Action Recognition from One Example

Hae Jong Seo, *Student Member*, *IEEE*, and Peyman Milanfar, *Fellow*, *IEEE*

Abstract—We present a novel action recognition method based on space-time locally adaptive regression kernels and the matrix cosine similarity measure. The proposed method uses a single example of an action as a query to find similar matches. It does not require prior knowledge about actions, foreground/background segmentation, or any motion estimation or tracking. Our method is based on the computation of novel space-time descriptors from the query video which measure the likeness of a voxel to its surroundings. Salient features are extracted from said descriptors and compared against analogous features from the target video. This comparison is done using a matrix generalization of the cosine similarity measure. The algorithm yields a scalar resemblance volume, with each voxel indicating the likelihood of similarity between the query video and all cubes in the target video. Using nonparametric significance tests by controlling the false discovery rate, we detect the presence and location of actions similar to the query video. High performance is demonstrated on challenging sets of action data containing fast motions, varied contexts, and complicated background. Further experiments on the Weizmann and KTH data sets demonstrate state-of-the-art performance in action categorization.

Index Terms—Action recognition, space-time descriptor, correlation, regression analysis.

1 Introduction

A huge number of videos (e.g., BBC[1] and Youtube[2]) are available online today and the number is rapidly growing. Human actions constitute one of the most important parts in movies, TV shows, and consumer-generated videos. Analysis of human actions in videos is considered a very important problem in computer vision because of such applications as human-computer interaction, content-based video retrieval, visual surveillance, analysis of sports events, and more. The term "action" refers to a simple motion pattern as performed by a single subject, and in general lasts only for a short period of time, namely, just a few seconds. *Action* is often distinguished from *activity* in the sense that action is an individual atomic unit of activity. In particular, human action refers to physical body motion. Recognizing human actions from video is a very challenging problem due to the fact that physical body motion can look very different depending on the context. For instance, similar actions with different clothes or in different illumination and background can result in a large appearance variation, or the same action performed by two different people may look quite dissimilar in many ways.

1.1 Problem Specification

We present a novel approach to the problem of human action recognition as a video-to-video matching problem. Here, recognition is generally divided into two parts: category classification and detection/localization. The goal of action classification is to classify a given action query into one of several prespecified categories (for instance, six categories from the KTH action data set [1]: boxing, hand clapping, hand waving, jogging, running, and walking). Meanwhile, action detection is meant to separate an action of interest from the background in a target video (for instance, spatiotemporal localization of a walking person). This paper tackles both action detection and category classification problems simultaneously by searching for an action of interest within other "target" videos with only a *single* "query" video. We focus on a sophisticated feature representation with an efficient and reliable similarity measure, which also allows us to avoid the difficult problem of explicit motion estimation.[3]

In general, the target video may contain actions similar to the query, but these will typically appear in completely different context (see Fig. 1a) Examples of such differences can range from rather simple optical or geometric differences (such as different clothes, lighting, action speed, scale, and view changes) to more complex inherent structural differences, such as a hand-drawn action video clip (e.g., animation) rather than a real human action.

1.2 Related Work

Over the past two decades, many studies have attempted to tackle this problem and made impressive progress. Approaches can be categorized on the basis of *action representation*, namely, appearance-based representation [2], [3], [4], [5], shape-based representation [6], [7], [8], [9], optical-flow-based representation [10], [11], [12], [13], interest-point-based representation [1], [14], [15], [16], [17], [18], and volume-based representation [19], [20], [21], [22], [23], [24], [25]. We refer the interested reader to [26], [27], [28] and references therein for a good summary.

[1] http://www.bbcmotiongallery.com.
[2] http://www.youtube.com.

[3] Project website: http://users.soe.ucsc.edu/~rokaf/ActionRecognitionFromSingleExample.html.

• *The authors are with the University of California Santa Cruz, 1156 High Street, Mailcode SOE2, Santa Cruz, CA 95064.*
E-mail: rokaf@soe.ucsc.edu, milanfar@ee.ucsc.edu.

Manuscript received 26 Aug. 2009; revised 3 Mar. 2010; accepted 26 June 2010; published online 17 Aug. 2010.
Recommended for acceptance by B. Schiele.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2009-08-0558.
Digital Object Identifier no. 10.1109/TPAMI.2010.156.

0162-8828/11/$26.00 © 2011 IEEE Published by the IEEE Computer Society

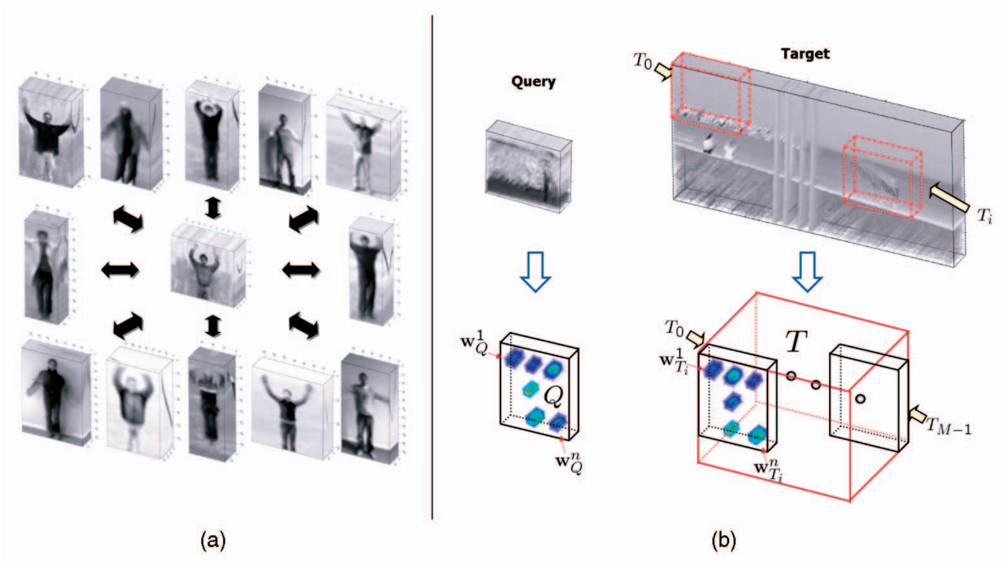


Fig. 1. (a) A hand-waving action and possibly similar actions. (b) The action detection problem. Given a query video Q, we wish to detect/localize actions of interest in a target video T. T is divided into a set of overlapping cubes. 3D LSKs capture the space-time geometric structure of underlying data.

As examples of the interest point-based approach, which has gained a lot of interest, Niebles et al. [15], [14] considered videos as spatiotemporal bag-of-words by extracting space-time interest points and clustering the features, and then used a probabilistic Latent Semantic Analysis (pLSA) model to localize and categorize human actions. Yuan et al. [29] also used spatiotemporal features as proposed by [16]. They extended the naive Bayes nearest neighbor (NN) classifier [30], which was developed for object recognition, to action recognition. By modifying the efficient searching method based on branch-and-bound [31] for the 3D case, they provided a very fast action detection method. However, the performance of these methods can degrade due to 1) the lack of enough training samples and 2) misdetections and occlusions of the interest points since they ignore global space-time information.

Shechtman and Irani [22] employed a 3D correlation scheme for action detection. They focused on subvolume matching in order to find similar motion between the two space-time volumes, which can be computationally heavy. Ke et al. [23] presented an approach which uses boosting on 3D Haar-type features inspired by similar features in 2D object detection [32]. While these features are very efficient to compute, many examples are required to train an action detector in order to achieve good performance. They further proposed a part-based shape and flow matching framework [33] and showed good action detection performance in crowded videos. Recently, Kim and Cipolla [24] generalized canonical correlation analysis to tensors and showed very good accuracy on the KTH action data set, but their method requires a manual alignment process for camera motion compensation. Ning et al. [25] proposed a system to search for human actions using a coarse-to-fine approach with a five-layer hierarchical space-time model. These volumetric methods do not require background subtraction, motion estimation, or complex models of body configuration and kinematics. They tolerate variations in appearance, scale, rotation, and movement to some extent.

As opposed to 2D object recognition, which has recently proven capable of learning a respectably large number of categories (a couple of hundred), action recognition is still only limited to about a dozen categories at best (6 for the KTH, 10 for the Weizmann, and 12 for the Hollywood2 action data sets). Even though learning-based action recognition methods appear to be practical in a small number of categories, they have not yet proven to be scalable with a larger number of categories.[4] Due to the advent of large database-driven nonparametric approaches [34], [35], [36], instead of training sophisticated parametric models, we can reduce the inference problem to matching a query to an existing set of annotated databases, posing a video-to-video matching problem. As a successful example, Boiman et al. [30] showed that a rather simple NN-based image classifier in the space of the local image descriptors is efficient and even outperforms the leading learning-based image classifiers, such as SVM-KNN [37] and pyramid match kernel [38].

Methods such as those in [33], [22], [25], [39], and [40], which aim at recognizing actions based solely on one query, are very useful for applications such as video retrieval from the Web (e.g., viewdle[5] and videosurf[6]). In these methods, a single query video is provided by users and every gallery video in the database is compared with the given query.

1.3 Overview of the Proposed Approach

In this paper, our contributions to the action recognition task are mainly two-fold. First, we propose a novel feature representation that is derived from space-time local (steering) regression kernels (3D LSKs), which capture the underlying structure of the data quite well, even in the presence of significant distortions and data uncertainty. In

[4] The heavy computational complexity of action recognition methods compared to object recognition is a possible reason, but the lack of large action recognition data sets covering many categories is the major impediment.

[5] http://www.viewdle.com.

[6] http://www.videosurf.com.

fact, 3D LSKs measure the likeness of a voxel to its surroundings based on computation of a distance between points measured (along the shortest path) on a manifold[7] defined by the embedding of the video data in 4D as $[x_1, x_2, t, z(x_1, x_2, t)]$. Second, we generalize a training-free nonparametric detection scheme to 3D, which we developed earlier for 2D object detection [41]. We report state-of-the-art performance on action category classification by using the resulting nearest neighbor classifier. In order to achieve better classification performance, we apply space-time saliency detection [42] to larger videos in order to automatically crop to a short action clip.

We propose to use 3D LSKs for the problems of detection/localization of actions of interest between a query video and a target video, as nicely formulated in [22] and also addressed in [40]. The key idea behind 3D LSKs is to robustly obtain local space-time geometric structures by analyzing the photometric (voxel value) differences based on estimated space-time gradients, and use this structure information to determine the shape and size of a canonical kernel (descriptor). The motivation to use these 3D LSKs is the earlier successful work on adaptive kernel regression for image denoising, interpolation [43], deblurring [44], and superresolution [45]. The 3D LSKs implicitly contain information about the local motion of the voxels across time, thus requiring no explicit motion estimation.

Referring to Fig. 2, by denoting the target video (T) and the query video (Q), we compute a dense set of 3D LSKs from each. These densely computed descriptors are highly informative, but, taken together, tend to be overcomplete (redundant). Therefore, we derive features by applying dimensionality reduction (namely, Principal Component Analysis (PCA)) to these resulting arrays in order to retain the most salient characteristics of the 3D LSKs. The feature collections from Q and T_i (a chunk of the target, which is the same size as the query; see Fig. 1b) form feature volumes $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$. We compare the feature volumes $\mathbf{F}_{T_i}$ and $\mathbf{F}_Q$ from the ith cube of T and Q to look for matches. Inspired in part by many studies [46], [47], [48], [49], [50] which took advantage of cosine similarity over the conventional euclidean distance, we employ *Matrix Cosine Similarity* (MCS) as a similarity measure which generalizes the notion of cosine similarity between two vectors [51], [52], [53]. The optimality properties of this approach are described in [41] within a naive Bayes framework.

In general, it is assumed that the query video is smaller than the target video. However, this is not true in practice and a query video may indeed include a complex background, which deteriorates recognition accuracy. In order to deal with this problem, it is necessary to have a procedure which automatically segments from the query video a small cube that only contains a valid human action. For this we employ space-time saliency detection [42]. This idea not only allows us to extend the proposed detection framework to action category classification but also to improve both detection and classification accuracy by automatically removing irrelevant background from the query video. Fig. 2 shows an overview of our proposed framework for action detection.

Shechtman and Irani [54] introduced a space-time local self-similarity descriptor for action detection and showed performance improvement over related earlier approaches such as [22]. It is worth mentioning that this (independently derived) local space-time self-similarity descriptor is a special case of 3D LSK and is also related to a number of other local data adaptive metrics, such as Optimal Space-Time Adaptation (OSTA) [55] and Nonlocal Means (NLMs) [56], which have been used very successfully for video restoration in the image processing community. A related, but different, temporal self-similarity-based descriptor [57] has been proposed for view-independent action recognition which shows good performance on action data sets, such as the Weizmann [9] and the IXMAS [58], but they were not developed for the action localization task.

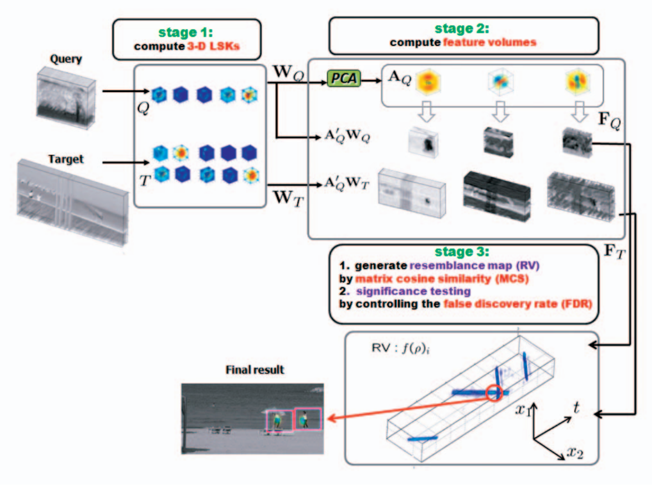


Fig. 2. System overview of action detection framework. (Broadly, there are three stages.)

Another related action representation, Ali and Shah [12] very recently proposed kinematic features (divergence, vorticity, symmetric, and antisymmetric optical flow and so forth) based on optical flows. By applying PCA to these features, they extracted dominant kinematic features and used them for action recognition, along with the multiple instance learning approach [59]. Our action representation is somewhat similar to theirs in the sense that we both use PCA to extract feature sets, but their method depends strongly on the number of (both positive and negative) training examples and explicitly estimates motion flows, while our method uses a single query (positive example) for localization and our descriptors implicitly contain both shape and flow information at the same time. Very recently, [40] also made use of motion descriptors based on optical flows and focused on learning a distance function, which is transferable to unseen action classes.

The proposed action detection method is distinguished from our earlier 2D work in [41], proposed for object detection, in the following respects:

1. action detection addressed in this paper is more challenging than static (2D) object detection due to additional problems such as variations in individual motion and camera motion;
2. we use space-time local steering kernels, which capture both *spatial* and *temporal* geometric structure;

[7] See Section 2.1.2 for details.

3. while [41] assumed that a query image is always smaller than a target and only contains an object of interest, we relax this assumption to deal with more realistic scenarios by incorporating space-time saliency detection [42]; and
4. while [41] focused on detection tasks, in this paper, we further achieved state-of-the-art action *classification* performance as well as high-detection accuracy.

A preliminary version of this paper appeared in the IEEE International Conference on Computer Vision (ICCV '09) [60]. This paper is different from [60] in the following respects:

1. we provide more detailed description about what the proposed descriptors capture from video data;
2. we show that 3D LSKs outperform simple linear 3D Gabor filters and a state-of-the-art 3D descriptor called "HOG3D [61]" in our action detection framework by providing both quantitative and qualitative comparisons in Section 3.1.1;
3. a multiscale approach is implemented to deal with large variations in scale of actions and is shown to outperform single-scale version in Section 3.1.1; and
4. we test our method on more complicated and challenging data set [62] for action localization.

2 Technical Details

As outlined in the previous section, our approach to detecting actions broadly consists of three stages (see Fig. 2) Below, we describe each of these steps in detail. In order to make the concepts more clear, we first briefly describe the local steering kernels in 2D. For extensive detail on this subject, we refer the reader to [43], [41].

2.1 Local Steering Kernel as a Descriptor

2.1.1 Local Steering Kernel in 2D

The key idea behind LSK is to robustly obtain the local structure of images by analyzing the photometric (pixel value) differences based on estimated gradients and to use this structure information to determine the shape and size of a canonical kernel. The local steering kernel is defined as follows:

$$K(\mathbf{x}_l - \mathbf{x}_i) = \sqrt{\det(\mathbf{C}_l)} \exp\left\{ \frac{(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i)}{-2h^2} \right\}, \quad (1)$$

where $\mathbf{x}_i = [x_1, x_2]_i^T$ is a pixel of interest, $l = 1, \ldots, P$, $\mathbf{x}_l = [x_1, x_2]_l^T$ are local neighboring pixels, h is a global smoothing parameter, P is the total number of samples in a local analysis window around a sample position at $\mathbf{x}_i$, and the matrix $\mathbf{C}_l \in \mathbb{R}^{(2\times2)}$ is a covariance matrix estimated from a collection of first derivatives along spatial axes. More specifically, the covariance matrix $\mathbf{C}_l$ can be first naively estimated as $\mathbf{J}_l^T \mathbf{J}_l$ with

$$\mathbf{J}_l = \begin{bmatrix} \vdots & \vdots \\ z_{x_1}(\mathbf{x}_k), & z_{x_2}(\mathbf{x}_k) \\ \vdots & \vdots \end{bmatrix}, \quad k \in \Omega_l, \quad (2)$$

where $z_{x_1}(\cdot)$ and $z_{x_2}(\cdot)$ are the first derivatives along the x_1, and x_2-axes and Ω_l is a local analysis window centered at $\mathbf{x}_l$.

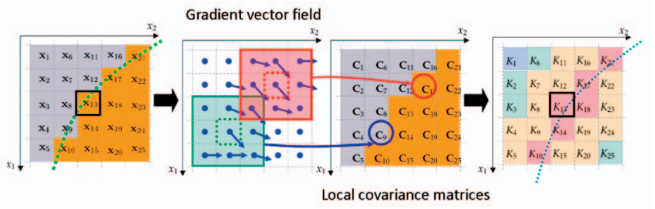


Fig. 3. Graphical description of how LSK values centered at pixel of interest $\mathbf{x}_{13}$ are computed in an edge region. Note that each pixel location has its own $\mathbf{C}_l \in \mathbb{R}^{2\times2}$ computed from the gradient vector field within a local window Ω_l (see green and red boxes). In K values, red means higher values (higher similarity).

Fig. 3 illustrates how the covariance matrices and respective LSK values are computed.

At this point, it is useful to provide the reader with an interpretation of the information captured and represented by the LSK descriptors. Specifically, in order to measure the similarity of two pixels, in general we can naturally consider both the spatial distance and the gray level distance (see Fig. 4). An effective way to combine these distances is to define a "signal-induced" distance or "Riemannian metric" [64], which basically stands for a distance between the points measured along the shortest path on the signal manifold. We can rewrite the matrix $\mathbf{C}_l$ as follows:

$$\mathbf{C}_l = \sum_{k\in\Omega_l} \begin{bmatrix} z_{x_1}^2(\mathbf{x}_k) & z_{x_1}(\mathbf{x}_k) z_{x_2}(\mathbf{x}_k) \\ z_{x_1}(\mathbf{x}_k) z_{x_2}(\mathbf{x}_k) & z_{x_2}^2(\mathbf{x}_k) \end{bmatrix}. \quad (3)$$

Then, the term $(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i)$ in (1) is closely related to the Riemannian metric as follows:

$$\begin{aligned}(\mathbf{x}_l - \mathbf{x}_i)^T \mathbf{C}_l (\mathbf{x}_l - \mathbf{x}_i) + (dx_1)_l^2 + (dx_2)_l^2 \\ = \sum_{k\in\Omega_l} z_{x_1}^2(\mathbf{x}_k)(dx_1)_l^2 + 2 z_{x_1}(\mathbf{x}_k) z_{x_2}(\mathbf{x}_k)(dx_1)_l(dx_2)_l \\ + z_{x_2}^2(\mathbf{x}_k)(dx_2)_l^2 + (dx_1)_l^2 + (dx_2)_l^2,\end{aligned} \quad (4)$$

where $(dx_1)_l = (x_1)_l - (x_1)_i$ and $(dx_2)_l = (x_2)_l - (x_2)_l$. See the Appendix for details.

For the sake of robustness, we compute a more stable estimate of $\mathbf{C}_l$ by invoking the singular value decomposition (SVD) of $\mathbf{J}_l$ with regularization as [43], [41]

$$\mathbf{C}_l = \gamma \sum_{q=1}^{2} a_q^2 \mathbf{v}_q \mathbf{v}_q^T \in \mathbb{R}^{(2\times2)}, \quad (5)$$

with

$$a_1 = \frac{s_1 + \lambda'}{s_2 + \lambda'}, \quad a_2 = \frac{s_2 + \lambda'}{s_1 + \lambda'}, \quad \gamma = \left(\frac{s_1 s_2 + \lambda''}{P} \right)^{\alpha}, \quad (6)$$

where λ' and λ'' are parameters[8] that dampen the noise effect and keep the denominators of a_qs from being zero, and α is a parameter[9] that restricts γ. The singular values (s_1, s_2) and the singular vectors $(\mathbf{v}_1, \mathbf{v}_2)$ are given by the compact SVD of $\mathbf{J}_l = \mathbf{U}_l \mathbf{S}_l \mathbf{V}_l^T = \mathbf{U}_l \mathrm{diag}[s_1, s_2]_l [\mathbf{v}_1, \mathbf{v}_2,]_l^T$. Note that $\sqrt{\det(\mathbf{C}_l)}$ in (1) plays a role as a general edge or corner indicator, thus giving higher weight to corresponding

[8] λ' and λ'' are set to 1 and 10^{-8}, respectively, and they are fixed for all experiments.

[9] α is set to 0.29 and fixed for all experiments.

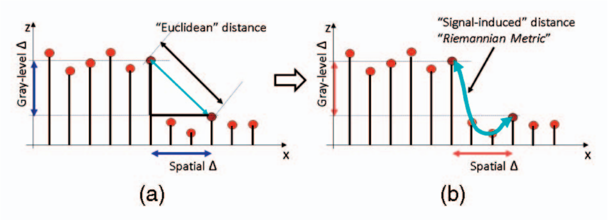


Fig. 4. (b) LSK captures distance between points measured along the shortest path on the image manifold, whereas (a) bilateral kernel [63], nonlocal means kernel [56], and self-similarity kernel [54] are based on simple euclidean distance.

pixels. Since we use a robust estimate of $\mathbf{C}_l$, the LSKs reliably capture local geometry of the data manifold even in the presence of noise. The shape of the LSKs is not simply a Gaussian, despite the simple definition in (1) above. It is important to note that this is because, for each pixel $\mathbf{x}_l$ in the vicinity of $\mathbf{x}_i$, a different matrix $\mathbf{C}_l$ is used, therefore leading to a far more complex and rich set of possible shapes for the resulting LSKs. Therefore, the LSKs can capture more sophisticated local geometry than histogram of gradients-based descriptors, such as SIFT and HOG, which use locally quantized gradients information. The key idea explained above is equally valid in 3D, as we describe below.

2.1.2 Space-Time Local Steering Kernel (3D)

Now, we introduce the time axis to the data model so that $\mathbf{x}_l = [x_1, x_2, t]_l^T$: x_1 and x_2 are the spatial coordinates and t is the temporal coordinate. Similarly to the 2D case, the covariance matrix $\mathbf{C}_l$ can be naively estimated as $\mathbf{J}_l^T\mathbf{J}_l$ with

$$\mathbf{J}_l = \begin{bmatrix} \vdots & \vdots & \vdots \\ z_{x_1}(\mathbf{x}_k), & z_{x_2}(\mathbf{x}_k), & z_t(\mathbf{x}_k) \\ \vdots & \vdots & \vdots \end{bmatrix}, \quad k \in \Omega_l, \tag{7}$$

where $z_{x_1}(\cdot)$, $z_{x_2}(\cdot)$, and $z_t(\cdot)$ are the first derivatives along the x_1, x_2, and t-axes, and Ω_l is a *space-time* local analysis window (or cube) around a sample position at $\mathbf{x}_l$.

As explained in the 2D LSK case, the term $(\mathbf{x}_l - \mathbf{x}_i)^T\mathbf{C}_l(\mathbf{x}_l - \mathbf{x}_i)$ in (1) now captures the distance between the voxels measured along the shortest path on the embedded manifold of the video data. Fig. 5 illustrates how 3D LSKs are computed in a space-time region. Again, $\mathbf{C}_l$ is estimated by invoking the SVD of $\mathbf{J}_l$ with regularization as follows [45]:

$$\mathbf{C}_l = \gamma \sum_{q=1}^{3} a_q^2 \mathbf{v}_q \mathbf{v}_q^T \in \mathbb{R}^{(3\times3)}, \tag{8}$$

with

$$a_1 = \frac{s_1 + \lambda'}{\sqrt{s_2 s_3} + \lambda'}, \quad a_2 = \frac{s_2 + \lambda'}{\sqrt{s_1 s_3} + \lambda'},$$
$$a_3 = \frac{s_3 + \lambda'}{\sqrt{s_1 s_2} + \lambda'}, \quad \gamma = \left(\frac{s_1 s_2 s_3 + \lambda''}{P}\right)^{\alpha}, \tag{9}$$

where λ' and λ'' are parameters[10] that dampen the noise effect and restrict γ and the denominators of a_qs from being zero. As mentioned earlier, the singular values (s_1, s_2, and s_3) and the singular vectors ($\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$) are given by the compact SVD of $\mathbf{J}_l = \mathbf{U}_l\mathbf{S}_l\mathbf{V}_l^T = \mathbf{U}_l\mathrm{diag}[s_1, s_2, s_3]_l[\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3]_l^T$.

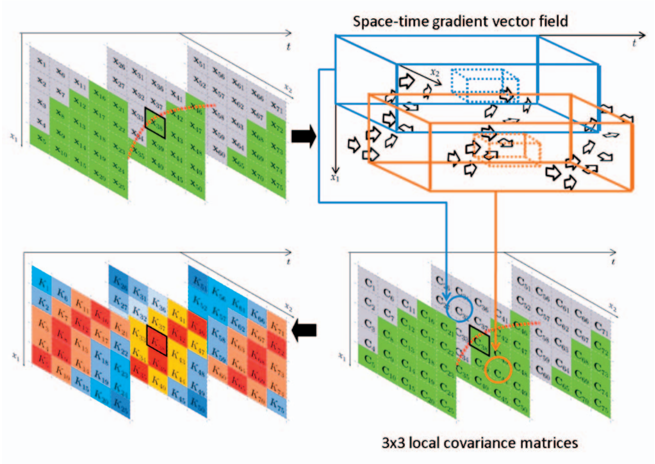

Space-time gradient vector field
3x3 local covariance matrices
t
x1
x2

Fig. 5. Graphical description of how 3D LSK values centered at voxel of interest $\mathbf{x}_{38}$ are computed in a space-time edge region. Note that each voxel location has its own $\mathbf{C}_l \in \mathbb{R}^{3\times3}$ computed from the space-time gradient vector field within a local space-time window.

In the 3D case, orientation information captured in 3D LSK contains the motion information implicitly [45]. It is worth noting that a significant strength of using this implicit framework (as opposed to the direct use of estimated motion vectors) is the flexibility it provides in terms of smoothly and adaptively changing descriptors. This flexibility allows the accommodation of even complex motions so long as their magnitudes are not excessively large.[11]

Fig. 6 shows examples of 3D local steering kernels capturing 3D local underlying geometric structure in various space-time regions. As can be seen in (1), the values of the kernel K are based on the covariance matrices $\mathbf{C}_l$ along with their space-time locations $\mathbf{x}_l$. Intuitively, $\mathbf{C}_l$s computed from the local analysis window Ω_l are similar to one another in the motion-free region (see Fig. 6 [1]). On the other hand, in the region where motion exists (see Fig. 6 [2], [3], [4], [5]), the kernel size and shape depend on both $\mathbf{C}_l$ and its space-time location $\mathbf{x}_l$ in the local space-time window. Thus, if the pixel of interest (center pixel of kernel) is located in space-time edge region, high values in the kernel are yielded along the space-time edge, whereas the rest of the kernel values are near zero.

In what follows, at a position $\mathbf{x}_i$, we will essentially be using (a normalized version of) the function $K(\mathbf{x}_l - \mathbf{x}_i)$ as descriptors, representing a video's inherent local space-time geometry. To be more specific, the 3D LSK function $K(\mathbf{x}_l - \mathbf{x}_i)$ is densely calculated and normalized as follows:

$$W_I^i = \frac{K(\mathbf{x}_l - \mathbf{x}_i)}{\sum_{l=1}^{P} K(\mathbf{x}_l - \mathbf{x}_i)}, \tag{10}$$

where I can be Q or T for query or target, respectively.[12] Normalization of this kernel function yields invariance to

[10] λ', λ'', and α are set to the same values as 2D LSKs and fixed for all experiments.

[11] When the magnitude of the motions is large (relative to the support of the local steering kernels specifically), a basic form of coarse but explicit motion compensation will become necessary. We refer the reader to [45] for more detail.

[12] Note that videos here are gray scale. The case of color is worth treating independently and is discussed in [41].

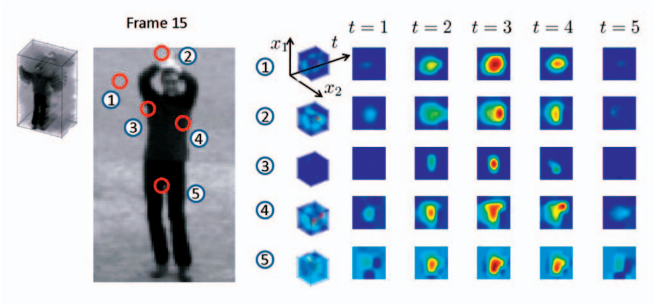


Fig. 6. Examples of 3D LSKs capturing 3D local underlying geometric structure in various regions. In order to compute 3D LSKs, five frames (frame 13 to frame 17) were used. 3D LSKs are shown upsampled for illustration only.

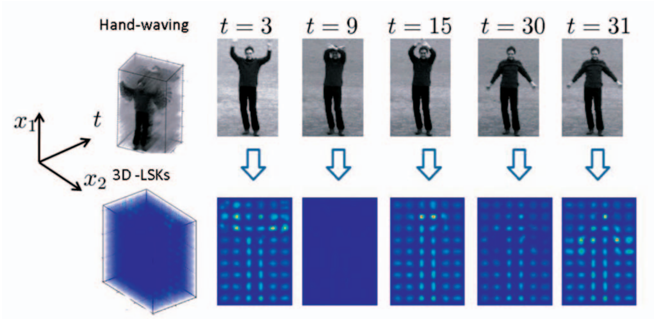


Fig. 7. 3D LSKs computed from a hand-waving action are shown. For graphical description, we only computed 3D LSKs at nonoverlapping $5 \times 5 \times 5$ cubes, even though we compute 3D LSKs densely in practice.

brightness change and robustness to contrast change (as was similarly shown for 2D LSKs in [41]).

Fig. 7 shows that 3D LSKs are effective at capturing local space-time geometry individually, and global space-time geometry collectively. It is interesting to note that 3D LSKs[13] seem related to "HOG3D" introduced in [61]. However, our method is quite different in that our descriptors capture voxel relationships based on the locally measured distance between voxels using a natural signal induced metric, whereas HOG3D mostly makes use of the histogram of quantized local space-time gradients. Furthermore, we extract salient characteristics of 3D LSKs by further applying PCA as described in the following section. We believe that quantization of oriented gradients, while useful in reducing computational complexity, can lead to a significant degradation in the discriminative power of descriptors. This effect is particularly severe in the case where there is only a single positive example available without any prior information, which we will explain in Section 2.3. Superior performance of 3D LSKs over HOG3D is demonstrated in Section 3.1.

2.2 Feature Representation

It has been shown in [41] that the normalized LSKs in 2D follow a power law (i.e., a long tail) distribution. That is to say, the features are scattered out in a high-dimensional feature space, and thus there basically exists no dense cluster in the descriptor space. The same principle applies to 3D LSK. In order to illustrate and verify that the normalized 3D LSKs also satisfy this property, we computed an empirical bin density (100 bins) of the normalized 3D LSKs (using a total of 50,000 3D LSKs) computed from 90 videos of the Weizmann action data set [9] using the K-means clustering method. The utility of this observation becomes clear in the next paragraphs.

In the previous section, we computed a dense set of 3D LSKs from Q and T. These densely computed descriptors are highly informative, but, taken together, tend to be overcomplete (redundant). Therefore, we derive features by applying dimensionality reduction (namely, PCA) to these resulting arrays in order to retain only the salient characteristics of the 3D LSKs. As also observed in [67], [30], an ensemble of local features with even a little discriminative power can together offer significant discriminative power. However, both quantization and informative feature selection on a long-tail distribution can lead to a precipitous drop in performance. Therefore, instead of any quantization and informative feature selection, we focus on reducing the dimension of 3D LSKs using PCA.[14]

This idea results in a new feature representation with a moderate dimension which inherits the desirable discriminative attributes of 3D LSK. The distribution of the resulting features sitting on the low-dimensional manifold also tends to follow a power law distribution and this allows us to use the MCS measure, which will be illustrated in Section 2.3. The optimality property and justification of MCS can be found in [41].

In order to organize W_Q and W_T, which are densely computed from Q and T, let $\mathbf{W}_Q$ and $\mathbf{W}_T$ be matrices, whose columns are vectors $\mathbf{w}_Q$ and $\mathbf{w}_T$, which are column-stacked (rasterized) versions of W_Q and W_T, respectively:

$$\begin{aligned}\mathbf{W}_Q &= \left[\mathbf{w}_Q^1, \ldots, \mathbf{w}_Q^n\right] \in \mathbb{R}^{P \times n},\\ \mathbf{W}_T &= \left[\mathbf{w}_T^1, \ldots, \mathbf{w}_T^{n_T}\right] \in \mathbb{R}^{P \times n_T},\end{aligned} \tag{11}$$

where n and n_T are the number of cubes where 3D LSKs are computed in the query Q and the target T, respectively.

As described in Fig. 2, the next step is to apply PCA to $\mathbf{W}_Q$ and retain the first (largest) d principal components,[15] which form the columns of a matrix $\mathbf{A}_Q \in \mathbb{R}^{P \times d}$. Next, the lower dimensional features are computed by projecting $\mathbf{W}_Q$ and $\mathbf{W}_T$ onto $\mathbf{A}_Q$:

$$\begin{aligned}\mathbf{F}_Q &= \left[\mathbf{f}_Q^1, \ldots, \mathbf{f}_Q^n\right] = \mathbf{A}_Q^T \mathbf{W}_Q \in \mathbb{R}^{d \times n},\\ \mathbf{F}_T &= \left[\mathbf{f}_T^1, \ldots, \mathbf{f}_T^{n_T}\right] = \mathbf{A}_Q^T \mathbf{W}_T \in \mathbb{R}^{d \times n_T}.\end{aligned} \tag{12}$$

Fig. 8 illustrates that the principal components $\mathbf{A}_Q$ learned from different actions, such as surfing and diving actions, are quite distinct from each other. Fig. 9 shows what features $\mathbf{F}_Q$ and $\mathbf{F}_T$ look like for a walking action. In order to show where actions appear, we drew red ovals around each action in the target video. These examples illustrate (as quantified later in

[13] HoG [65] and HoF [66] are also related to our 2D LSKs ($x_1 - x_2$-axes) and 2D LSKs (either $x_1 - t$-axes or $x_2 - t$-axes).

[14] Ali and Shah [12] also pointed out that interest point descriptor-based action recognition methods have a limitation in that useful pieces of global information may be lost.

[15] Typically, d is selected to be a small integer, such as three or four, so that 80 to 90 percent of the information in the LSKs would be retained (i.e., $\frac{\sum_{i=1}^{d} \lambda_i}{\sum_{i=1}^{P} \lambda_i} \geq 0.8$ (to 0.9), where λ_i are the eigenvalues).

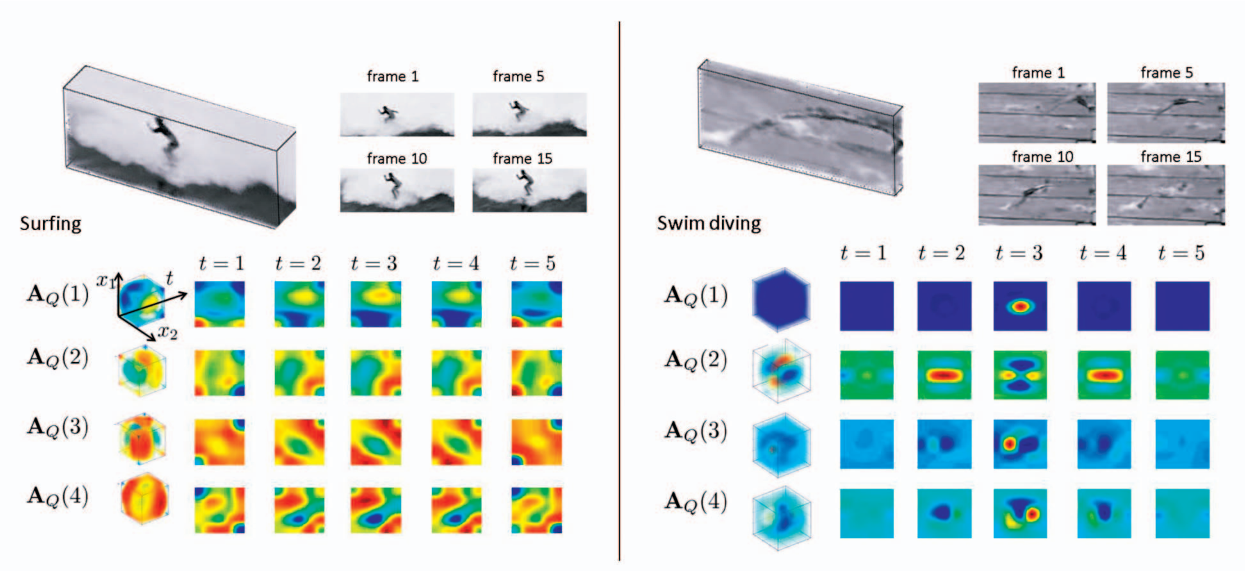


Fig. 8. Examples of the top four principal components in $\mathbf{A}_Q$ for actions such as surfing and diving. Note that these eigenvectors reveal the geometric characteristic of queries in both space and time domain, and thus they are totally different from linear 3D Gabor filters. Eigenvectors $\mathbf{A}_Q$ were upscaled for illustration purposes.

the paper) that the derived feature volumes have good discriminative power even though we do not involve any learning over a set of training examples.

It is worth noting that features derived from 3D LSKs are not similar to 3D Gabor filter responses. In fact, 3D LSKs are highly nonlinear, but stable in the presence of uncertainty in the data, while Gabor filters are linear and provide a fixed basis no matter what the given query. The Gabor representation may work reasonably well with supervised learning methods, but this does not necessarily mean that it is appropriate for the single-query framework of interest to us, which we describe in the next section. We justify these points by showing both quantitative and qualitative comparison between 3D LSK and 3D Gabor filter responses in Section 3.1.1.

A similar approach was also taken by [68], where PCA was applied to interest point descriptors such as SIFT, leading to enhanced performance. Very recently, [12] proposed a set of kinematic features that extract different aspects of motion dynamics present in the optical flow. They obtained bags of kinematic modes for action recognition by applying PCA to a set of kinematic features. We differentiate our proposed method from [12] in the sense that 1) motion information is implicitly contained in 3D LSK while [12] explicitly compute optical flow, 2) background subtraction was used as a preprocessing step while our method is fully automatic, and 3) [12] employed multiple instance learning for action classification, while our proposed method deals with both action detection and classification from a single example.

2.3 Detecting Similar Actions Using the Matrix Cosine Measure

2.3.1 Matrix Cosine Similarity

The next step in the proposed framework is a decision rule based on the measurement of a *distance* between the computed feature volumes $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$. We were motivated by earlier works, such as [50], [46], [47], that have shown the effectiveness of correlation-based similarity.

The MCS between two feature matrices $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$ which consist of a set of feature vectors can be defined as the Frobenius inner product between two normalized matrices as follows:

$$\rho(\mathbf{F}_Q,\mathbf{F}_{T_i}) = <\overline{\mathbf{F}}_Q,\overline{\mathbf{F}}_{T_i}>_F = \text{trace}\left(\frac{\mathbf{F}_Q^T\mathbf{F}_{T_i}}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}\right) \in [-1,1], \tag{13}$$

where $\overline{\mathbf{F}}_Q = \frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F} = \frac{1}{\|\mathbf{F}_Q\|_F}[\mathbf{f}_Q^1,\dots,\mathbf{f}_Q^n]$ and $\overline{\mathbf{F}}_{T_i} = \frac{\mathbf{F}_{T_i}}{\|\mathbf{F}_{T_i}\|_F} = \frac{1}{\|\mathbf{F}_{T_i}\|_F}[\mathbf{f}_{T_i}^1,\dots,\mathbf{f}_{T_i}^n]$. Equation (13) can be rewritten as a weighted sum of the vector cosine similarities $\rho(\mathbf{f}_Q,\mathbf{f}_{T_i}) = \frac{\mathbf{f}_Q^T\mathbf{f}_{T_i}}{\|\mathbf{f}_Q\|\|\mathbf{f}_{T_i}\|}$ [50], [46], [47] between each pair of corresponding feature vectors (i.e., columns) in $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$ as follows:

$$\rho(\mathbf{F}_Q,\mathbf{F}_{T_i}) = \sum_{\ell=1}^{n}\frac{\mathbf{f}_Q^{\ell\,T}\mathbf{f}_{T_i}^{\ell}}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F} = \sum_{\ell=1}^{n}\rho(\mathbf{f}_Q^{\ell},\mathbf{f}_{T_i}^{\ell})\frac{\|\mathbf{f}_Q^{\ell}\|\|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}. \tag{14}$$

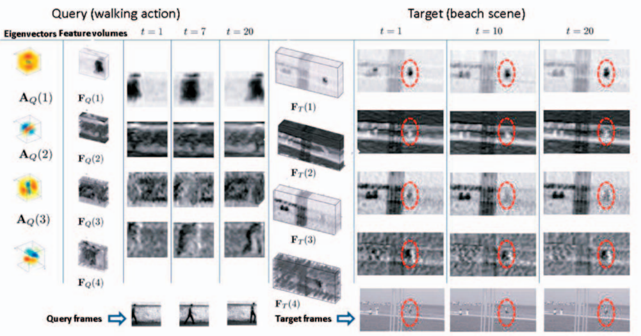


Fig. 9. $\mathbf{A}_Q$ is learned from a collection of 3D LSKs $\mathbf{W}_Q$, and feature row vectors of $\mathbf{F}_Q$ and $\mathbf{F}_T$ are computed from query Q and target video T, respectively. Eigenvectors and feature vectors were transformed to volume and upscaled for illustration purposes.

The weights are represented as the product of $\frac{\|\mathbf{f}_Q^{\ell}\|}{\|\mathbf{F}_Q\|_F}$ and $\frac{\|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_{T_i}\|_F}$, which indicate the relative importance of each feature in the feature sets $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$. We see here an advantage of the MCS in that it takes account of the strength and angle similarity of vectors at the same time. Hence, this measure not only generalizes the cosine similarity naturally, but also overcomes the disadvantages of the conventional euclidean distance, which is sensitive to outliers.[16]

It is worth noting that [22] proposed 3D volume correlation score (global consistency measure between query and target cube) by computing a weighted average of local consistency measures. The difficulty with that method is that local consistency values should be explicitly computed from each corresponding subvolume of the query and target video. Furthermore, the weights to calculate a global consistency measure are based on a sigmoid function, which is somewhat ad hoc. Here, we claim that our MCS measure is better motivated, more general, and more effective than their global consistency measure for action detection, as we also allude to in Section 3.1.1.

The next step is to generate a so-called resemblance volume (RV), which will be a volume of voxels, each indicating the likelihood of similarity between the Q and T_i. As for the final test statistic comprising the values in the resemblance volume (as also described in [41]), we use the *proportion* of shared variance (ρ_i^2) to that of the "residual" variance $(1-\rho_i^2)$. More specifically, RV is computed as follows[17]:

$$RV : f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}. \tag{15}$$

The resemblance volume generated from $f(\rho_i)$ provides better contrast and dynamic range in the result $(f(\rho_i) \in [0, \infty])$. More importantly from a quantitative point of view, we note that $f(\rho_i)$ is essentially the Lawley-Hotelling trace statistic [70], [71], which is used as an efficient test statistic for detecting correlation between two data sets. Furthermore, historically, this statistic has been suggested in the pattern recognition literature as an effective means of measuring the separability of two data clusters (e.g., [67].)

2.3.2 Significance Testing by Controlling the False Discovery Rate (FDR) [72]

If the task is to find the most similar cube (T_i) to the query (Q) in the target video, one can choose the cube which results in the largest value in the RV (i.e., $\max f(\rho_i)$) among all of the cubes, no matter how large or small the value is in the range of $[0, \infty]$. This, however, is unwise because there may be no instances of the action of interest or perhaps multiple actions of interest. Therefore, more generally, we are interested in multiple simultaneous hypotheses. We associate each voxel $(f(\rho_i))$ of the RV with M possible hypotheses $(\mathcal{H}_0, \dots, \mathcal{H}_{M-1})$ as follows:

$\mathcal{H}_0$:	T_0 is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_0) < \tau$,
$\mathcal{H}_1$:	T_1 is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_1) < \tau$,
$\vdots$	$\vdots$		$\vdots$
$\mathcal{H}_{M-1}$:	T_{M-1} is not similar to the given query Q	$\Leftrightarrow$	$f(\rho_{M-1}) < \tau$.

where τ is a threshold for detection. Suppose that there are m_0 true null hypotheses among the M test hypotheses. Let R denote the number of hypotheses rejected. This observable random variable R can be decomposed as $V + S$, where V is the number of *incorrectly* rejected null hypotheses and S is the number of *correctly* rejected null hypotheses. The proportion of errors committed by falsely rejecting null hypotheses can be viewed through $\frac{V}{R}$. Let U be the unobservable random quotient,

$$U = \begin{cases} \frac{V}{R}, & \text{if} \quad R > 0, \\ 0, & \text{otherwise.} \end{cases} \tag{16}$$

The FDR is defined as $\mathbf{E}(U)$, the expected error rate. The Benjamini-Hochberg procedure proposed in [72] controls the FDR at a desired level α while maximizing $\mathbf{E}(R)$. Let $\{p_0, p_1, \dots, p_{M-1}\}$ denote the p-values corresponding to the test statistics $\{f(\rho_0), f(\rho_1), \dots, f(\rho_{M-1})\}$ and $p_{(0)} \le p_{(1)} \le \dots \le p_{(M-1)}$ denote the ordered p-values corresponding to the hypotheses $\{\mathcal{H}_{(0)}, \mathcal{H}_{(1)}, \dots, \mathcal{H}_{(M-1)}\}$. By definition, $p_i = 1 - P_{\mathcal{H}_i}$, where $P_{\mathcal{H}_i}$ is the cumulative distribution function of resemblance volume under the null hypothesis $\mathcal{H}_i$. The FDR-controlling procedure is easily implemented. For the M voxels being tested, the general procedure is as follows:

1. Select a desired FDR bound α between 0 and 1. This is the maximum FDR that we are willing to tolerate on average.
2. Order the p values from the smallest to largest: $p_{(0)} \le p_{(1)} \le \dots \le p_{(M-1)}$.
   Let $f(\rho_{(i)})$ be the voxel corresponding to $p_{(i)}$.
3. Let γ be the largest i for which $p_{(i)} \le \frac{i}{M}\alpha$.
4. Identify the threshold τ corresponding to $p_{(\gamma)}$ and declare that the voxels of RV which are above τ contain similar actions to the given query Q.

After the significance testing with τ is performed, we employ the idea of nonmaxima suppression [73] for the final detection. Namely, we take the volume region with the highest $f(\rho_i)$ score and eliminate the possibility that any other action is detected within some radius[18] of the center of that volume again. This enables us to avoid multiple false detections of nearby actions already detected. Then, we

[16] We compute $\rho(\mathbf{F}_Q, \mathbf{F}_{T_i})$ over M (a possibly large number of) target cubes and this can be efficiently implemented by column-stacking the matrices $\mathbf{F}_Q$ and $\mathbf{F}_{T_i}$ and simply computing the (vector) cosine similarity between two long-column vectors as follows:

$$\begin{aligned} \rho_i \equiv \rho(\mathbf{F}_Q, \mathbf{F}_{T_i}) &= \sum_{\ell=1}^{n} \frac{{\mathbf{f}_Q^{\ell}}^T \mathbf{f}_{T_i}^{\ell}}{\|\mathbf{F}_Q\|_F \|\mathbf{F}_{T_i}\|_F} \\ &= \rho(\mathrm{colstack}(\mathbf{F}_Q), \mathrm{colstack}(\mathbf{F}_{T_i})) \in [-1, 1], \end{aligned}$$

where $\mathrm{colstack}(\cdot)$ means an operator, which column-stacks (rasterizes) a matrix.

[17] While the transformation is a monotonic function of the ρ statistic, its effect is not superfluous. Clearly, the distribution of $f(\rho)$ is different from that of ρ. Indeed, it is known that this transformation yields a new random variable which asymptotically approaches a fixed density, namely, a squared Student-*t* variable, regardless of the density of the input data [69]. Practically speaking, the usefulness of this transformation is, in fact, that it normalizes the chosen threshold.

[18] The size of this exclusion region will depend on the application at hand and the characteristics of the query video.

iterate this process until the local maximum value falls below the threshold τ.

3 Experimental Results

In this section, we demonstrate the performance of the proposed method with comprehensive experiments on four data sets, namely, the general action data set [22], the drinking data set, [62], the Weizmann action data set [9], and the KTH action data set [1]. The general action and the drinking data sets are used to evaluate the detection performance of the proposed method, while the Weizmann action and the KTH action data sets are employed for action categorization. Comparison is made with state-of-the-art methods that have reported their results on these data sets.

3.1 Action Detection

In this section, we show several experimental results on searching with a short query video against a (typically longer and larger) target video. Our method detects the presence and location of actions similar to the given query and provides a series of bounding cubes with resemblance volume embedded around detected actions. Note again that no background/foreground segmentation and no explicit motion estimation are required in the proposed method. Our proposed method can also handle modest variations in rotation (up to ± 15 degrees) and spatial and temporal scale change (up to ± 20 percent). For larger variations in scale, we use a multiscale approach, as done similarly in [41], and show in the following section that this results in improvement over the single-scale implementation.

Given Q and T, we spatially blur and downsample both Q and T by a factor of three in order to reduce the time complexity. We then compute 3D LSK of size 3×3 (space) $\times 7$ (time) as descriptors so that every space-time location in Q and T yields a 63D local descriptor $\mathbf{W}_Q$ and $\mathbf{W}_T$, respectively. The reason why we choose a larger time-axis size than space axis of the cube is that we focus on detecting similar actions, regardless of different appearances. Thus, we give a higher priority to temporal evolution information than spatial appearance. We end up with $\mathbf{F}_Q$ and $\mathbf{F}_T$ by further reducing the dimension of descriptors[19] to d using PCA. Finally, we obtain RV by computing the MCS measure between $\mathbf{F}_Q$ and $\mathbf{F}_T$. After significance testing by controlling the FDR with a specified α value[20] and nonmaxima suppression explained in Section 2.3.2, the proposed method localizes actions of interest.[21]

3.1.1 The General Action Data Set [54]

This data set contains three pairs of action query and target videos. Note that, in all cases, the query video is not from the target video sequence (see Fig. 10).

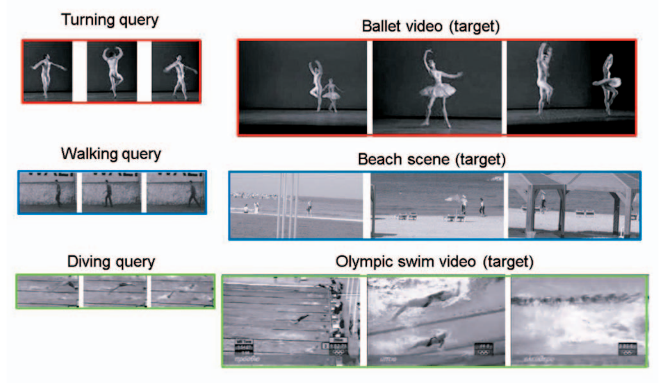


Fig. 10. Examples of a general action data set [22]: a turning query and ballet video, a walking query and beach scene video, and a diving query and Olympic swim relay video.

1. The query video contains a single turn of a male dancer (13 frames of 90×110 pixels), while the target video (766 frames of 144×192 pixels) includes ballet actions from a male and a female dancer.
2. The query video contains a very short walking action moving to the left (14 frames of 60×70 pixels) with a stationary stone wall in the background, while the target video has walking people in a beach scene (456 frames of 180×360 pixels) with crashing waves in the background.
3. The query video contains a swimmer's dive into a pool (16 frames of 70×140 pixels), while the target is an Olympic relay-match video (757 frames of 240×360 pixels), which was severely MPEG compressed.

As we alluded to in Section 1.3, we compare our 3D LSK with 3D Gabor filter response [74] and HOG3D [61] both qualitatively and quantitatively.[22] Fig. 11 shows a comparison of resemblance volumes with 3D LSK, HOG3D, and 3D Gabor filter for three data sets. Note that we plugged in HOG3D and 3D Gabor instead of 3D LSK, while the rest of the process in the proposed action detection framework remains exactly same. A red value in RVs signifies higher resemblance to the given query actions, while blue means lower resemblance. 3D LSKs provide the most consistent results with the ground truth. We observe that RVs with 3D LSKs reveal most relevant actions with a few false positives, whereas HOG3D results in many false positives and 3D Gabor filter misses most actions. It is worth noting that actions in target videos vary in scale. This can be better dealt with multiscale approach as described below.

Multiscale action detection. We construct a multiscale pyramid of the target feature volume $\mathbf{F}_T$. We resize the

[19] Note that $d = 4$ for the walking query, whereas $d = 7$ for the ballet turning and diving queries.

[20] In our experiments, $\alpha = 0.01$ works well.

[21] The localization is considered to be correct when the detected region is 50 percent overlapped with the ground truth.

[22] We set parameters for HOG3D and 3D Gabor filters as follows:

1) HOG3D [61]: A 3D patch of interest is divided into $3 \times 3 \times 2$ space-time cells. The corresponding descriptor concatenates oriented gradient (10 orientations) histograms of all cells and is then normalized. With dense sampling (x_1x_2-stride: six pixels apart and t-stride: one pixel apart), the resulting descriptors have 180 dimensions at every sampled position. We use the executable binary from the authors' website. (Downloadable from http://lear.inrialpes.fr/people/klaser/software_3d_video_descriptor. We set the parameters for this method to achieve its best performance. These parameters were not the same as those setting recommended at the website. This is because the recommended settings were not best suited for the general action data set.)

2) 3D Gabor [74]: We used 16 of 3D Gabor filter responses (0, $\pi/4$, $\pi/2$, $3\pi/4$: preferred direction of motion) and (1, 2, 3, 4: preferred speed of the filter (in pixels per frame)). We use a Matlab code from the website. (Downloadable from http://www.cs.rug.nl/imaging/spatiotemporal_Gabor_function/GaborApp.html.)

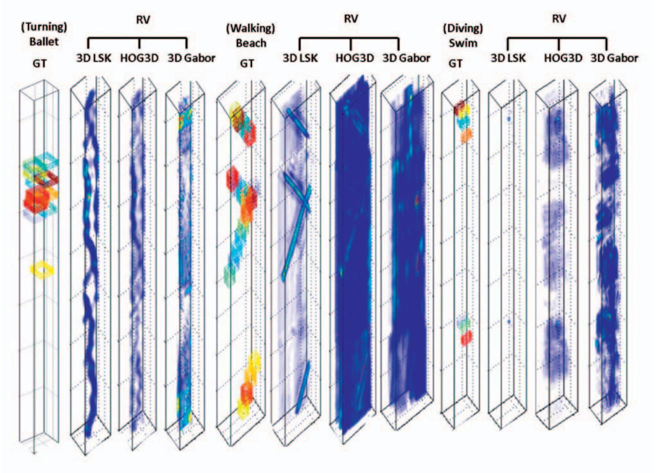


Fig. 11. Comparison of RV among 3D LSK, HOG3D, and 3D Gabor for three pairs of videos (ballet with a turning query, beach with a walking query, and swim with a diving query). HOG3D was computed densely for a fair comparison. Note that colors in the ground truth volume are used to distinguish individual actions from each other. This figure is better viewed in color.

target feature volume size by steps of 10 percent so that a relatively fine quantization of spatial scales are taken into account. By using five scale factors from 0.9 to 1.3, we obtain five resemblance volumes. These resemblance volumes represent the likelihood functions $p(f(\rho_i)|S_i)$, where S_i is the scale at $\mathbf{x}_i$. However, the sizes of the respective resemblance volumes are naturally different. Therefore, we simply rescale all the resemblance volumes by voxel replication so that they match the dimensions of the original target volume. Next, the maximum likelihood estimate of the scale at each position is arrived at by comparing the rescaled resemblance volumes as follows:[23]

$$\hat{S}_i = \arg\max_{S_i} p(\underline{\mathrm{RV}}|S_i). \tag{17}$$

It is worth noting that action detection methods [22], [54], [25] which were also tested on this data set only presented qualitative results with either empirically chosen threshold values or no description about how the threshold values are determined. On the other hand, the threshold values were automatically chosen in our algorithm by controlling the FDR with respect to the specified α. Unlike [22], [54], [25], we provide the precision-recall curves in Fig. 12 for quantitative evaluation. For these experiments, we used the entire frames, while [22], [54], [25] used a part of the video frames. The detection result of the proposed method on this video outperforms those in [22] and [25], and compares favorably to that in [54] in terms of visual detection accuracy. As shown in Figs. 12 and 13 and expected from qualitative comparison in Fig. 11, 3D LSK clearly outperforms HOG3D and 3D Gabor.

Effect of parameters. We examined how the performance of the proposed method is affected by the choice of parameters P (the size of 3D LSK) and h (the smoothing parameter). Fig. 14 illustrates equal error rates for 3D LSKs in single-scale implementation. As shown in Fig. 14, the overall performance of the proposed method changes gracefully with the particular choice of parameter h and P. It appears that the best performance can be achieved with the fixed choice of $P = 3 \times 3 \times 7$ and $h = 2.3$ across three video data seta.

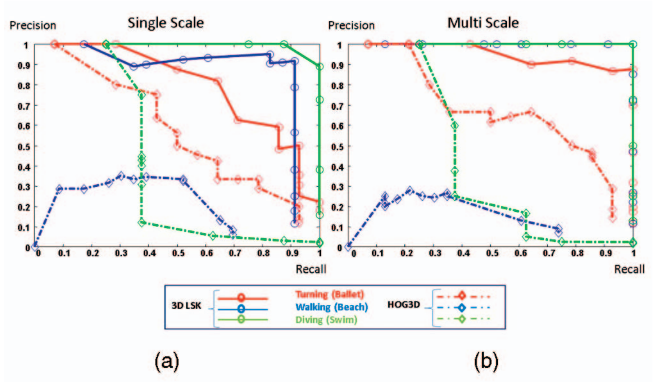
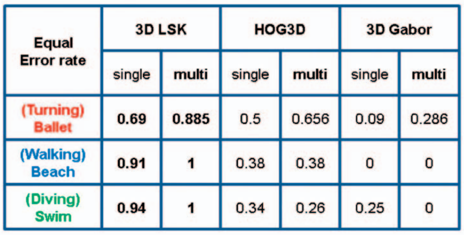


Fig. 12. (a) Comparison of precision-recall curves between 3D LSK and HOG3D for three different actions (walking, ballet turning, and diving) in single-scale implementation. (b) Multiscale comparison. Note that other state-of-the-art action detection methods in [22], [54], [25] did not provide any quantitative performance on these examples. This figure is better viewed in color.

3.1.2 The Drinking Action Data Set [62]

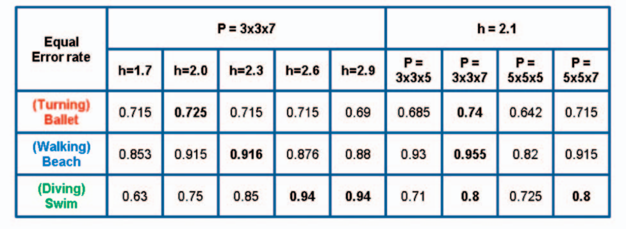
In this section, we further evaluate our method on more challenging scenarios, such as real movie scenes. The drinking action data set is comprised of a total of 36,000 frames from two episodes of the movie *Coffee and Cigarette*. The data set includes 37 drinking actions from the episodes "Cousins?" and "Delirium." Fig. 15b illustrates how drinking actions in target video samples largely vary in scales and viewpoints as well as the background clutter. Furthermore, there are abrupt scene changes, and the size and appearance of cups also vary. We chose one drinking action (55 frames of 107 × 101 pixels) as a query (see Fig. 15a) from the episode called "No Problem." Thus, there is no overlap between the query and the target videos. We take the multiscale approach in temporal axis as well as in spatial axis because temporal extents of drinking actions in the test set vary from 30 to 200 frames with the mean length of 70 frames. More specifically, we used nine spatial scales from 0.7 to 1.5 and six temporal scales from 0.8 to 1.3. As explained in Section 3.1.1, we take a maximum value across all scales at each voxel and end up with one RV. In order to deal with variations in view points, we used mirror-reflected version of the query as well. By voting the higher score among



Equal Error rate	3D LSK		HOG3D		3D Gabor	
	single	multi	single	multi	single	multi
(Turning) Ballet	0.69	0.885	0.5	0.656	0.09	0.286
(Walking) Beach	0.91	1	0.38	0.38	0	0
(Diving) Swim	0.94	1	0.34	0.26	0.25	0

Fig. 13. Comparison of equal error rates between 3D LSK, HOG3D, and 3D Gabor filter for three different actions (walking, ballet turning, and diving).

[23] By $\underline{\mathrm{RV}}$, we mean a collection of RV indexed by i at each position.



Equal Error rate	P = 3x3x7					h = 2.1			
	h=1.7	h=2.0	h=2.3	h=2.6	h=2.9	P = 3x3x5	P = 3x3x7	P = 5x5x5	P = 5x5x7
(Turning) Ballet	0.715	**0.725**	0.715	0.715	0.69	0.685	**0.74**	0.642	0.715
(Walking) Beach	0.853	0.915	**0.916**	0.876	0.88	0.93	**0.955**	0.82	0.915
(Diving) Swim	0.63	0.75	0.85	**0.94**	**0.94**	0.71	**0.8**	0.725	**0.8**

Fig. 14. Equal error rates with respect to different parameter settings on three data sets, where equal error rate means recall rate when recall rate is the same as precision rate.

Fig. 15. The drinking data set [62]. (Left) A query video chosen from the episode "No Problem." (Right) Some target video samples from the episodes "Cousin?" and "Delirium."

values from two RVs at every space-time location, we arrive at one RV, which includes correct locations of drinking action. The performance of our method on this test set in comparison to Laptev's methods [62] is illustrated in Fig. 16 in terms of precision-recall curves and average precision (AP) values. Note that Laptevs 1, 2, and 3 are based on discrete AdaBoost using 106 positive examples for training. As discussed in [62], Laptev 1 uses HOF with additional keyframe priming, while Laptevs 2 and 3 use HOG3D. Even though we use a single frontal view query, the proposed method performs favorably with Laptevs 1 and 2. The 20 strongest detections (sorted in decreasing order of resemblance volume score) with the proposed method are illustrated in Fig. 17. In spite of a substantial variation in subject appearance, motion, surrounding scenes, viewpoints, and scales, and also abrupt scene change in the video, the proposed method retrieved most of the actions at the correct locations. We expect that our method might also benefit from keyframe priming, as discussed in [62].

3.2 Action Category Classification

As opposed to action detection, action category classification aims to classify a given action query into one of several prespecified categories. In earlier discussion on action detection, we assumed that, in general, the query video is smaller than the target video. Now we relax this assumption and thus we need a preprocessing step which selects a valid human action from the query video. This idea allows us not only to extend the proposed detection framework to action category classification but also improves both detection and classification accuracy by removing unnecessary background from the query video.

Once the query video is cropped to a short-action clip, the cropped query is searched against each labeled video in the database, and the value of the RV is viewed as the likelihood of similarity between the query and each labeled video. Then, we classify a given query video as one of the predefined action categories using an NN classifier.

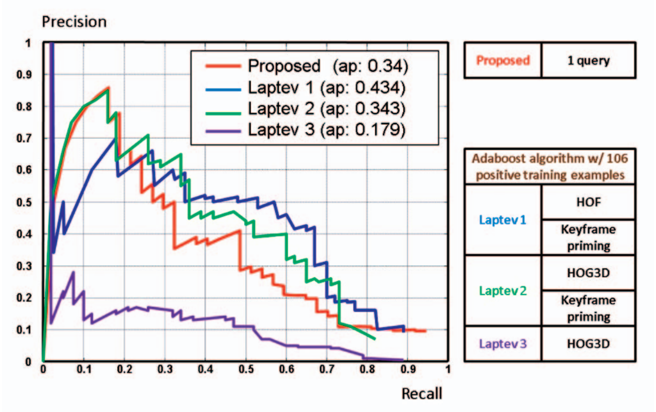


Fig. 16. Precision-recall curves comparison between the proposed method and three action detection methods by [62]. The proposed method performs favorably with Laptev 1 and 2 even though there is a single query video used. The AP means an average precision over the entire range of recall. This figure is better viewed in color.

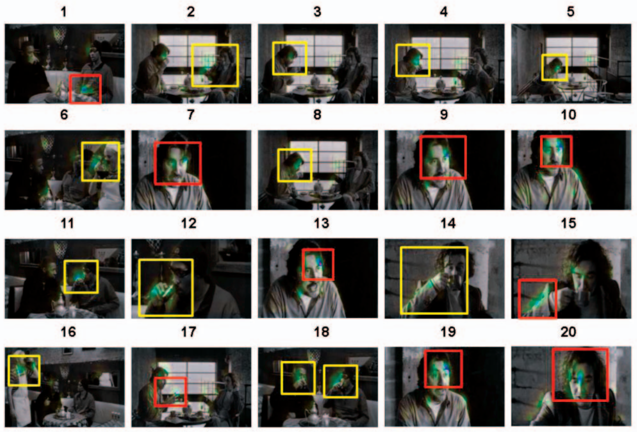


Fig. 17. Detection of drinking actions (yellow: true positives and red: false positives) sorted in decreasing confidence order by the proposed method. This figure is better viewed in color.

3.2.1 Action Cropping in Videos

In this section, we introduce a procedure which automatically extracts from the query video a small cube that only contains a valid action. Space-time saliency detection [42] can provide such a mechanism. We downsample each frame of query video Q to a coarse spatial scale (64×64) in order to reduce the time complexity.[24] We then compute 3D LSK of size $3 \times 3 \times 3$ as features and generate feature matrices $\mathbf{F}_i$ in a $(3 \times 3 \times 7)$ local space-time neighborhood. We generated space-time saliency maps S by computing self-resemblance measure, as shown in Fig. 18. Then we again use the idea of nonparametric significance testing to detect space-time proto-objects, namely, we compute an empirical PDF from all the saliency values and set a threshold by controlling FDR[25] with $\alpha = 0.05$ in deciding whether the given saliency values are in the extreme (right) tails of the empirical PDF. The approach is based on the assumption that in the video, a salient action is a relatively rare event, and thus results in values which are in the tails

[24] We do not downsample the video in the time domain.

[25] We select a somewhat loose α level here, since we do not wish to miss the relevant action in the query.

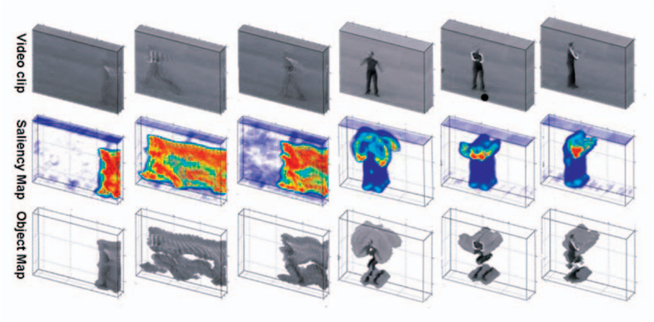


Fig. 18. Found space-time proto-objects from the KTH data set [1].

of the distribution of saliency map values. After making a binary map by thresholding the space-time saliency map, a morphological filter is applied. More specifically, we dilate the binary object map with a disk shape of size 5×5. Proto-objects are extracted from corresponding locations of the original video. Fig. 18 shows that the space-time saliency detection method[26] successfully detects only salient human actions in the KTH data set [1]. Next, we crop the valid human action region by fitting a 3D rectangular box to space-time proto-objects.

3.2.2 The Weizmann Action Data Set [9]

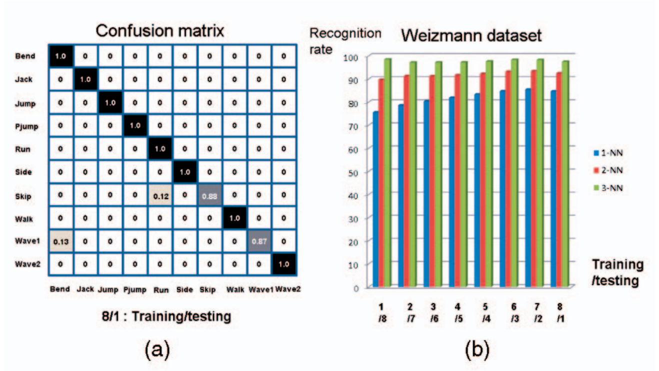
The Weizmann action data set contains 10 actions (bend, jumping jack, jump forward, jump in place, jump sideways, skip, run, walk, wave with two hands, and wave with one hand) performed by nine different subjects. This data set contains videos with static cameras and simple background, but it provides a good testing environment to evaluate the performance of the algorithm when the number of categories is large compared to the KTH data set (a total of six categories). We conducted experiments on the Weizmann data set under various data split setups. For example, the videos of m subjects are randomly drawn for testing (query) and the videos of the remaining $9 - m$ subjects are labeled for each run where $m \in [1, \ldots, 8]$. We applied the automatic action cropping method introduced in the previous section to the query video. Then, the resulting short action clip is matched against the remaining labeled videos using the proposed method. We classify each testing video as one of the 10 action types by 3-NN (nearest neighbor) as similarly done in [25]. The results are reported as the average of 100 runs. To begin, we achieved a recognition rate of 97.5 percent for all 10 actions in the leave-one-out setting ($m = 1$). The recognition rate comparison is provided in Table 1 as well. The proposed method performs favorably against state-of-the-art methods [14], [57], [20], [75], [12], [76], [77], [61]. We observe that these results also compare favorably to several state-of-the-art methods even though our method involves no training phase and requires no background/foreground segmentation. As an added bonus, our method provides localization of actions as a side benefit. Fig. 19a shows the confusion matrix for our method.

Next, we provide further results using 1-NN and 2-NN in comparison to 3-NN in Fig. 19b with respect to various split setups. It is worth noting that the recognition rates are quite stable, regardless of the split used.

TABLE 1
Comparison of Average Recognition Rate on the Weizmann Data Set [9]

Our approach	3-NN	2-NN	1-NN
Recognition rate	**97.5%**	**92.5%**	**84.7%**

Method	Junejo et al. [57]	Liu et al. [20]	Klaser et al. [61]	Schindler and van Gool [78]
Recog. rate	95.33%	90%	84.3%	100%
Method	Niebles et al. [14]	Ali et al. [12]	Sun et al. [79]	Fathi and Mori [80]
Recog. rate	90%	95.75%	97.8%	100%
Method	Jhuang et al. [75]	Batra et al. [76]	Bregonzio et al. [81]	Zhang et al. [77]
Recog. rate	98.8%	92%	96.6%	92.89%


Confusion matrix

	Bend	Jack	Jump	Pjump	Run	Side	Skip	Walk	Wave1	Wave2
Bend	1.0	0	0	0	0	0	0	0	0	0
Jack	0	1.0	0	0	0	0	0	0	0	0
Jump	0	0	1.0	0	0	0	0	0	0	0
Pjump	0	0	0	1.0	0	0	0	0	0	0
Run	0	0	0	0	1.0	0	0	0	0	0
Side	0	0	0	0	0	1.0	0	0	0	0
Skip	0	0	0	0	0.12	0	0.88	0	0	0
Walk	0	0	0	0	0	0	0	1.0	0	0
Wave1	0.13	0	0	0	0	0	0	0	0.87	0
Wave2	0	0	0	0	0	0	0	0	0	1.0

8/1 : Training/testing

(a)

Fig. 19. (a) Confusion matrix on the Weizmann data set for the leave-one-out setting. (b) Average recognition rate according to various data split setups (Weizmann data set).

3.2.3 The KTH Action Data Set [1]

In order to further quantify the performance of our algorithm, we also conducted experiments on the KTH data set. The KTH action data set contains six types of human actions (boxing, hand waving, hand clapping, walking, jogging, and running), performed repeatedly by 25 subjects in four different scenarios: outdoors (c_1), outdoors with camera zoom (c_2), outdoors with different clothes (c_3), and indoors (c_4). This data set seems more challenging than the Weizmann data set because there are large variations in human body shape, view angles, scales, and appearance. We also evaluate our method on the KTH data set under various split setups. First, we use the same setup as in [1], i.e., eight people for training[27] and nine for testing for each run. The recognition rates are reported as the average of 100 runs for this setup. We were able to achieve a recognition rate of 95.1 percent on these six actions. Fig. 20a shows the average confusion matrix across all scenarios for this setup. The recognition rate comparison with competing methods is provided in Table 2 as well. It is worth noting that our method outperforms all of the other state-of-the-art methods and is fully automatic. We further tried other data-split setups as similarly done in the previous section. The videos of m subjects are randomly

[26] We refer the reader to Fig. 19 in [42] for more challenging cases where the background is very cluttered and moving as well.

[27] We use the term "training" here to be consistent with notation used in the literature, even though our method does not require training mechanisms.

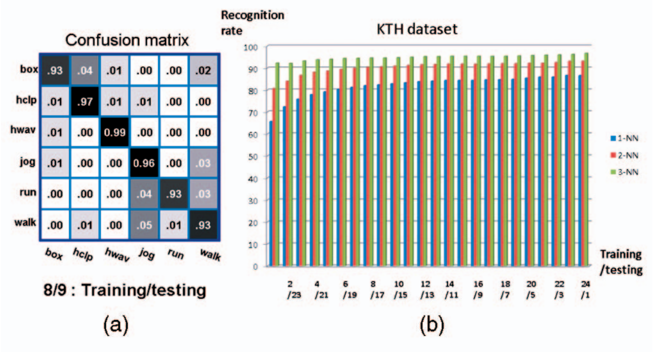


Fig. 20. (a) Confusion matrix on the KTH data set for the eight training/nine testing setup. (b) Average recognition rate according to different data split setup (KTH data set).

drawn for testing (query) and the videos of the remaining subject $25 - m$ are labeled for each run, where $m \in [1, \ldots, 24]$. As shown in Fig. 20b, it is consistent with the results on the Weizmann data set that the recognition rates are quite stable regardless of the split used, as similarly stated in [82].

3.2.4 Discussion

It is important to note that our features computed using the PCA process are a function of the input query video and therefore are adapted to each changing query. As such, one would expect them to serve better in identifying actions that are similar to the given query in a way that is more accurate than would a generic basis. Indeed, the trade-off between having a fixed basis for all input queries and a basis that is extracted from each query manifests itself as a trade-off between stability and specificity. Despite the higher computational cost we pay, our process for extraction of features appear to be stable, yet showing rather high specificity at the same time, resulting in overall very good performance.

Our system is designed with recognition accuracy as a high priority. A typical run of the action detection system implemented in Matlab takes a little over 1 minute on a target video T (50 frames of 144×192 pixels, Intel Pentium CPU 2.66 GHz machine) using a query Q (13 frames of 90×110). Most of the runtime is taken up by the computation of MCS (about 9 seconds, and 16.5 seconds for the computation of 3D LSKs from Q and T, respectively, which needs to be computed only once). There are many factors that affect the precise timing of the calculations, such as query size, complexity of the video, and 3D LSK size. Coarse-to-fine search [86] or branch and bound [31] can be applied to speed up the method. As another way of reducing time complexity, we could use look-up table instead of computing the local covariance matrix $\mathbf{C}$ at every pixel. Even though our method is stable in the presence of a moderate amount of camera motion, our system can benefit from camera stabilization methods as done in [87] and [88] in case of large camera movements.

In the Weizmann and the KTH data sets, target videos contain only one type of action. However, a target video may contain multiple actions in practice. In this case, simple nearest neighbor classifiers can possibly fail. Therefore, we might benefit from contextual information to increase the accuracy of action recognition systems, as similarly done in [89]. In fact, there is broad agreement in the computer vision community about the valuable role that context plays in any image understanding task [90], [91].

TABLE 2
Comparison of Average Recognition Rate on the KTH Data Set

Our approach	3-NN	2-NN	1-NN
Recognition rate	**95.1%**	**91%**	**82.7%**

Method	Kim et al. [24]	Ning et al. [25]	Klaser et al. [61]	Schindler [78] and van Gool [78]
Recog. rate	95.33%	92.31% (3-NN)	91.4%	92.7%
Method	Ali et al. [12]	Niebles et al. [14]	Liu and Shah [82]	Sun et al. [79]
Recog. rate	87.7%	81.5%	94.2%	94%
Method	Dollar et al. [83]	Wong et al. [84]	Rapantzikos et al. [85]	Laptev et al. [66]
Recog. rate	81.17%	84%	88.3%	91.8%

4 Conclusion and Future Directions

In this paper, we have proposed a novel action recognition algorithm by employing *space-time local steering kernels*, which robustly capture underlying space-time data structure, and by using a training-free nonparametric detection scheme based on *Matrix Cosine Similarity*. The proposed method can automatically detect in the target video the presence, the number, as well as the location of actions similar to the given query video by controlling the FDR. Multiscale implementation dealt with large variations in scale of actions and outperformed the single-scale version. In order to increase the detection accuracy and further deal with action classification, we employed an action cropping method based on space-time saliency detection. Challenging sets of real-world human action experiments demonstrated that the proposed approach achieves high-recognition accuracy and improves upon other state-of-the-art methods. Unlike most state-of-the-art methods that involve training, background/foreground segmentation, and manual alignment of actions, the proposed method operates using a *single* example of an action of interest to find similar matches, does not require any prior knowledge (learning) about actions being sought, and does not require any segmentation or preprocessing step of the target video.

The usefulness of 3D LSK descriptors was justified for both action detection and recognition tasks in the example-based, single-query detection scenario in Section 3. It would be interesting to see how the proposed descriptors perform in comparison to state-of-the-art 3D descriptors such as HOG/HOF, HOG3, etc. (see [92] and references therein) in other state-of-the-art action recognition frameworks based on learning mechanisms [12], [14]. In the case where a collection of negative action examples is available, we may be able to boost the action detection performance using the "one-shot similarity (OSS) [93], [94]" kernel, which was recently developed for the face recognition task. Extending the proposed detection framework to joint learning from multiple queries is an excellent direction which we intend to pursue in our future research. Since the proposed method is designed with detection accuracy as a high priority, extension of the method to a large-scale data set requires a significant improvement of the computational complexity of the proposed method. Toward this end, we could benefit from an efficient searching method (coarse-to-fine search) and/or a fast nearest neighbor search method[28] (e.g., vantage point tree [97] and kernelized locality-sensitive hashing [98]).

Since local regression kernels in 2D and 3D were originally designed for image (video) restoration, the proposed framework should become useful in jointly addressing the problems of enhancement and recognition where there might be a degraded query or target. By computing local regression kernels from images (video) at once, we may be able to not only detect objects (actions) of interest, but also enhance images (videos) at the same time. These aspects of the work are the subject of ongoing research.

Appendix

Consider the parameterized surface $S(x_1, x_2) = \{x_1, x_2, z(x_1, x_2)\}$, embedded in the euclidean space $\mathbb{R}^3$. The arclength on the surface is given by $ds^2 = dx_1^2 + dx_2^2 + dz^2$. Applying the chain rule, we have

$$dz(x_1, x_2) = \frac{\partial z}{\partial x_1} dx_1 + \frac{\partial z}{\partial x_2} dx_2 = z_{x_1} dx_1 + z_{x_2} dx_2. \tag{18}$$

Plugging $dz(x_1, x_2)$ into the arclength definition, we have

$$\begin{aligned} ds^2 &= dx_1^2 + dx_2^2 + dz^2 \\ &= dx_1^2 + dx_2^2 + \left(z_{x_1} dx_1 + z_{x_2} dx_2\right)^2 \\ &= \left(1 + z_{x_1}^2\right) dx_1^2 + 2 z_{x_1} z_{x_2} dx_1 dx_2 + \left(1 + z_{x_2}^2\right) dx_2^2, \end{aligned} \tag{19}$$

from which we can extract the metric coefficients

$$\begin{pmatrix} 1 + z_{x_1}^2 & z_{x_1} z_{x_2} \\ z_{x_1} z_{x_2} & 1 + z_{x_2}^2 \end{pmatrix} = \mathbf{C} + \mathbf{I}, \tag{20}$$

where $\mathbf{C}$ is the same covariance matrix in (3) and $\mathbf{I}$ is an identity matrix. In practice, the identity matrix here is absorbed in our calculation of $\mathbf{C}$ in the sense that we find a regularized estimate of $\mathbf{C}$.

Acknowledgments

This work was supported in part by US Air Force Office of Scientific Research Grant FA 9550-07-01-0365.

References

[1] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach," *Proc. IEEE Conf. Pattern Recognition,* June 2004.

[2] T. Darrell and A. Pentland, "Classifying Hand Gestures with a View-Based Distributed Representation," *Proc. Advances in Neural Information Processing Systems,* vol. 6, pp. 945-952, 1993.

[3] J. Yamato, J. Ohya, and K. Ishii, "Recognizing Human Action in Time Sequential Image Using Hidden Markov Model," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1992.

[4] H. Jiang, M. Crew, and Z. Li, "Successive Convex Matching for Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[5] T. Starner and A. Pentland, "Visual Recognition of American Sign Language Using Hidden Markov Model," *Proc. Int'l Workshop Automatic Face and Gesture Recognition,* 1995.

[6] C. Carlsson and J. Sullivan, "Action Recognition by Shape Matching to Key Frame," *Proc. Workshop Models versus Examplars in Computer Vision,* 2001.

[7] A. Yilmaz and M. Shah, "Actions Sketch: A Novel Action Representation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[8] K. Cheung, S. Baker, and T. Kanade, "Shape-from-Silhouette of Articulated Objects and Its Use for Human Body Kinematics Estimation and Motion Capture," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2003.

[9] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-Time Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 12, pp. 2247-2253, Dec. 2007.

[10] A.F. Bobick and J.W. Davis, "The Recognition of Human Movement Using Temporal Templates," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 3, pp. 1257-1265, Mar. 2001.

[11] J. Little and J. Boyd, "Recognizing People by Their Gait: The Shape of Motion," *J. Computer Vision Research,* vol. 1, pp. 2-32, 1998.

[12] S. Ali and M. Shah, "Human Action Recognition in Videos Using Kinematic Features and Multiple Instance Learning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 2, pp. 288-303, Feb. 2010.

[13] Y. Yacoob and M. Black, "Parameterized Modeling and Recognition of Activities," *Computer Vision and Image Understanding,* vol. 73, pp. 232-247, 1999.

[14] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised Learning of Human Action Categories Using Spatial-Temporal Words," *Int'l J. Computer Vision,* vol. 79, no. 3, pp. 299-318, Mar. 2008.

[15] J. Niebles and L. Fei-Fei, "A Hierarchical Models of Shape and Appearance for Human Action Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2007.

[16] Z. Laptev and T. Lindeberg, "Space-Time Interest Points," *Proc. IEEE Int'l Conf. Computer Vision,* Oct. 2003.

[17] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[18] A. Oikonomopoulous, I. Patras, and M. Pantic, "Spationtemporal Saliency for Human Action Recognition," *Proc. IEEE Int'l Conf. Multimedia and Expo,* 2005.

[19] T. Mahmood, A. Vasilescu, and S. Sethi, "Recognition of Action Events from Multiple Video Points," *Proc. IEEE Workshop Detection and Recognition of Events in Video,* 2001.

[20] J. Liu, S. Ali, and M. Shah, "Recognizing Human Actions Using Multiple Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* June 2008.

[21] P. Scovanner, S. Ali, and M. Shah, "A 3-Dimensional SIFT Descriptor and Its Application to Action Recognition," *Proc. ACM Multimedia Conf.,* 2007.

[22] E. Shechtman and M. Irani, "Space-Time Behavior-Based Correlation—or—How to Tell If Two Underlying Motion Fields Are Similar without Computing Them?" *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 11, pp. 2045-2056, Nov. 2007.

[23] Y. Ke, R. Sukthankar, and M. Hebert, "Efficient Visual Event Detection Using Volumetric Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[24] T. Kim and R. Cipolla, "Canonical Correlation Analysis of Video Volume Tensors for Action Categorization and Detection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 8, pp. 1415-1428, Aug. 2009.

[25] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang, "Hierarchical Space-Time Model Enabling Efficient Search for Human Actions," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 19, no. 6, pp. 808-820, June 2009.

[26] C. Cedras and M. Shah, "Motion Based Recognition: A Survey," *Image and Vision Computing,* vol. 13, pp. 129-155, 1995.

[27] J. Aggarwal and Q. Cai, "Human Motion Analysis: A Review," *Computer Vision and Image Understanding,* vol. 73, pp. 428-440, 1999.

[28] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea, "Machine Recognition of Human Activities: A Survey," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 18, no. 11, pp. 1473-1488, Nov. 2008.

[29] J. Yuan, Z. Liu, and Y. Wu, "Discriminative Subvolume Search for Efficient Action Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[28] Note that in order for the fast nearest neighbor search method to be applicable, a generic basis such as sparse coding [95], [96], unlike query-dependent PCA basis, can be utilized at the expense of accuracy.

[30] O. Boiman, E. Shechtman, and M. Irani, "In Defense of Nearest-Neighbor Based Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[31] C.H. Lampert, M.B. Blaschko, and T. Hofmann, "Beyond Sliding Windows: Object Localization by Efficient Subwindow Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[32] P. Viola and M. Jones, "Robust Real-Time Object Detection," *Int'l J. Computer Vision,* vol. 57, no. 2, pp. 137-154, 2004.

[33] Y. Ke, R. Sukthankar, and M. Hebert, "Event Detection in Crowded Videos," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[34] A. Torralba, R. Fergus, and W. Freeman, "80 Million Tiny Images: A Large Data Set for Non-Parametric Object and Scene Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 11, pp. 1958-1970, Nov. 2008.

[35] B.C. Russell, A. Torralba, K.P. Murphy, and W.T. Freeman, "LabelMe: A Database and Web-Based Tool for Image Annotation," *Int'l J. Computer Vision,* vol. 77, nos. 1-3, pp. 157-173, 2008.

[36] J. Hays and A. Efros, "Scene Completion Using Millions of Photographs," *Proc. ACM SIGGRAPH,* 2007.

[37] H. Zhang, A. Berg, M. Maire, and J. Malik, "SVM-KNN: Discriminative Nearest Neighbor Classification for Visual Category Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2006.

[38] K. Grauman and T. Darrell, "The Pyramid Match Kernel: Efficient Learning with Sets of Features," *J. Machine Learning Research,* vol. 8, pp. 725-760, 2007.

[39] C. Yeo, P. Ahammad, K. Ramchandran, and S.S. Satry, "High-Speed Action Recognition and Localization in Compressed Domain Videos," *IEEE Trans. Circuits and Systems for Video Technology,* vol. 18, no. 8, pp. 1006-1015, Aug. 2008.

[40] W. Yang, Y. Wang, and G. Mori, "Human Action Recognition from a Single Clip Per Action," *Proc. Second Int'l Workshop Machine Learning for Vision-Based Motion Analysis,* 2009.

[41] H.J. Seo and P. Milanfar, "Training-Free, Generic Object Detection Using Locally Adaptive Regression Kernels," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 9, pp. 1688-1704, Sept. 2010.

[42] H.J. Seo and P. Milanfar, "Static and Space-Time Visual Saliency Detection by Self-Resemblance," *J. Vision,* vol. 9, no. 12, no. 15, pp. 1-27, 2009, http://journalofvision.org/9/12/15/ (doi:10.1167/9.12.15).

[43] H. Takeda, S. Farsiu, and P. Milanfar, "Kernel Regression for Image Processing and Reconstruction," *IEEE Trans. Image Processing,* vol. 16, no. 2, pp. 349-366, Feb. 2007.

[44] H. Takeda, S. Farsiu, and P. Milanfar, "Deblurring Using Regularized Locally-Adaptive Kernel Regression," *IEEE Trans. Image Processing,* vol. 17, no. 4, pp. 550-563, Apr. 2008.

[45] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-Resolution without Explicit Subpixel Motion Estimation," *IEEE Trans. Image Processing,* vol. 18, no. 9, pp. 1958-1975, Sept. 2009.

[46] Y. Fu, S. Yan, and T.S. Huang, "Correlation Metric for Generalized Feature Extraction," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 12, pp. 2229-2235, Dec. 2008.

[47] Y. Fu and T.S. Huang, "Image Classification Using Correlation Tensor Analysis," *IEEE Trans. Image Processing,* vol. 17, no. 2, pp. 226-234, Feb. 2008.

[48] C. Liu, "The Bayes Decision Rule Induced Similarity Measures," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 6, pp. 1086-1090, June 2007.

[49] D. Lin, S. Yan, and X. Tang, "Comparative Study: Face Recognition on Unspecific Persons Using Linear Subspace Methods," *Proc. IEEE Int'l Conf. Image Processing,* 2005.

[50] Y. Ma, S. Lao, E. Takikawa, and M. Kawade, "Discriminant Analysis in Correlation Similarity Measure Space," *Proc. IEEE Int'l Conf. Machine Learning,* 2007.

[51] J.W. Schneider and P. Borlund, "Matrix Comparison, Part 1: Motivation and Important Issues for Measuring the Resemblance between Proximity Measures or Ordination Results," *J. Am. Soc. for Information Science and Technology,* vol. 58, no. 11, pp. 1586-1595, 2007.

[52] P. Ahlgren, B. Jarneving, and R. Rousseau, "Requirements for a Cocitation Similarity Measure, with Special Reference to Pearson's Correlation Coefficient," *J. Am. Soc. for Information Science and Technology,* vol. 54, no. 6, pp. 550-560, 2003.

[53] J. Rodgers and W. Nicewander, "Thirteen Ways to Look at the Correlation Coefficient," *Am. Statistician,* vol. 42, no. 1, pp. 59-66, 1988.

[54] E. Shechtman and M. Irani, "Matching Local Self-Similarities across Images and Videos," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2007.

[55] J. Boulanger, C. Kervrann, and P. Bouthemy, "Space-Time Adaptation for Patch-Based Image Sequence Restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 29, no. 6, pp. 1096-1102, June 2007.

[56] A. Buades, B. Coll, and J.M. Morel, "Nonlocal Image and Movie Denoising," *Int'l J. Computer Vision,* vol. 76, no. 2, pp. 123-139, 2008.

[57] I.N. Junejo, E. Dexter, I. Laptev, and P. Perez, "View-Independent Action Recognition from Temporal Self-Similarities," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 33, no. 1, pp. 172-185, Jan. 2011.

[58] D. Weinland, E. Boyer, and R. Ronfard, "Action Recognition from Arbitrary Views Using 3D Exemplars," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[59] Y. Chen, J. Bi, and J. Wang, "MILES: Multiple-Instance Learning via Embedded Instance Selection," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 28, no. 12, pp. 1931-1947, Dec. 2006.

[60] H.J. Seo and P. Milanfar, "Generic Human Action Detection from a Single Example," *Proc. IEEE Int'l Conf. Computer Vision,* Sept. 2009.

[61] A. Klaser, M. Marszalek, and C. Schmid, "A Spatio-Temporal Descriptor Based on 3d-Gradients," *Proc. British Machine Vision Conf.,* 2008.

[62] I. Laptev and P. Perez, "Retrieving Actions in Movie," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[63] C. Tomasi and R. Manduchi, "Bilateral Filtering for Gray and Color Images," *Proc. IEEE Int'l Conf. Computer Vision,* 1998.

[64] R. Kimmel, *Numerical Geometry of Images.* Springer, 2003.

[65] N. Dalal and B. Triggs, "Histogram of Oriented Gradietns for Human Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2005.

[66] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[67] R. Duda, P. Hart, and D. Stork, *Pattern Classification,* second ed. John Wiley and Sons, Inc., 2000.

[68] Y. Ke and R. Sukthankar, "PCA-SIFT: A More Distinctive Representation for Local Image Descriptors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2004.

[69] M. Kendall and A. Stuart, "The Advanced Theory of Statistics, Volume 2: Inference and Relationship," *Griffin (Section 31.19),* 1973.

[70] M. Tatsuoka, *Multivariate Analysis.* Macmillan, 1988.

[71] T. Calinski, M. Krzysko, and W. Wolynski, "A Comparison of Some Tests for Determining the Number of Nonzero Canonical Correlations," *Comm. in Statistics, Simulation, and Computation,* vol. 35, pp. 727-749, 2006.

[72] Y. Benjamini and Y. Hochberg, "Controlling the False Discovery Rate: A Practical and Powerful Approach to Multiple Testing," *J. Royal Statistical Soc. Series B,* vol. 57, no. 1, pp. 289-300, 1995.

[73] F. Devernay, "A Non-Maxima Suppression Method for Edge Detection with Sub-Pixel Accuracy," Technical Report RR-2724, INRIA, 1995.

[74] N. Petkov and E. Subramanian, "Motion Detection, Noise Reduction, Texture Suppression and Contour Enhancement by Spatiotemporal Gabor Filters with Surround Inhibition," *Biological Cybernetics,* vol. 97, pp. 423-439, 2007.

[75] H. Jhuang, T. Serre, L. Wolf, and T. Poggio, "A Biologically Inspired System for Action Recognition," *Proc. IEEE Int'l Conf. Computer Vision,* Oct. 2007.

[76] D. Batra, T. Chen, and R. Sukthankar, "Space-Time Shapelets for Action Recognition," *Proc. IEEE Workshop Motion and Video Computing,* Jan. 2008.

[77] Z. Zhang, Y. Hu, S. Chan, and L.-T. Chia, "Motion Context: A New Representation for Human Action Recognition," *Proc. European Conf. Computer Vision,* 2008.

[78] K. Schindler and L. van Gool, "Action Snippets: How Many Frames Does Human Action Recognition Require," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[79] X. Sun, M. Chen, and A. Hauptmann, "Action Recognition via Local Descriptors and Holistic Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[80] A. Fathi and G. Mori, "Action Recognition by Learning Mid-Level Motion Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[81] M. Bregonzio, S. Gong, and T. Xiang, "Recognising Actions as Clouds of Space-Time Interest Points," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[82] J. Liu and M. Shah, "Learning Human Actions via Information Maximization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[83] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior Recognition via Sparse Spatio-Temporal Features," *Proc. IEEE Int'l Workshop Visual Surveillance and Performance Evaluation of Tracking and Surveillance,* Oct. 2005.

[84] A. Wong and J. Orchard, "A Nonlocal-Means Approach to Examplar-Based Inpainting," *Proc. IEEE Int'l Conf. Image Processing,* 2008.

[85] K. Rapantzikos, Y. Avrithis, and S. Kollias, "Dense Saliency-Based Spationtemporal Feature Points for Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[86] A. Bandopadhay and J. Fu, "Searching Parameter Spaces with Noisy Linear Constraints," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 1988.

[87] G. Medioni, I. Cohen, F. Bremond, S. Hongeng, and R. Nevatia, "Event Detection and Analysis from Video Streams," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 8, pp. 873-890, Aug. 2001.

[88] T. Veit, F. Cao, and P. Bouthemy, "Probabilistic Parameter-Free Motion Detection," *Prof. IEEE Conf. Computer Vision and Pattern Recognition,* June 2004.

[89] M. Marszalek, I. Laptev, and C. Schmid, "Actions in Context," *Prof. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[90] S.K. Divvala, D. Hoiem, J.H. Hays, A. Efros, and M. Hebert, "An Empirical Study of Context in Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2008.

[91] A. Oliva and A. Torralba, "The Role of Context in Object Recognition," *Trends Cognitive Science,* vol. 11, no. 12, pp. 520-527, Nov. 2007.

[92] H. Wang, M.M. Ullah, A. Klser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-Temporal Features for Action Recognition," *Proc. British Machine Vision Conf.,* 2009.

[93] L. Wolf, T. Hassner, and Y. Taigman, "The One-Shot Similarity Kernel," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.

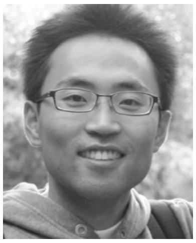
[94] L. Wolf, T. Hassner, and Y. Taigman, "Descriptor Based Methods in the Wild," *Proc. Faces in Real-Life Image Workshop in European Conf. Computer Vision,* 2008.

[95] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear Spatial Pyramid Matching Using Sparse Coding for Image Classification," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2009.

[96] H. Lee, A. Battle, R. Raina, and A.Y. Ng, "Efficient Sparse Coding Algorithms," *Proc. Advances in Neural Information Processing Systems,* 2006.

[97] N. Kumar, L. Zhang, and S.K. Nayar, "What Is a Good Nearest Neighbors Algorithm for Finding Similar Patches in Images," *Proc. European Conf. Computer Vision,* 2008.

[98] B. Kulis and K. Grauman, "Kernelized Locality-Sensitive Hashing for Scalable Image Search," *Proc. IEEE Int'l Conf. Computer Vision,* 2009.



Hae Jong Seo received the BS and MS degrees in electrical engineering from Sungkyunkwan University, Seoul, Korea, in 2005 and 2006, respectively. He is currently working toward the PhD degree in electrical engineering at the University of California, Santa Cruz. His research interests include the domain of image processing (denoising, interpolation, deblurring, and super-resolution) and computer vision (visual object recognition). He is a student member of the IEEE.

Peyman Milanfar received the BS degree in electrical engineering and mathematics from the University of California, Berkeley, in 1988, and the MS, EE, and PhD degrees in electrical engineering from the Massachusetts Institute of Technology, Cambridge, in 1990, 1992, and 1993, respectively. Until 1999, he was a senior research engineer at SRI International, Menlo Park, California. He is currently a professor of electrical engineering at the University of California, Santa Cruz. From 1998 to 2000, he was a consulting assistant professor of computer science at Stanford University, California, where he was also a visiting associate professor in 2002. His research interests include statistical signal, image processing, and inverse problems. He won the US National Science Foundation CAREER award, and the best paper award from the IEEE Signal Processing Society in 2010. From 1998 to 2001, he was an associate editor for the *IEEE Signal Processing Letters*, and was an associate editor for the *IEEE Transactions on Image Processing* from 2005-2010. He is currently on the editorial board of the *SIAM Journal of Imaging Science*, and *Image and Vision Computing*. He is a member of the Signal Processing Society Image, Video, and Multidimensional Signal Processing (IVMSP) Technical Committee. He is a fellow of the IEEE and a member of the IEEE Computer Society.