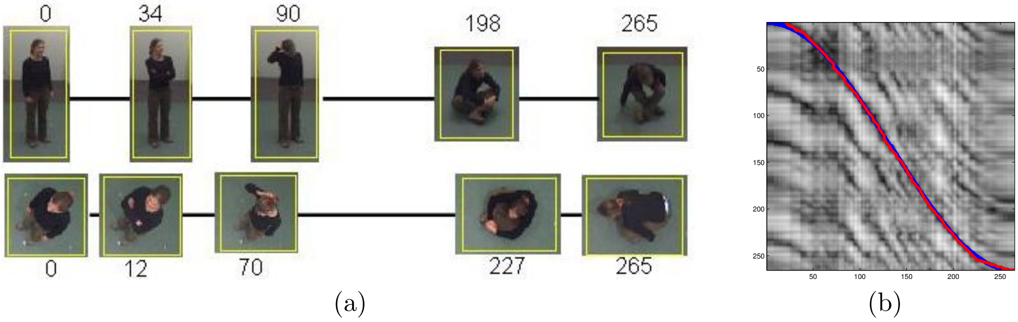


Figure 5: Local descriptors for SSM are centered at every diagonal point $i = 1...n$ and have log-polar block structure. Histograms of gradient directions are computed separately for each block and concatenated into descriptor vector h_i.



(a) (b)

Figure 6: Temporal sequence alignment. (a): Two sequences with the side and the top views of the same action are represented by corresponding key-frames. The lower sequence has been time warped according to $t' = a\cos(bt)$ transformation. (b): Alignment of two sequences in (a) using SSM-based action descriptions and Dynamic Programming (red curve) recovers the original warping (blue curve) almost perfectly despite substantial view variations.

instances of action classes. In the case of NNC, we assign a test sequence H_{tst} with the label of a training sequence H_{tr}^{i} with $i = \text{argmin}_j \, D_{NN}(H_{tst}, H_{tr}^{j})$ minimizing the distance over all training sequences. The distance D_{NN} is defined by the greedy matching of local descriptors as described in [22]. We apply NNC to datasets with a limited number of samples.

For SVMs we construct histograms of visual words and use them as input for SVM training and classification according to [24]. Visual vocabulary is obtained by k-means clustering of 10000 local SSM descriptors h from the training set into $k = 1000$ clusters. Each feature is then assigned to the closest (we use Euclidean distance) vocabulary word and the histogram of visual words is computed for each image sequence. We train non-linear SVMs using χ^2 kernel and adopt one-against-all approach for multi-class classification.

For all recognition experiments in the next section 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.

4 Experimental results

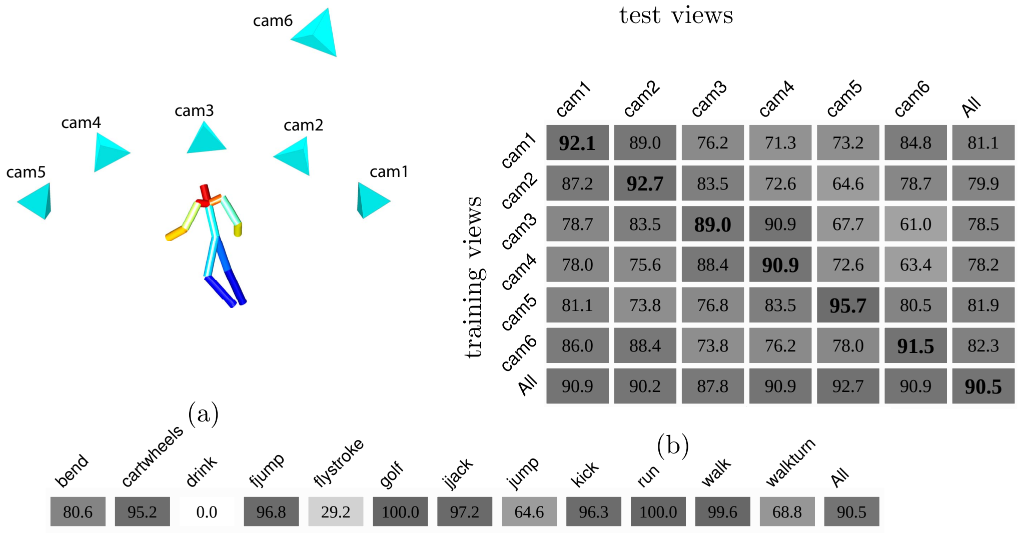
In this section we evaluate SSM-based action descriptors for the task of multi-view action recognition. The first experiment in Section 4.1 aims to validate the approach in controlled multi-view settings using motion capture data. In Section 4.2 we demonstrate and compare the discriminative power of our method on the standard single-view action dataset [13]. We finally evaluate the performance of the method on the comprehensive multi-view action dataset [12] in Section 4.3.

4.1 Experiments with CMU MoCap dataset

To simulate multiple and controlled view settings we have used 3D motion capture data from CMU dataset (*mocap.cs.cmu.edu*). Trajectories of 13 points on the human body were projected to six cameras with pre-defined orientation with respect to the human body (see Fig. 7(a)). We have used 164 sequences in total corresponding to 12 action classes. To simulate potential failures of the visual tracker we also randomly subdivided trajectories into parts with the average length of 2 seconds. Fig. 7(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 changes slowly with substantial view changes and remains high across top and side views. When training and testing on all views, the average accuracy is 90.5%. The per-class accuracy is illustrated in Fig. 7(c).

4.2 Experiments with Weizman actions dataset

To asses the discriminative power of our method on real sequences we apply it to the standard single-view video dataset with nine classes of human actions performed by nine sub-



(a)

training views \ test views	cam1	cam2	cam3	cam4	cam5	cam6	All
cam1	**92.1**	89.0	76.2	71.3	73.2	84.8	81.1
cam2	87.2	**92.7**	83.5	72.6	64.6	78.7	79.9
cam3	78.7	83.5	**89.0**	90.9	67.7	61.0	78.5
cam4	78.0	75.6	88.4	**90.9**	72.6	63.4	78.2
cam5	81.1	73.8	76.8	83.5	**95.7**	80.5	81.9
cam6	86.0	88.4	73.8	76.2	78.0	**91.5**	82.3
All	90.9	90.2	87.8	90.9	92.7	90.9	**90.5**

(b)

bend	cartwheels	drink	fjump	flystroke	golf	jjack	jump	kick	run	walk	walkturn	All
80.6	95.2	0.0	96.8	29.2	100.0	97.2	64.6	96.3	100.0	99.6	68.8	90.5

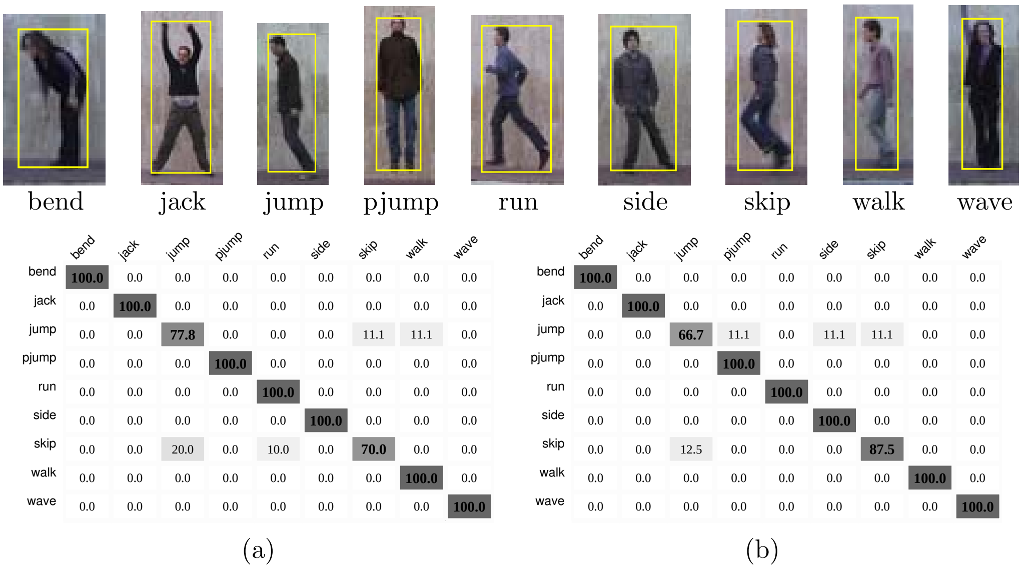
(c)

Figure 7: CMU dataset. (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. (c): Recognition accuracy for all classes.

jects [13](see Fig. 8(top)). On this dataset we compute NNC recognition accuracy when using either image-based self-similarities in terms of SSM-of-ofx-ofy-hog or trajectory-based SSM. Given the low resolution of image sequences in this dataset, the trajectories were acquired by [8] via semi-automatic tracking of body joints. Recognition accuracy achieved by our method for image-based and trajectory-based self-similarities is 94.6% and 95.3% respectively and the corresponding confusion matrices are illustrated in Fig. 8(a)-(b). The recognition results are similarly high for both types of self-similarity descriptors and outperform 92.6% achieved in [8].

4.3 Experiments with IXMAS dataset

We finally present results for IXMAS video dataset [12] 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. 9. Given the relatively large number of training samples, we apply SVM classification to image-based self-similarity descriptors in terms of SSM-oh-ofx-ofy-hog. Fig. 10(a) illustrates recognition accuracy for cross-view training and testing. Similar to results on CMU dataset in Section 4.1, here we observe high stability of action recognition over view changes, now using visual data only. The method achieves reasonable accuracy even for top views when using side-views for



	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

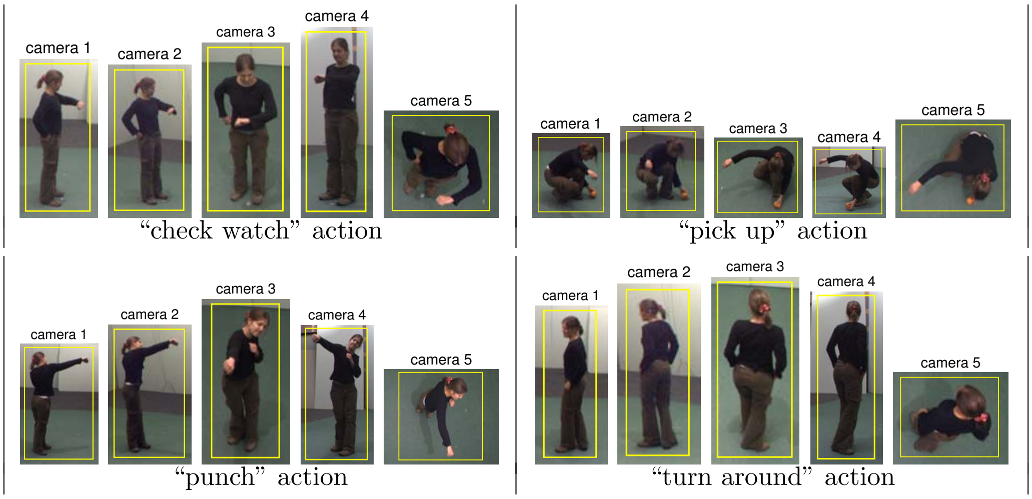
(b)

Figure 8: (Top): Example frames for Weizman action dataset [13] with image sequences for nine classes of actions. (a)-(b) confusion matrices corresponding to NNC action recognition using image-based self-similarities (a) and trajectory-based self-similarities (b).

training only. Fig. 10(c) illustrates recognition scores for different types of self-similarities and their combinations. We can observe the advantage of SSM-of over SSM-hog, however, the best results are achieved when combining self-similarities for several complementary features. In comparison to other methods, our method outperforms both 2D and 3D based recognition methods in [12] for all test scenarios as shown in Fig. 10(d). We may add that our method relies on the rough localization and tracking of people in the scene and, hence, relies on weaker assumptions compared to [12] that uses human silhouettes.

5 Conclusion

We propose a self-similarity based descriptor for view-independent action recognition. Experimental validation on several datasets using different types of self-similarities clearly confirms the stability of our approach to view variations. The proposed method does not rely on the structure recovery nor on the correspondence estimation, but makes only mild assumptions about the rough localization of a person in the frame. This lack of strong assumptions is likely to make our method applicable to action recognition beyond controlled datasets when combined with the modern techniques for person detection and tracking. We plan to investigate this direction in the future work.



“check watch” action

“pick up” action

“punch” action

“turn around” action

Figure 9: Example frames for four action classes and five views of the IXMAS dataset.

test views

training views	cam1	cam2	cam3	cam4	cam5	All
cam1	**76.4**	77.6	69.4	70.3	44.8	67.2
cam2	77.3	**77.6**	73.9	67.3	43.9	67.4
cam3	66.1	70.6	**73.6**	63.6	53.6	65.0
cam4	69.4	70.0	63.0	**68.8**	44.2	63.9
cam5	39.1	38.8	51.8	34.2	**66.1**	45.2
All	74.8	74.5	74.8	70.6	61.2	**72.7**

(a)

	check–watch	cross–arms	scratch–head	sit–down	get–up	turn–around	walk	wave	punch	kick	pick–up
check–watch	**83.3**	0.0	0.7	1.3	0.7	1.3	8.0	0.7	0.0	0.0	4.0
cross–arms	0.0	**94.0**	2.0	1.3	0.7	0.7	0.0	0.7	0.0	0.0	0.7
scratch–head	0.0	0.0	**68.7**	2.0	9.3	2.0	1.3	4.7	10.0	2.0	0.0
sit–down	0.7	4.7	3.3	**55.3**	1.3	20.0	3.3	0.7	10.7	0.0	0.0
get–up	2.0	3.3	7.3	0.7	**59.3**	0.7	0.0	23.3	2.7	0.7	0.0
turn–around	3.3	1.3	0.0	27.3	0.0	**56.7**	3.3	2.0	2.7	0.0	3.3
walk	10.0	0.7	0.0	2.7	0.7	2.7	**68.7**	1.3	1.3	0.0	12.0
wave	3.3	0.7	6.7	2.0	14.7	0.0	0.7	**63.3**	8.7	0.0	0.0
punch	0.7	0.0	6.0	6.0	0.7	2.7	0.0	1.3	**74.0**	8.7	0.0
kick	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	**100.0**	0.0
pick–up	2.0	0.0	0.0	2.7	0.7	4.7	13.3	0.7	0.0	0.0	**76.0**

(b)

	All-to-All
hog	57.8%
of	65.9%
of+ofx+ofy	66.5%
of+hog	71.9%
of+hog+ofx+ofy	**72.7%**

(c)

	cam1	cam2	cam3	cam4	cam5
This paper	**76.4%**	**77.6%**	**73.6%**	**68.8%**	**66.1%**
Weinland et al. [12] 3D	65.4%	70.0%	54.3%	66.0%	33.6%
Weinland et al. [12] 2D	55.2%	63.5%	—	60.0%	—

(d)

Figure 10: Results for action recognition on IXMAS dataset. (a): Recognition accuracy for cross-view training and testing. (b): confusion matrix for action recognition in “all-training all-testing” setting. (c): relative performance of combined self-similarity descriptors. (d): Comparison with [12] for “camN-training camN-testing” setup.

References

[1] Moeslund, T.B., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. Computer Vision and Image Understanding **104** (2006) 90–126

[2] Wang, L., Hu, W., Tan, T.: Recent developments in human motion analysis. Pattern Recognition **36** (2003) 585–601

[3] Rao, C., Shah, M.: A view-invariant representation of human action. Intl. Journal of Compter Vision 50(2) (2002) 203–226

[4] Parameswaran, V., Chellappa, R.: View invariance for human action recognition. IJCV **66** (2006) 83–101

[5] Syeda Mahmood, T., Vasilescu, M., Sethi, S.: Recognizing action events from multiple viewpoints. In: EventVideo. (2001) 64–72

[6] Cuntoor, N., Chellappa, R.: Epitomic representation of human activities. In: CVPR. (2007) 1–8

[7] Weinland, D., Ronfard, R., Boyer, E.: Free viewpoint action recognition using motion history volumes. CVIU **103** (2006) 249–257

[8] Ali, S., Basharat, A., Shah, M.: Chaotic invariants for human action recognition. In: ICCV. (2007) 1–8

[9] Li, R., Tian, T., Sclaroff, S.: Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In: ICCV. (2007) 1–8

[10] Yilmaz, A., Shah, M.: Recognizing human actions in videos acquired by uncalibrated moving cameras. In: In Proc. ICCV Volume 1. (2005) 150–157

[11] Ogale, A.S., Karapurkar, A., Aloimonos, Y.: View-invariant modeling and recognition of human actions using grammars. International Conference on Computer Vision, Workshop on Dynamical Vision (ICCV-WDM) (2005)

[12] Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3d exemplars. In: ICCV. (2007) 1–7

[13] Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. PAMI **29** (2007) 2247–2253

[14] Yilmaz, A., Shah, M.: Actions sketch: A novel action representation. In Proc. CVPR **1** (2005) 984–989

[15] Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: In Proc. CVPR. (2007)

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

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

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

[19] Tenenbaum, J.B., de Silva, V., Langford, J.C.: A global geometric framework for nonlinear dimensionality reduction. Science **290** (2000) 2319–2323

[20] Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: In Proc. CVPR. Volume 2. (2005) 886–893

[21] Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Image Understanding Workshop. (1981) 121–130

[22] Laptev, I., Caputo, B., Schüldt, C., Lindeberg, T.: Local velocity-adapted motion events for spatio-temporal recognition. CVIU **108** (2007) 207–229

[23] Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised learning of human action categories using spatial-temporal words. In: BMVC. (2006)

[24] Marszałek, M., Schmid, C., Harzallah, H., van de Weijer, J.: Learning object representations for visual object class recognition (2007) The PASCAL VOC'07 Challenge Workshop, in conjunction with ICCV.