Cross-View Action Recognition from Temporal Self-similarities

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

INRIA Rennes - Bretagne Atlantique
35042 Rennes Cedex - France

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 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 for training only.

1 Introduction

Visual recognition and understanding of human actions has attracted much of attention over the past three decades [1,2] and remains an active research area of computer vision. A good solution to the problem holds a huge 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. Previous work demonstrated the high 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. In addition, actions frequently involve and depend on manipulated objects adding another layer of variability. As a consequence, current methods often resort to restricted and simplified scenarios with simple backgrounds, a few simple kinematic action classes, static cameras and limited view variations.

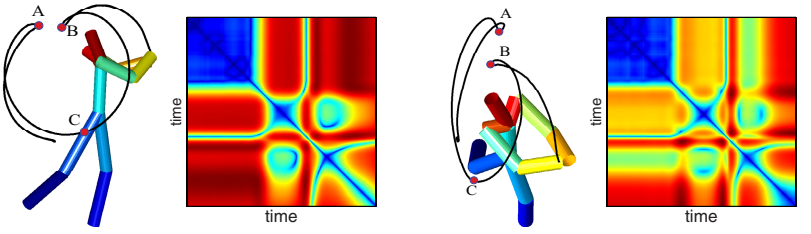
D. Forsyth, P. Torr, and A. Zisserman (Eds.): ECCV 2008, Part II, LNCS 5303, pp. 293–306, 2008.

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 pose substantial challenges for computer vision algorithms.

To address the multi-view problem, [3,4] employ the use of epipolar geometry. Point correspondences between actions are assumed to be known for imposing fundamental matrix constraints and performing view-invariant action recognition. [5] 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 [6] showed that fundamental ratios can be used for view-invariant action recognition as well. However, obtaining automatic and reliable point correspondences for daily videos with natural human actions is a very challenging and currently unsolved problem prohibiting the application of above mentioned methods in practice. One alternative to the geometric approach is to represent actions by samples recorded for different views. [7,8] 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. Another method [9] performs 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.

In this work we address view-independent action recognition from a different perspective and avoid many assumptions of previous methods. In contrast to the geometry-based methods above we require neither 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 we compute distances between action representations for all pairs of time-frames and store results in a Self-Similarity Matrix (SSM). We claim SSMs to be approximately invariant under view changes of an action. The intuition behind this claim is the following. If body poses of an action are similar at moments t_1, t_2, the value of $\mathrm{SSM}(t_1, t_2)$ will be low for any view of that action. On the contrary, if the body poses are different at t_1, t_2, the value of $\mathrm{SSM}(t_1, t_2)$ is likely to be large for most of the views. Fig. 1 illustrates this idea with an example of a golf swing action seen from two different views. For this example we compute SSMs using distances of points on the hand trajectory illustrated in Fig. 1(a,c). Close trajectory points **A**, **B** remain close in both views while the distanced 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 difference in the projections of the action.



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

Fig. 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 SSMs is very similar.

In the rest of the paper we operationalize SSMs for human action sequences and deploy SSMs 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 rest of the paper is organized as follows. In the next section we review related work. Section 2 gives a formal definition of SSM using alternative image feature. Section 3 describes the representation and training of action sequences based on SSMs. In Section 4 we test the method on three public datasets and demonstrate the practicality and the potential of the proposed method. Section 5 concludes the paper.

1.1 Related Work

The methods most closely related to our approach are that of [10,11,12,13]. Recently for image and video matching [10] explored *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 [10].

Our approach has a closer relation to the notion of video self-similarity used by [11,12]. In the domain of periodic motion detection, Cutler and Davis [12] 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 use the Time-Frequency analysis. Following this, [11] use 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

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

Fig. 2. (a)-(d) are four images from a sequence of a walking person. (e) represents the SSM obtained for this sequence by [12].

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 [13] 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.

2 Self-Similarity Matrix (SSM)

In this section we define self-similarity matrices for different image features and illustrate SSMs computed for several action classes and multiple views.

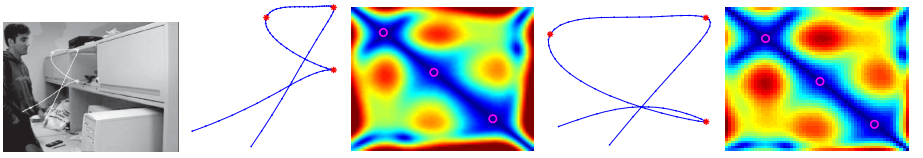
For a sequence of images $\mathcal{I} = \{\mathcal{I}_1, \mathcal{I}_2, \ldots \mathcal{I}_T\}$ in discreet (x, y, t)-space the square symmetric distance matrix $\mathcal{D}(\mathcal{I})$ in $\mathbb{R}^{T \times T}$ is defined as an exhaustive table of *distances* between image features taken by pair 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 a frame 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, [11,12] 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)[14].

Before describing the features that we use, some words about the importance of matrix $\mathcal{D}$ are in order. From morphometrics and isometric reconstruction



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

Fig. 3. Comparison of the proposed SSM with [16]: 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 [16]), 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.

to non-linear dimensionality reduction, this matrix has proven to be a useful tool for a variety of applications. For example, Isomap [15], 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.

To get an intuitive understanding of the proposed method, a comparison of SSM with the notion of "dynamic instances" proposed by Rao et al.[16] is illustrated in Fig. 3. The authors of [16] 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 [16] captures local discontinuities in the spatio-temporal curvature, the SSM captures more information about other dynamics of the actions represented in the off-diagonal parts of the matrix. We will argue that for actions recorded from different viewpoints the patterns of SSM are stable and discriminative.

SSMs are fairly robust, handle missing (or noisy) data robustly, and are fairly easy to compute [14]. The computation of SSM is flexible in the sense that we can choose to compute SSMs 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 a subset of points distributed over the person body are tracked over some time, the mean Euclidean distance for k pairs of points at any two frames of the sequence can be computed as

$$d_{ij} = \frac{1}{k} \sum_{k} \|x_i^k - x_j^k\|_2 \tag{2}$$

where x^k, x_j^k indicate positions of points on the track k at frames i and j. We denote the self-similarity matrix computed from (2) by SSM-pos. In our experiments with motion capture dataset, we track 13 joints on a person performing different actions [17], 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.

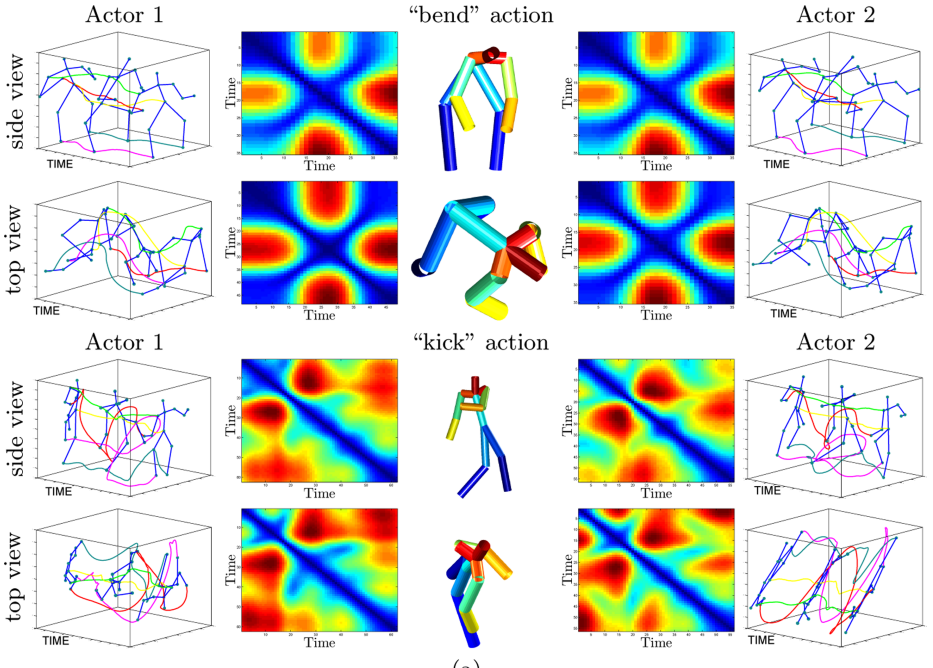
In this work we assume point tracks to be provided by an external module such as KLT point tracker. Our method is not restricted to any particular subset of points as far as the points are distributed over moving body parts. SSMs can be accumulated from any number of tracks with arbitrary length and starting time.

2.2 Image-Based Self-similarities

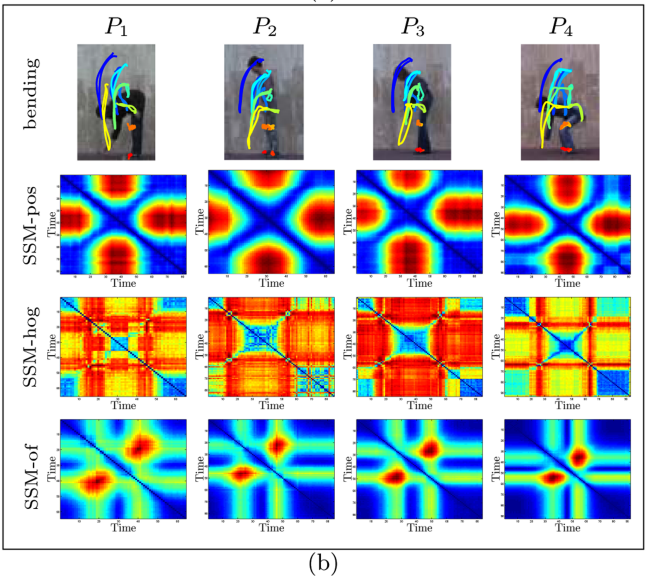
Next to the trajectories, alternative image features can be used to construct additional $\mathcal{D}(\mathcal{I})$ for the same image sequence. To describe spatial appearance of a person at each image frame we compute Histograms of Oriented Gradients (HoG) features [18]. This descriptor, originally 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. d_{ij} is then computed as the Euclidean distance between two HoG vectors corresponding to the 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 the estimated optical flow vector as an input feature. The optical flow is computed by Lucas and Kanade method [19] on person-centered bounding boxes using two consecutive frames. We consider v_x and v_y components of optical flow and define three features by concatenating responses of v_x, v_y and (v_x,v_y) into descriptor vectors. The SSMs computed for these three features are denoted as SSM-ofx, SSM-ofy and SSM-of respectively. We measure d_{ij} using Euclidean distance between the flow descriptor 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 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)



(a)



(b)

Fig. 4. (a) Examples from CMU mocap dataset. Columns 1 and 5 represent two actors while columns 2 and 4 represent corresponding SSM-pos, 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 [20]. Rows $2, 3, 4$ represent SSM-pos, SSM-hog and SSM-of respectively for four bending actions in row 1. Note the similarity column-wise.

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. Fig. 4(b) illustrates SSMs obtained for the bending action from the video dataset [20]. Row 2 shows SSM-pos computed using point tracks illustrated in the row 1. Rows 3 and 4 show SSM-hog and SSM-of for the same sequences respectively. Note the similarity of SSMs for the same feature type and for the different action instances. 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.

3 SSM-Based Action Description and Recognition

As illustrated 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 i of SSM. Our patch descriptor has a log-polar block structure as illustrated in Fig. 5. For each of the 11 descriptor blocks j we compute 8-bin histogram of SSM gradient directions within a block and concatenate the normalized histograms into a descriptor vector h_i. When constructing a joint local descriptor for multiple SSMs computed for k different features, we concatenate k corresponding descriptors from each SSM into a single vector. 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 SSM.

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

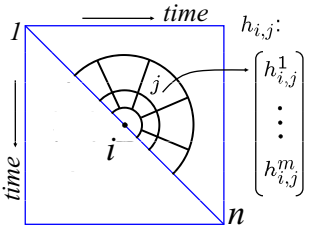
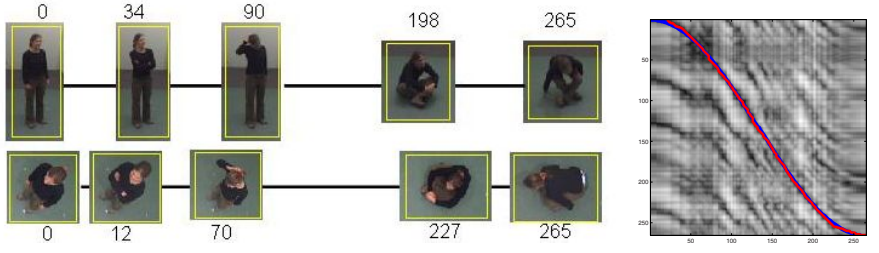


Fig. 5. Local descriptors for SSM are centered at every diagonal point $i = 1...n$ and have log-polar block structure. Histograms of m gradient directions are computed separately for each of 11 blocks j and are concatenated into a descriptor vector h_i.



(a) (b)

Fig. 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.

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 [21,22] and represent each video as a bag of local SSM descriptors H. We then apply either Nearest Neighbour Classifier (NNC) or Support Vector Machines (SVM) to train and classify instances of action classes. In the case of NNC, we assign a test sequence H_{tst} with the label of a training sequence H^i_{tr} with $i = \text{argmin}_j\, D_{NN}(H_{tst}, H^j_{tr})$ minimizing the distance over all training sequences. The distance D_{NN} is defined by the greedy matching of local descriptors as described in [21]. 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 [23]. 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

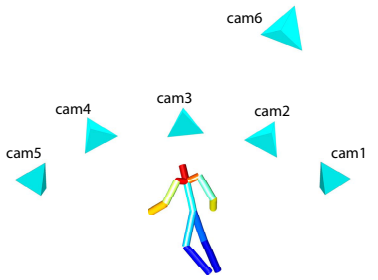
In this section we evaluate SSM-based action descriptors for the task of multi-view action recognition. The first experiment in Section 4.1 aims to validate the approach in controlled multi-view settings using motion capture data. In Section 4.2 we demonstrate and compare the discriminative power of our method on the standard single-view action dataset [20]. We finally evaluate the performance of the method on the comprehensive multi-view action dataset [9] 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 (*http://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%.

4.2 Experiments with Weizman Actions Dataset

To asses the discriminative power of our method on real video sequences we apply it to the standard single-view video dataset with nine classes of human actions performed by nine subjects [20](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 [17] via semi-automatic tracking of body joints. Recognition accuracy achieved by our



(a)

test views

training 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)

Fig. 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.

	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)

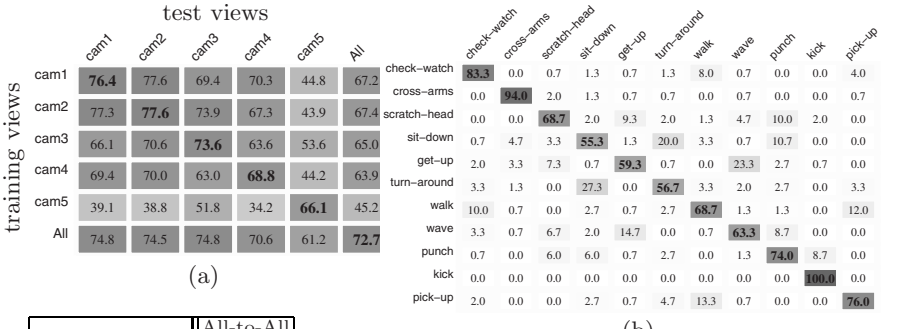
	bend	jack	jump	pjump	run	side	skip	walk	wave
bend	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jack	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
jump	0.0	0.0	66.7	11.1	0.0	11.1	11.1	0.0	0.0
pjump	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0
run	0.0	0.0	0.0	0.0	100.0	0.0	0.0	0.0	0.0
side	0.0	0.0	0.0	0.0	0.0	100.0	0.0	0.0	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. 8. (Top): Example frames for Weizman action dataset [20] 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).

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 high for both types of self-similarity descriptors and outperform 92.6% achieved in by a recent trajectory-based method in [17]. Higher recognition rates on the same dataset have been reported e.g. in [24].

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



	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. [9] 3D	65.4%	70.0%	54.3%	66.0%	33.6%
Weinland et al. [9] 2D	55.2%	63.5%	—	60.0%	—

(d)

Fig. 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 [9] for "camN-training camN-testing" setup.

4.3 Experiments with IXMAS Dataset

We finally present results for IXMAS video dataset [9] 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 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 [9] 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 [9] 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.

References

1. Moeslund, T., Hilton, A., Krüger, V.: A survey of advances in vision-based human motion capture and analysis. CVIU 103, 90–126 (2006)
2. Wang, L., Hu, W., Tan, T.: Recent developments in human motion analysis. Pattern Recognition 36, 585–601 (2003)
3. Yilmaz, A., Shah, M.: Recognizing human actions in videos acquired by uncalibrated moving cameras. In: Proc. ICCV, pp. I:150–157 (2005)
4. Syeda-Mahmood, T., Vasilescu, M., Sethi, S.: Recognizing action events from multiple viewpoints. In: Proc. EventVideo, pp. 64–72 (2001)
5. Parameswaran, V., Chellappa, R.: View invariance for human action recognition. IJCV 66, 83–101 (2006)
6. Shen, Y., Foroosh, H.: View invariant action recognition using fundamental ratios. In: Proc. CVPR (2008)
7. Li, R., Tian, T., Sclaroff, S.: Simultaneous learning of nonlinear manifold and dynamical models for high-dimensional time series. In: Proc. ICCV (2007)
8. Ogale, A., Karapurkar, A., Aloimonos, Y.: View-invariant modeling and recognition of human actions using grammars. In: Proc. W. on Dyn. Vis., pp. 115–126 (2006)
9. Weinland, D., Boyer, E., Ronfard, R.: Action recognition from arbitrary views using 3D exemplars. In: Proc. ICCV (2007)
10. Shechtman, E., Irani, M.: Matching local self-similarities across images and videos. In: Proc. CVPR (2007)

11. Benabdelkader, C., Cutler, R., Davis, L.: Gait recognition using image self-similarity. EURASIP J. Appl. Signal Process 2004, 572–585 (2004)
12. Cutler, R., Davis, L.: Robust real-time periodic motion detection, analysis, and applications. PAMI 22, 781–796 (2000)
13. Carlsson, S.: Recognizing walking people. In: Vernon, D. (ed.) ECCV 2000. LNCS, vol. 1842, pp. 472–486. Springer, Heidelberg (2000)
14. Lele, S.: Euclidean distance matrix analysis (EDMA): Estimation of mean form and mean form difference. Mathematical Geology 25, 573–602 (1993)
15. Tenenbaum, J., de Silva, V., Langford, J.: A global geometric framework for nonlinear dimensionality reduction. Science 290, 2319–2323 (2000)
16. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. IJCV 50(2), 203–226 (2002)
17. Ali, S., Basharat, A., Shah, M.: Chaotic invariants for human action recognition. In: Proc. ICCV (2007)
18. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR, pp. I: 886–893 (2005)
19. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Image Understanding Workshop, pp. 121–130 (1981)
20. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. PAMI 29, 2247–2253 (2007)
21. Laptev, I., Caputo, B., Schüldt, C., Lindeberg, T.: Local velocity-adapted motion events for spatio-temporal recognition. CVIU 108, 207–229 (2007)
22. Niebles, J., Wang, H., Li, F.: Unsupervised learning of human action categories using spatial-temporal words. In: Proc. BMVC (2006)
23. Marszałek, M., Schmid, C., Harzallah, H., van de Weijer, J.: Learning object representations for visual object class recognition. In: The PASCAL VOC 2007 Challenge Workshop, in conjunction with ICCV (2007)
24. Ikizler, N., Duygulu, P.: Human action recognition using distribution of oriented rectangular patches. In: Workshop on Human Motion, pp. 271–284 (2007)