Action Recognition using Rank-1 Approximation of Joint Self-Similarity Volume

Chuan Sun[1], Imran Junejo[2], Hassan Foroosh[1]
[1]Division of Computer Science, University of Central Florida, USA
[2]Department of Computer Science, University of Sharjah, United Arab Emirates
{csun[1],foroosh[1]}@cs.ucf.edu, {ijunejo[2]}@sharjah.ac.ae

Abstract

In this paper, we make three main contributions in the area of action recognition: (i) We introduce the concept of Joint Self-Similarity Volume (Joint SSV) for modeling dynamical systems, and show that by using a new optimized rank-1 tensor approximation of Joint SSV one can obtain compact low-dimensional descriptors that very accurately preserve the dynamics of the original system, e.g. an action video sequence; (ii) The descriptor vectors derived from the optimized rank-1 approximation make it possible to recognize actions without explicitly aligning the action sequences of varying speed of execution or different frame rates; (iii) The method is generic and can be applied using different low-level features such as silhouettes, histogram of oriented gradients, etc. Hence, it does not necessarily require explicit tracking of features in the space-time volume. Our experimental results on three public datasets demonstrate that our method produces remarkably good results and outperforms all baseline methods.

1. Introduction

Various approaches have been proposed over the years for action recognition. On the basis of representation, they can be categorized as: time evolution of human silhouettes [20], *action cylinders*, *space-time shapes* [22], and *local 3D patch analysis* [13], generally coupled with some machine learning techniques. Almost all the works mentioned above rely primarily on an effective feature extraction technique. These feature extraction methods can be roughly categorized into: *motion-based* [4], *appearance based* [6], *space-time volume* based [22], *space-time interest points or local features* based [14, 16], and the closely related methods to our approach that are based on the notion of self-similarity [1, 7].

Our framework is shown schematically in Fig.1. We construct a Self-Similarity Matrix (SSM) for each frame of the video sequence using a feature vector. We then construct Joint SSMs from this sequence of SSMs, leading to a Joint Self-Similarity Volume (Joint SSV). Joint SSV is then decomposed into its rank-1 approximation vectors using an optimized iterative tensor decomposition algorithm. This yields a set of compact vector descriptors that are highly discriminative between different actions. To evaluate our method on human action recognition, we used three public datasets. To show that our method is generic and does not depend on the input feature vector, we tested our method using low-level features like silhouette, as well as middle-level features like HOG3D. The final step used a nearest neighbor classification using the descriptor vectors produced by the rank-1 decomposition of Joint SSV.

The remainder of this paper is organized as follows: Section 2 presents some preliminaries on the SSM and the Joint SSM. Section 3 describes the construction of a Joint SSV, followed by an optimized rank-1 tensor decomposition algorithm in Section 4. Section 5 then describes the similarity measure used to classify actions. Experimental results and their analysis are presented in Sections 6 and 7.

2. Joint Self-similarity Matrix

Below are some preliminary results on SSM:

Definition 1: *An SSM can be expressed by a $N \times N$ matrix $R_{i,j}(\eta, v) = \Theta(\eta - \|v_i - v_j\|_p), i, j = 1, ..., N$, where N is the length of a feature vector v, and η is a threshold distance.*

The threshold η filters the values of each SSM element. We set $\eta = 0$ in this paper because this will give us a complete representation for the Joint SSMs. $\Theta(\cdot)$ can be the Heaviside function (i.e. $\theta(x) = 0$, if $x < 0$, and $\theta(x) = 1$ otherwise) and $\|\cdot\|$ is chosen as an ℓ_p-norm in this paper.

It can be verified that the SSM holds the following properties: $R_{i,j} = R_{j,i}$ (Symmetry); $R_{i,j} >= 0$ for all i and j (Positivity); and $R_{i,k} <= R_{i,j} + R_{j,k}$ for all i, j, k (Triangle inequality), and hence it is a metric. SSM provides important insights into the dynamics of a vector, which is especially advantageous in high dimensional spaces [1]. The intuition behind the SSM is that, according to recurrent plot theory, if we view the vector v as a trajectory in 2D space, the SSM itself captures the internal dynamics of this trajectory in a matrix form [2].

We further extend the SSM to Joint SSM based on the

2011 IEEE International Conference on Computer Vision
978-1-4577-1102-2/11/$26.00 ©2011 IEEE

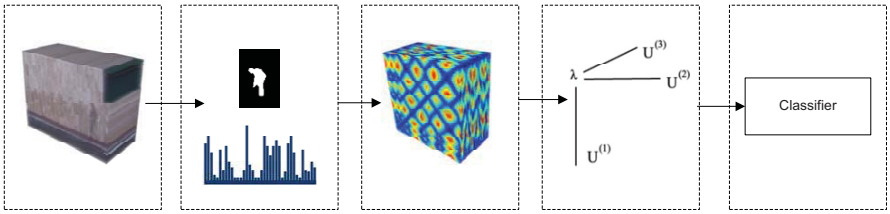

Figure 1: The flow of our action recognition framework. *First*, given an input action video, we extract either low-level features like silhouettes in a frame-by-frame manner, or middle-level features like HOG3D from the partitioned video blocks. *Second*, we transform the feature vector in each frame into an SSM. From the sequence of SSMs we then construct a symmetric and unique 3D structure, which we refer to as the Joint SSV. Joint SSV carries information about action dynamics. However, in order to handle its large dimension, it is decomposed into three compact and discriminative vectors, two of which are identical (due to symmetry). These descriptor vectors characterize the internal dynamics of an action. *Finally*, the vectors are used for measuring similarity with a database of actions for classification.

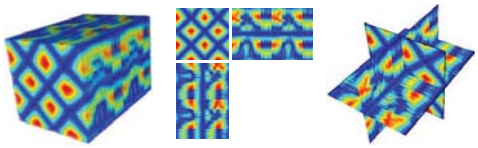

Figure 2: Visualization of the symmetric Joint SSV. The middle figure shows its cut in three direction. The right figure shows the X-section of the volume.

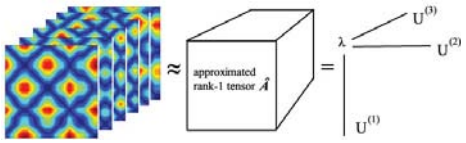

Figure 3: Rank-1 approximation $\hat{\mathcal{A}} = \lambda U^{(1)} \circ U^{(2)} \circ U^{(3)}$ for original Joint SSV $\mathcal{A}$.

idea of Joint Recurrence Plot (JPR) theory, which will be used in the construction of the Joint SSV.

Definition 2: *The Joint SSM is defined as* $JR^{v,w}_{i,j}(\eta_v, \eta_w, v, w) = \Theta(\eta^v - \|v_i - v_j\|_{p_1})\Theta(\eta^w - \|w_i - w_j\|_{p_2})$, *in which* $i, j = 1, ..., N$, η_v *and* η_w *are two internal thresholds,* p_1 *and* p_2 *are two distance norms.*

This extension is motivated by the fact that a recurrence will take place if a point v_j on the first trajectory v returns to the neighborhood of a former point v_i, and simultaneously a point w_j on the second trajectory w returns to the neighborhood of a former point w_i.

3. Construction of Joint SSV

Suppose we have vectors $\Psi = \{V_1, V_2, ..., V_t\}$ with $V_i \in R^d$. These vectors can be regarded as some specific feature vectors varying over time T, say, extracted features from video sequence. Our objective here is to build a unique volume that simultaneously characterizes the dynamics of not only each element of Ψ but also the relation amongst consecutive ones. Based on the recurrent plot theory, the Laplacian operator is applied on the SSM sequence to fuse the consecutive SSMs. We define the gradient operator ∇_t on Ψ as $\nabla_t \Psi = \frac{d\Psi}{dt} = V_i - V_{i-1}$. Since $\Gamma(\Psi) = \{\Gamma(V_i)\}_{i=1..t}$, we have $\nabla_t \Gamma(V_i) = \Gamma(V_i) - \Gamma(V_{i-1})$. It can be verified that $\Gamma(\nabla_t(V_i)) = \Gamma(V_i - V_{i-1}) = \nabla_t \Gamma(V_i)$. Therefore, $\Gamma\nabla_t^2(\Psi) = \nabla_t^2\Gamma(\Psi)$, and we can further arrive at the following theorem:

Theorem 1: *Given a random vector* Ψ *and a self-similarity matrix operator* $\Gamma : R^d \to R^{d\times d}$, *it holds that* $\Gamma\nabla_t^2(\Psi) = \nabla_t^2\Gamma(\Psi)$.

The self-similarity matrix operator Γ and the second order Laplacian operator ∇_t^2 are exchangeable in terms of the SSM sequence. Now we define the Joint Self-Similarity Volume based on Definition 2. Let $\circ$ be the element-wise multiplication operator between two matrices:

Definition 3: *The Joint Self-Similarity Volume corresponding to a random vector* Ψ *is built via a map* $\Xi : R^{d\times t} \to R^{d\times d\times t}$ *such that each element in* T *dimension is defined by a matrix satisfying* $\Xi_i|_{i=1..t} = \Gamma(\Psi_i) \circ \Gamma\nabla_t^2(\Psi_i)$.

This generates a symmetric 3D volume, that we refer to as the Joint SSV.

4. Rank-1 tensor approximation

To obtain an optimal rank-1 approximation of Joint SSV, we propose an alternating least-squares (ALS) method by optimizing the components of the factorization of a given SSV in an iterative fashion similar to [10, 11]. Given a real Nth-order tensor $\mathcal{A} \in \mathbb{R}^{I_1\times I_2\times\cdots\times I_N}$, there exists a scalar λ and N unit-norm vectors $U^{(1)}, U^{(2)}, \cdots, U^{(N)}$ such that a rank-1 tensor $\hat{\mathcal{A}} = \lambda U^{(1)} \circ U^{(1)} \circ \cdots \circ U^{(N)}$ minimizes the least-squares cost function

$$f(\hat{\mathcal{A}}) = \|\mathcal{A} - \hat{\mathcal{A}}\|_2$$

over the manifold of rank-1 tensors, which can be analyzed using the Lagrange multipliers and yields the following

equations [3]:

$$\mathcal{A} \times_1 U^{(1)^T} \cdots \times_{n-1} U^{(n-1)^T} \times_{n+1} U^{(n+1)^T} \cdots \times_N U^{(N)^T} = \lambda U^{(n)},$$
$$\mathcal{A} \times_1 U^{(1)^T} \times_2 U^{(2)^T} \cdots \times_N U^{(N)^T} = \lambda,$$
$$\|U^{(n)}\| = 1.$$

Specifically, our objective is to find a rank-1 approximation of Joint SSV such that there exists a scalar λ and three vectors $U^{(1)}, U^{(2)}$ and $U^{(3)}$ with objective function

$$min \sum_{i,j,k} (a_{ijk} - \lambda U_i^{(1)} \circ U_j^{(2)} \circ U_k^{(3)})^2, \qquad (1)$$

where a_{ijk} denotes the Joint SSV, a 3-order tensor, as shown in Fig.3. The $\circ$ is the outer product operator for vector, i and j are spatial mode indices and $i, j \in [1, I]$, I is the size of Joint SSM; while $k \in [1, K]$, K is the frame number for this Joint SSV. Since each vector $U^{(1)}, U^{(2)}$ and $U^{(3)}$ is determined only up to a scaling factor, we have

$$\|U^{(1)}\|_2 = \|U^{(2)}\|_2 = \|U^{(3)}\|_2 = 1.$$

On the other hand, Joint SSV is symmetric in spatial dimension since its elements remain constant under any permutation of the indices i and j, i.e. $a_{ijk} = a_{jik}$, therefore

$$U^{(1)} = U^{(2)}. \qquad (2)$$

For clarity of presentation, we denote $U^{(1)}, U^{(2)}$ and $U^{(3)}$ as ρ, ρ and ε, and we will call them the *primary vector* ρ, and *secondary vector* ε, respectively. Under the constraint of Eq.(2), the Eq.(1) can be solved by the technique of Generalized Rayleigh Quotient (GRQ) in [23], and we adopt the alternating least squares algorithm (ALS) in this paper for the optimal SSV approximation.

Algorithm 1: Joint SSV rank-1 approximation

input : A 3-order tensor Joint SSV $\mathcal{A} \in \mathbb{R}^{I \times I \times K}$, where I is the spatial dimension of Joint SSM, and K is the temporal dimension of $\mathcal{A}$

output: Two vectors ρ and ε that minimize $\|\mathcal{A} - \lambda \rho \circ \rho \circ \varepsilon\|_2$, where $\rho \in \mathbb{R}^I, \varepsilon \in \mathbb{R}^K$, and $\|\rho\|_2 = \|\varepsilon\|_2 = 1$

Initialize $U^0 = [\rho^{(0)}, \varepsilon^{(0)}]^T$;

for $t \leftarrow 0$ **to** $N_{maxiteration}$ **do**

$\widetilde{\rho}^{(t+1)} = \mathcal{A} \overline{\times}_2 \rho^{(t)} \overline{\times}_3 \varepsilon^{(t)}$;

$\widetilde{\varepsilon}^{(t+1)} = \mathcal{A} \overline{\times}_1 \rho^{(t)} \overline{\times}_2 \rho^{(t)}$;

$\rho^{(t+1)} = \widetilde{\rho}^{(t+1)} / \|\widetilde{\rho}^{(t+1)}\|$;

$\varepsilon^{(t+1)} = \widetilde{\varepsilon}^{(t+1)} / \|\widetilde{\varepsilon}^{(t+1)}\|$;

$\lambda^{(t+1)} = \mathcal{A} \overline{\times}_1 \rho^{(t+1)} \overline{\times}_2 \rho^{(t+1)} \overline{\times}_3 \varepsilon^{(t+1)}$;

end

In **Algorithm 1**, the $\overline{\times}_i$ for $i = 1, 2, 3$ denotes the multiplication between a tensor and a vector in mode-i of that tensor, whose result is also a tensor, namely,

$$\mathcal{B} = \mathcal{A} \overline{\times}_i \rho \Longleftrightarrow (\mathcal{B})_{jk} = \sum_{i=1}^{I} \mathcal{A}_{ijk} \rho_i.$$

Starting with random initial values for ρ and ε, the algorithm alternately changes ρ (or ε) while fixing the other one, and iteratively achieves the optimal approximation. The iteration stops when the difference between $\mathcal{A}$ and $\hat{\mathcal{A}}$ arrives at a sufficiently small value.

5. Similarity measure for classification

Let Ψ and Ψ' be the two initial input vectors, whose corresponding decomposed vector pairs are $\mathbf{v} = \{\rho, \rho, \varepsilon\}$ and $\mathbf{w} = \{\rho', \rho', \varepsilon'\}$, respectively. We first normalize ρ and ρ' (as well as ε and ε') to zero mean and unit variance, and make ρ and ρ' (as well as ε and ε') of equal dimension. The similarity between Ψ and Ψ' is then defined as

$$D(\Psi, \Psi') = \sum_{i=1}^{3} \max d(v_i, w_i),$$

where $d(v_i, w_i)$ denotes the cross-correlation of the i^{th} elements in $\mathbf{v}$ and $\mathbf{w}$.

6. Experiments

We evaluated our method on 3 well-known public datasets: Weizmann, KTH, and UCF sports dataset. Our goal was to evaluate the feasibility of our technique on various datasets with different Joint SSV schemes.

6.1. Two schemes

HOG3D-based Joint SSV (JSSV-hog3d) We employed the dense representation as in [20], and used the HOG3D descriptor [8] at densely distributed locations within a Region of Interest (ROI) centered around the actor, and partition the volume into regular overlapping *blocks*. All blocks were then partitioned into small regular *cells*. Histograms of 3D gradient orientations, generated using dodecahedron based quantization [8] with 6 orientation bins, for cells within a block, were then computed, and concatenated to form a block descriptor. Here we name all blocks within the same temporal location a *slice*, as shown in Fig.4.

We used the same configuration as in [20] for defining ROIs but different block setup. We used $2^\kappa \times 2^\kappa \times 2^\tau$ pixel blocks subdivided by $2 \times 2 \times 2$ cells, and computed the HOG3D descriptor for each block. Note that κ and τ are parameters that control the size of blocks. We let κ range from 2 to 4. Otherwise, the larger the κ is, the less the number of blocks for each slice will be, which may be disadvantageous for the computation of Joint SSVs. The τ

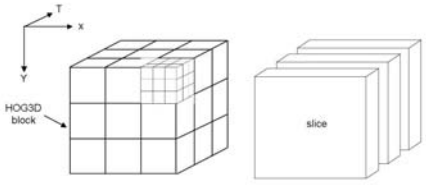

Figure 4: (Left) Extracting HOG3D feature descriptor after the dense sampling for the action volume in ROI and the partitioning of the volume into blocks; (Right) All blocks with the same temporal location form a *slice*. Each slice is further vectorized to a vector feeding into the Joint SSV construction procedure.

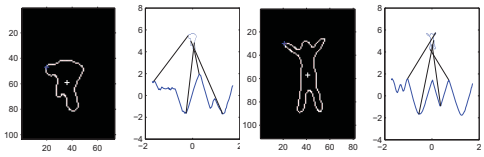

Figure 6: Converting silhouette features to time series using the method in [21] for *Bend* and *Jack* actions from the Weizmann dataset.

ranges from 1 to 5. It can control the depth of the generated volume.

Slices overlap with each other between consecutive ones, yielding a redundant representation, which enhances the discriminative power [20]. Within each slice, all blocks are concatenated in row order into a block sequence. This sequence is a vector used for building the self-similarity matrix. Using all slices, we then construct a Joint SSV out of both SSMs and Joint SSMs using the procedure described in Section 3.

Silhouette-based Joint SSV (JSSV-silh) We extracted the contour from the silhouette in each frame and transformed the contour into a time series using the method in [21], as shown in Fig.6. The time series were normalized to zero mean and unit variance before being fed into the framework as input vectors to generate the Joint SSV. Silhouettes can be easily extracted from static or uniform action background, but harder or even impractical for more challenging action sequences. For this reason, we merely tested this scheme on Weizmann dataset, which provides well-extracted silhouette features. Fig.8 shows a sequence of generated SSMs for *Bend* action in Weizmann dataset, and Fig.7 shows the visual difference between four different actions.

6.2. Datasets and recognition rate

For all the results reported in this section, we performed the recognition using nearest neighbor classification and leave-one-out cross validation.

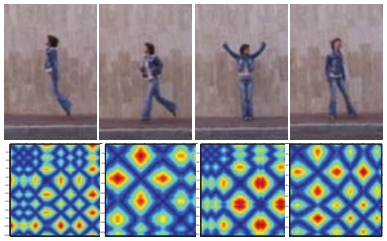

Figure 7: SSM comparison among various action frames using silhouette feature. (Top) Selected frames from 4 actions *Jack*, *Run*, *Wave2*, and *Side*; (Bottom) The corresponding silhouette-based SSMs.

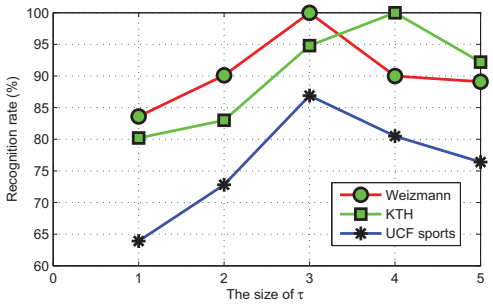

Figure 9: Recognition rate under different HOG3D block depth 2^{τ} for three datasets using JSSV-hog3d scheme.

Weizmann dataset The Weizmann dataset [1] consists of videos of 10 different actions performed by 9 actors. Each video clip contains one subject performing a single action. The 10 different action categories: *walking, running, jumping, gallop sideways, bending, one-hand-waving, two-hands-waving, jumping in place, jumping jack*, and *skipping*. Each of the clip lasts about 2 seconds at 25Hz with image frame size of 180×144.

We evaluated two schemes separately, namely the JSSV-silh and the JSSV-hog3d. For the former scheme, we used the provided well-extracted silhouettes in dataset to build input vectors for the whole framework, and we were able to achieve a recognition rate of 100%. For the latter one, we extract the ROI using the silhouettes and fitting a bounding box around each of them. To be consistent, all ROIs in our experiments are scaled and concatenated to form a $128 \times 64 \times t$ volume, where t is the frame number in sequence. We evaluated various block size setups (Fig.9 and Table 1) and observed that when $\kappa = 4$ and $\tau = 3$ (i.e. block size: $16 \times 16 \times 8$), the JSSV-hog3d scheme yields the best recognition rate of 100%, as shown in Table 1.

KTH dataset The KTH dataset [2] consists of 6 actions performed by 25 actors in four different scenarios. We followed the evaluation procedure in [20] but used slightly different settings for the block size. We extracted the ROIs

[1] http://www.wisdom.weizmann.ac.il/ vision/SpaceTimeActions.html
[2] http://www.nada.kth.se/cvap/actions/

Figure 5: Screenshots for different action classes of 3 public datasets. (The 1st row) The Weizmann dataset and the KTH dataset; (The 2nd row) The UCF sports dataset.

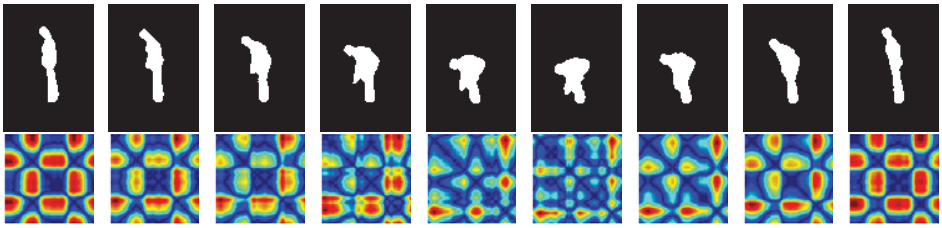

Figure 8: A sequence of computed SSMs for frames selected from the the *Bend* action in Weizmann dataset. Note that all above SSMs have identical dimension. Both salient and subtle differences between silhouette contours can be revealed by SSMs.

using the bounding boxes provided by [14], and evaluated the JSSV-hog3d scheme on this dataset under various block size configurations, as shown in Fig.9. When τ is small, the block depth is small, making the final decomposed vectors undiscriminating for classification. But as τ grows, the recognition rate grows accordingly. This also agrees with our intuition that larger blocks contain more cells, and capture more stable gradient information compared with the smaller ones. But as the block size becomes too large, more redundant information is introduced, leading a reduced recognition rate. Especially, our best recognition rate 100% is achieved when $\kappa = 4$ and $\tau = 4$, This outperforms both the result in [20] (92.4%), which has a similar experimental configuration as us, and the state-of-the-art approach in [5] (94.5%).

UCF sports dataset The UCF sports dataset [3] contains 11 actions: *golf swing (back, front, side), kicking (front, side), riding horse, run, skate boarding, swing bench, swing (side), and walk*. This dataset also provides the well-extracted bounding boxes for extracting the ROIs from each action sequence. Each action contains unequal number of samples. For consistency in our experiments, we chose 10 samples for each action class. For those actions having less than 10 samples such as "golf-swing-back", "golf-swing-side", and "golf-swing-front", we increased the amount of data samples by adding a horizontally flipped version of existing samples. This resulted in 110 samples in total. As shown in Table 1, our best recognition rate 86.9% is achieved when $\kappa = 3$ and $\tau = 3$, which is comparable with the state-of-the-art in [9] (87.27%).

[3] http://server.cs.ucf.edu/ vision/data.html

7. Conclusion

In this paper, we study the application of Joint Self-Similarity Volume for action recognition in video sequences. A new optimized rank-1 tensor approximation algorithm is proposed for dimensionality reduction, which can largely preserves the salient characteristics for scene dynamics. A significant saving in both memory and computational complexity can be achieved since only a collection of rank-1 tensors is adopted as the reference database. The algorithm also allows one to recognize actions without explicitly aligning the videos in temporal dimension. Due to the fact that the proposed formulation is not dependent on the low-level features extracted from the sequence, we can apply this framework using any type of low-level feature vector, including feature vectors that are view-invariant [18].

Table 1: (Upper table) Comparison of recognition rate for 3 action datasets under the JSSV-hog3d scheme. The κ and τ are parameters controlling the block size $2^\kappa \times 2^\kappa \times 2^\tau$; (Bottom table) Comparison of recognition rate for 3 datasets between our 2 different schemes and other methods.

	Weizmann					KTH					UCF sports				
	$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$	$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$	$\tau=1$	$\tau=2$	$\tau=3$	$\tau=4$	$\tau=5$
κ=2	70.5	78.4	82.0	80.1	71.1	70.5	73.9	64.8	76.3	60.2	68.0	64.7	69.8	72.0	48.6
κ=3	75.0	76.5	86.2	87.3	85.4	75.5	70.4	84.8	88.0	84.0	63.9	72.8	**86.9**	80.5	76.4
κ=4	83.6	90.1	**100.0**	90.0	89.1	80.2	83.0	94.8	**100.0**	92.2	68.2	81.9	64.8	77.5	76.1

Methods	Weizmann	KTH	UCF sports
JSSV-silh	**100.0**	-	-
JSSV-hog3d	**100.0**	**100.0**	**86.9**
Other methods	Schindler [17] 100.0	Gilbert [5] 94.5	Kovashka [9] 87.27
	Zhang [24] 97.8	Lin [14] 93.4	Klaser [12] 86.7
	J.Imran [7] 95.3	Schindler [17] 92.7	Wang [19] 85.6
	Niebles [16] 90.0	Weinland [20] 92.4	
	Liu [15] 89.3	Liu [15] 82.8	

References

[1] C. Benabdelkader, R. Cutler, and L. Davis. Gait recognition using image self-similarity. *EURASIP JASP*, pages 572–585, 2004. 1

[2] R. Blasco and M. Carmen. *Synchronization analysis by means of recurrences in phase space*. PhD thesis, Universitat sbibliothek, 2004. 1

[3] L. De Lathauwer, B. De Moor, and J. Vandewalle. On the Best Rank-1 and Rank-(R˜ 1, R˜ 2,..., R˜ N) Approximation of Higher-Order Tensors. *SIAM Journal on Matrix Analysis and Applications*, 21(4):1324–1342, 2000. 3

[4] A. A. Efros, A. C. Berg, E. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003. 1

[5] A. Gilbert, J. Illingworth, and R. Bowden. Fast realistic multi-action recognition using mined dense spatio-temporal features. In *ICCV*, pages 925–931, 2010. 5, 6

[6] M. Grundmann, F. Meier, and I. Essa. 3d shape context and distance transform for action recognition. In *Proc. ICPR*, pages 1–4, 2008. 1

[7] I. N. Junejo, E. Dexter, I. Laptev, and P. Perez. View-independent action recognition from temporal self-similarities. *IEEE Trans. PAMI*, 99, 2010. 1, 6

[8] A. Klaser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3D-gradients. In *British Machine Vision Conference*, pages 995–1004. Citeseer, 2008. 3

[9] A. Kovashka and K. Grauman. Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. 2010. 5, 6

[10] P. Kroonenberg. *Three-mode principal component analysis: Theory and applications*. DSWO press, 1983. 2

[11] P. Kroonenberg and J. De Leeuw. Principal component analysis of three-mode data by means of alternating least squares algorithms. *Psychometrika*, 45(1):69–97, 1980. 2

[12] A. Laptev and C. Schmid. Will person detection help bag-of-features action recognition? 2010. 6

[13] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. CVPR*, 2008. 1

[14] Z. Lin, Z. Jiang, and L. Davis. Recognizing actions by shape-motion prototype trees. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 444–451. IEEE, 2010. 1, 5, 6

[15] J. Liu and M. Shah. Learning human actions via information maximization. In *CVPR*, pages 1–8, 2008. 6

[16] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. *IJCV*, 79(3):299–318, 2008. 1, 6

[17] K. Schindler and L. Van Gool. Action Snippets: How many frames does human action recognition require? In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008. 6

[18] Y. Shen and H. Foroosh. View-invariant action recognition from point triplets. *IEEE transactions PAMI*, pages 1898–1905, 2009. 5

[19] H. Wang, M. Ullah, A. Klaser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. 2009. 6

[20] D. Weinland, M. Ozuysal, and P. Fua. Making Action Recognition Robust to Occlusions and Viewpoint Changes. *Proc. ECCV*, pages 635–648, 2010. 1, 3, 4, 5, 6

[21] L. Ye and E. Keogh. Time series shapelets: a new primitive for data mining. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 947–956. ACM, 2009. 4

[22] A. Yilmaz and M. Shah. Actions sketch: A novel action representation. In *Proc. CVPR*, pages I:984–989, 2005. 1

[23] T. Zhang and G. Golub. Rank-1 approximation of higher-order tensors. *SIAM J. Matrix Anal. Appl*, 23(534-550):4, 2001. 3

[24] Z. Zhang, Y. Hu, S. Chan, and L. Chia. Motion context: A new representation for human action recognition. *ECCV*, pages 817–829, 2008. 6