Detection of Human Actions from a Single Example

Hae Jong Seo and Peyman Milanfar
Electrical Engineering Department
University of California at Santa Cruz
1156 High Street, Santa Cruz, CA, 95064
{rokaf,milanfar}@soe.ucsc.edu

Abstract

We present an algorithm for detecting human actions based upon a single given video example of such actions. The proposed method is unsupervised, does not require learning, segmentation, or motion estimation. The novel features employed in our method are based on space-time locally adaptive regression kernels. Our method is based on the dense computation of so-called space-time local regression kernels (i.e. local descriptors) from a query video, which measure the likeness of a voxel to its spatio-temporal surroundings. Salient features are then extracted from these descriptors using principal components analysis (PCA). These are efficiently compared against analogous features from the target video using a matrix generalization of the cosine similarity measure. The algorithm yields a scalar resemblance volume; each voxel indicating the likelihood of similarity between the query video and all cubes in the target video. By employing non-parametric significance tests and non-maxima suppression, we accurately detect the presence and location of actions similar to the given query video. High performance is demonstrated on a challenging set of action data [8] indicating successful detection of multiple complex actions even in the presence of fast motions.

1. Introduction

A huge and growing number of videos are available online today. Human actions are one of the most important parts in movies, TV shows, and personal videos. Analysis of human actions in videos is considered a very important component in computer vision systems because of such applications as content-based video retrieval, visual surveillance, analysis of sports events and more.

The generic problem of interest addressed in this paper can be briefly described as follows: We are given a *single* "query" video of an action of interest (for instance a short ballet turn), and we are interested in detecting similar actions within other "target" videos. Detecting 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: 1) similar actions with different clothes, or in different illumination and background can result in a large appearance variation; 2) the same actions performed by two different people may look dissimilar in terms of action speed or frame rate of the video (See Fig. 1 (a)). There have been many efforts to model and recognize human actions broadly by means of parametric time-series approaches, frame-by-frame nonparametric approaches, and volumetric approaches. We refer the interested reader to [13] and references therein for a good summary. Volumetric approaches tend to outperform the other two approaches. These volumetric methods do not require background subtraction, motion estimation, and complex models of body configuration and kinematics. They tolerate variations in appearance, scale, rotation, and movement to some extent. Methods such as those in [5, 8] which aim at recognizing actions based solely on one query (what we shall call training-free) are very useful for video retrieval from the web. In these methods, a single query video is provided by users and every gallery video in the database is compared with the single query, posing a video-to-video matching problem.

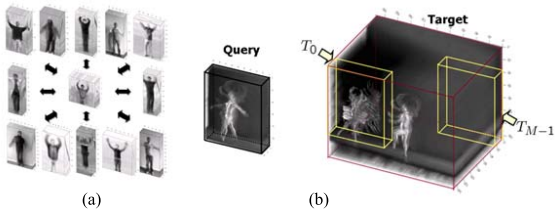

Figure 1. (a) A hand-waving action and possibly similar actions (b) Given a query video Q, we want to detect/localize actions of interest in a target video T. T can be divided into a set of overlapping cubes

Inspired by this trend toward training-free action analysis, this paper presents a novel training-free human action

2009 IEEE 12th International Conference on Computer Vision (ICCV)
978-1-4244-4419-9/09/$25.00 ©2009 IEEE

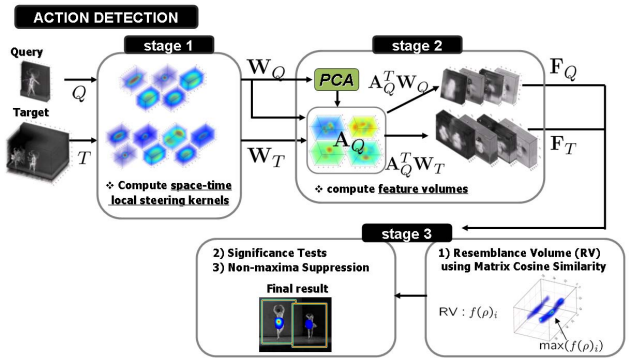

Figure 2. Action detection system overview (There are broadly three stages.)

detection framework. Our proposed method is based on the calculation and use of what we call *space-time local regression kernels* which are local weights computed directly from the given pixels in both the query and target videos. The original motivation to use these local regression kernels is the earlier successful work on adaptive kernel regression for image denoising, interpolation [9], deblurring [10], and (2-D) generic object detection [6]. Takeda et al. [11] extended the kernel regression framework to super-resolution by introducing space-time local steering kernels which capture the essential local behavior of a spatio-temporal neighborhood. The space-time local steering kernel (3-D LSK) is fundamentally based on the comparison of neighboring pixels in both space and time, thus it implicitly contains information about the local motion of the pixels across time, thus requiring no explicit motion estimation.

The space-time local steering kernel is defined as follows:

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

where $\mathbf{x}_s = [x_1, x_2, t]_s^T$ is the space-time coordinates, $s \in [1, \cdots, P]$, h is a global smoothing parameter, P is the total number of samples in a *space-time* local analysis window around a sample position at $\mathbf{x}$, and the matrix $\mathbf{C}_s \in \mathbb{R}^{(3\times 3)}$ is a covariance matrix estimated from a collection of first derivatives along spatial (x_1, x_2) and temporal (t) axes. The covariance matrix $\mathbf{C}_s$ modifies the shape and size of the local kernel in a way which robustly encodes the space-time local geometric structures present in videos. Normalization of this kernel function yields robustness to illumination, contrast, and color differences. For a more in depth analysis on local steering kernels, we refer the interested reader to [6, 9, 10, 11].

Very recently, Shechtman and Irani [7] introduced a space-time local self-similarity descriptor for action detection and showed performance improvement over their previous approach [8]. This (independently derived) local space-time self-similarity descriptor is a special case of our space-time local steering kernel and is also related to a number of other local data adaptive metrics such as Optimal Space-Time Adaptation (OSTA) [2] and Non-Local Means (NLM) [3] which have been used very successfully for video restoration in the image processing community. While the method proposed by Shechtman and Irani [7] is related to our method, their approach fundamentally differs from ours in the following respects: 1) Since the calculation of space-time local steering kernels is stable in the presence of uncertainty in the data [9], our approach is robust even in the presence of noise; 2) As opposed to [7] filtering out "non-informative" descriptors in order to reduce the time complexity, we automatically obtain the most salient feature volumes by applying Principal Components Analysis (PCA) to a collection of 3-D LSKs. From a practical standpoint, it is important to note that the proposed framework 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 pre-processing step or segmentation of the target video. Fig. 2 shows an overview of our proposed framework for action detection. To summarize the operation of the overall algorithm, we first compute the normalized space-time local steering kernels (3-D LSKs) $\mathbf{W}_Q, \mathbf{W}_T$ from both Q and T. In the second stage, we obtain the salient feature volumes $\mathbf{F}_Q, \mathbf{F}_T$ by projecting the descriptors $\mathbf{W}_Q, \mathbf{W}_T$ to a projection space $\mathbf{A}_Q$ derived from $\mathbf{W}_Q$. In the third stage, we compare the feature volumes $\mathbf{F}_{T_i}$ (=a chunk of $\mathbf{F}_T$ at i^{th} position) and $\mathbf{F}_Q$ using the Matrix Cosine Similarity measure. The final output is given after a sequence of significance tests, followed by non-maxima suppression [4].

This paper is organized as follows. In the next section, we provide further technical details about the various steps outlined above. In Section 3, we demonstrate the performance of the system with experimental results, and we conclude this paper in Section 4.

2. Technical Details

As outlined in the previous section, our approach to detect actions consists broadly of three stages. Assume that we are given a "target" video T and that we have a query video Q, where Q is generally smaller than T. The task at hand is to detect and locate cubes of T that are similar to Q. The first step is to calculate space-time local steering kernels (3-D LSKs). To be more specific, 3-D LSK function $K(\mathbf{x}_s - \mathbf{x})$ is densely calculated and normalized as follows:

$$W_I(\mathbf{x}_s - \mathbf{x}) = \frac{K_I(\mathbf{x}_s - \mathbf{x})}{\sum_{s=1}^{P} K_I(\mathbf{x}_s - \mathbf{x})}, \begin{cases} s = 1, \cdots, P, \\ I \in \{Q, T\}. \end{cases} \quad (2)$$

Fig. 3 illustrates what the normalized versions of 2-D LSKs and 3-D LSKs in various regions look like.

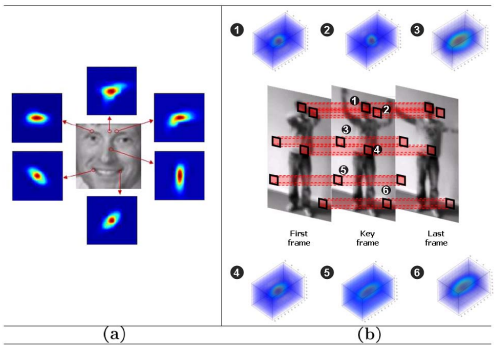

Figure 3. (a) Examples of 2-D LSK in various regions. (b) Examples of space-time local steering kernel (3-D LSK) in various regions. Note that key frame means the frame where the center of 3-D LSK is located.

In order to organize $W_Q(\mathbf{x}_s - \mathbf{x})$'s and $W_T(\mathbf{x}_s - \mathbf{x})$'s, which are densely computed from Q and T, let $\mathbf{W}_Q$,$\mathbf{W}_T$ be matrices whose columns are vectors $\mathbf{w}_Q$,$\mathbf{w}_T$, which are column-stacked (rasterized) versions of $W_Q(\mathbf{x}_s - \mathbf{x})$,$W_T(\mathbf{x}_s - \mathbf{x})$ respectively:

$$\begin{aligned} \mathbf{W}_Q &= [\mathbf{w}_Q^1, \cdots, \mathbf{w}_Q^n] \in \mathbb{R}^{P \times n}, \\ \mathbf{W}_T &= [\mathbf{w}_T^1, \cdots, \mathbf{w}_T^{n_T}] \in \mathbb{R}^{P \times n_T}. \end{aligned} \quad (3)$$

where n and n_T are the number of 3-D LSKs in the query video Q and the target video T respectively. As described in Fig. 2, the next step is to apply PCA to $\mathbf{W}_Q$ for dimensionality reduction and to retain only its salient characteristics. Applying PCA to $\mathbf{W}_Q$ we can retain the first (largest) d principal components[1] 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 $\mathbf{A}_Q$:

$$\begin{aligned} \mathbf{F}_Q &= [\mathbf{f}_Q^1, \cdots, \mathbf{f}_Q^n] = \mathbf{A}_Q^T \mathbf{W}_Q \in \mathbb{R}^{d \times n}, \\ \mathbf{F}_T &= [\mathbf{f}_T^1, \cdots, \mathbf{f}_T^{n_T}] = \mathbf{A}_T^T \mathbf{W}_T \in \mathbb{R}^{d \times n_T}. \end{aligned} \quad (4)$$

Fig. 4 illustrates the principal components in $\mathbf{A}_Q$ and shows what the features $\mathbf{F}_Q, \mathbf{F}_T$ look like for the ballet video case.

Very recently, Ali and Shah [1] 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 [1] in the sense that 1) motion information is implicitly contained in 3-D LSK while [1] explicitly computes optical flow; 2) Background subtraction was used as a pre-processing step while our method is fully automatic; 3) [1] employed multiple instance learning to a set of all kinematic modes in the dataset while our proposed method does not involve any training phase.

[1]Typically, d is selected to be a small integer such as 3 or 4 so that 80 to 90% 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.)

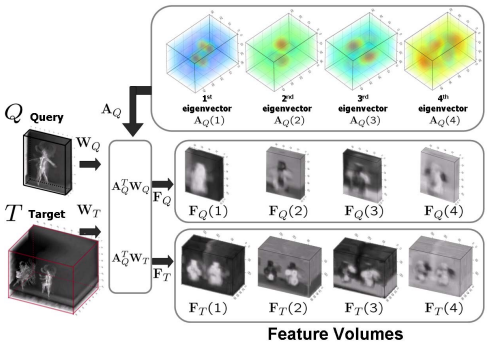

Figure 4. Ballet action : $\mathbf{A}_Q$ is learned from a collection of 3-D LSKs $\mathbf{W}_Q$, and Feature row vectors of $\mathbf{F}_Q$ and $\mathbf{F}_T$ are computed from query Q and target video T respectively. Eigenvectors and feature vectors were transformed to volume and up-scaled for illustration purposes.

The next step in the proposed framework is the measurement of a "distance" between the computed features $\mathbf{F}_Q, \mathbf{F}_{T_i}$. For this purpose, we employ the nonparametric detection framework [6] based on "Matrix Cosine Similarity".

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

$$\rho_i = < \overline{\mathbf{F}}_Q, \overline{\mathbf{F}}_{T_i} >_F = \mathrm{trace}\left(\frac{\mathbf{F}_Q^T \mathbf{F}_{T_i}}{\|\mathbf{F}_Q\|_F \|\mathbf{F}_{T_i}\|_F}\right) \in [-1, 1], \quad (5)$$

where, $\overline{\mathbf{F}}_Q = [\frac{\mathbf{f}_Q^1}{\|\mathbf{F}_Q\|_F}, \cdots, \frac{\mathbf{f}_Q^n}{\|\mathbf{F}_Q\|_F}]$ and $\overline{\mathbf{F}}_{T_i} = [\frac{\mathbf{f}_{T_i}^1}{\|\mathbf{F}_{T_i}\|_F}, \cdots, \frac{\mathbf{f}_{T_i}^n}{\|\mathbf{F}_{T_i}\|_F}]$. Equation 5 can be rewritten as a weighted average of the cosine similarities $\rho(\mathbf{f}_Q, \mathbf{f}_{T_i})$ between each pair of corresponding feature vectors (i.e., columns) in $\mathbf{F}_Q, \mathbf{F}_{T_i}$ as follows:

$$\rho_i = \sum_{\ell=1}^{n} \frac{\mathbf{f}_Q^{\ell\,T} \mathbf{f}_{T_i}^{\ell}}{\|\mathbf{F}_Q\|_F \|\mathbf{F}_{T_i}\|_F} = \sum_{\ell=1}^{n} \rho(\mathbf{f}_Q^{\ell}, \mathbf{f}_{T_i}^{\ell}) \frac{\|\mathbf{f}_Q^{\ell}\| \|\mathbf{f}_{T_i}^{\ell}\|}{\|\mathbf{F}_Q\|_F \|\mathbf{F}_{T_i}\|_F}. \quad (6)$$

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 $\mathbf{F}_Q, \mathbf{F}_{T_i}$. This measure [2] not only generalizes the cosine similarity, but also overcomes the disadvantages of the conventional Euclidean distance which

[2]We compute ρ_i over M target cubes and this can be efficiently implemented by column-stacking the matrices $\mathbf{F}_Q, \mathbf{F}_{T_i}$ and simply computing the cosine similarity between two long column vectors as follows:

$$\begin{aligned} \rho_i &\equiv \sum_{\ell=1}^{n} \sum_{j=1}^{d} \frac{f_Q^{(\ell,j)} f_{T_i}^{(\ell,j)}}{\sqrt{\sum_{\ell=1}^{n} \sum_{j=1}^{d} |f_Q^{(\ell,j)}|^2} \sqrt{\sum_{\ell=1}^{n} \sum_{j=1}^{d} |f_{T_i}^{(\ell,j)}|^2}}, \\ &= \rho(\mathrm{colstack}(\mathbf{F}_Q), \mathrm{colstack}(\mathbf{F}_{T_i})) \in [-1, 1], \end{aligned}$$

where $f_Q^{(\ell,j)}, f_{T_i}^{(\ell,j)}$ are elements in ℓ^{th} vector $\mathbf{f}_Q^{\ell}$ and $\mathbf{f}_{T_i}^{\ell}$ respectively, and $\mathrm{colstack}(\cdot)$ means an operator which column-stacks a matrix.

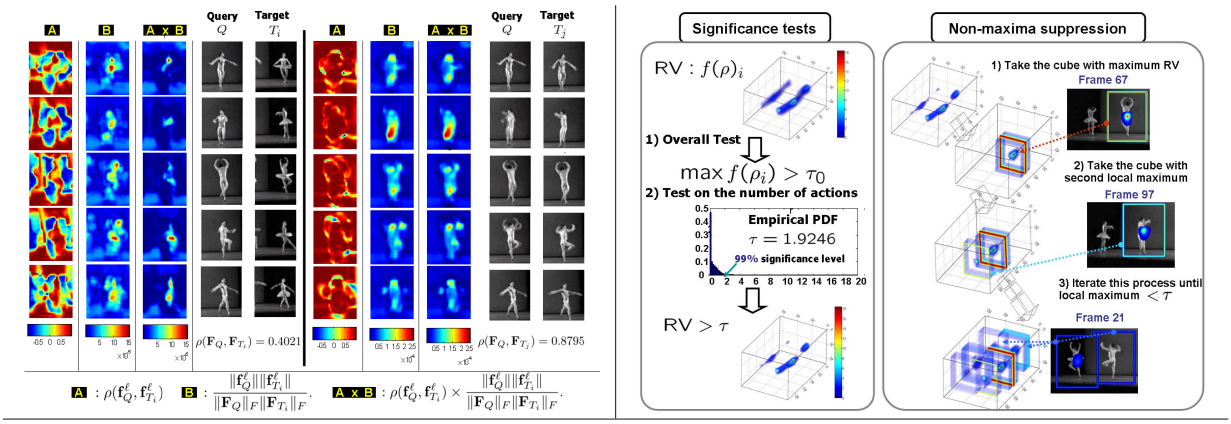

Figure 5. Left: Examples of A) $\rho(\mathbf{f}_Q^\ell, \mathbf{f}_{T_i}^\ell)$: cosine similarity, B) $\frac{\|\mathbf{f}_Q^\ell\|\|\mathbf{f}_{T_i}^\ell\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}$: weights, and A×B) $\rho(\mathbf{f}_Q^\ell, \mathbf{f}_{T_i}^\ell)\frac{\|\mathbf{f}_Q^\ell\|\|\mathbf{f}_{T_i}^\ell\|}{\|\mathbf{F}_Q\|_F\|\mathbf{F}_{T_i}\|_F}$: weighted cosine similarity. Note that query and target are same as those in Fig. 2(Left). Right: two significance tests and non-maxima suppression [4]are described.

is sensitive to outliers. Fig. 5(Left) shows examples of the computation of the MCS, which indicate that it provides a reliable measure of similarity.

It is worth noting that Shechtman and Irani [8] proposed 3-D 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 measure, MCS is better motivated, more appropriate, and more general than their global consistency measure for action detection.

The next step is to generate a so-called "resemblance volume" (RV), which will be a volume of voxels indicating the likelihood of similarity between Q and T at each spatio-temporal position. As for the final test statistic comprising the values in the resemblance volume, 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 using the function $f(\cdot)$ as follows:

$$\text{RV} : f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}, \quad i = 0, \cdots, M-1. \tag{7}$$

From a quantitative point of view, we note that $f(\rho_i)$ is essentially the Lawley-Hotelling Trace statistic [12], which is used as an efficient test statistic for detecting correlation between two data sets.

Next, we employ a two-step significance test as shown in Fig 5 (Right). The first is an overall threshold (τ_0) on the RV to decide whether there is any sufficiently similar action present in the target video at all. If the answer is yes at sufficiently high confidence, we would then want to know how many actions of interest are present in the target video and where they are. Therefore, we need two thresholds: an overall threshold [3] τ_o as mentioned above, and a threshold [4] τ to detect the (possibly) multiple occurrences of the same action in the target video.

After the two significance tests with τ_o, τ are performed, we employ the idea of non-maxima suppression [4] for the final detection. We take the volume region with the highest $f(\rho_i)$ value and eliminate the possibility that any other action is detected within some radius[5] of the center of that volume region again. This enables us to avoid multiple false detections of nearby actions already detected. Then we iterate this process until the local maximum value falls below the threshold τ. Fig. 5 (Right) shows a graphical illustration of significance tests and non-maxima suppression [4].

For the sake of completeness, the overall pseudo-code for the algorithm is given in **Algorithm** 1.

3. Experimental Results

Our method detects the presence and location of actions similar to the given query and provides a series of bound-

[3] In a typical scenario, we set the overall threshold τ_o to be, for instance, 0.96 which is about 50% of variance in common (i.e., $\rho^2 = 0.49$). In other words, if the maximal $f(\rho_i)$ is just above 0.96, we decide that there exists at least one action of interest.

[4] We employ the idea of nonparametric testing. We compute an empirical probability density function (PDF) from M samples $f(\rho_i)$ and we set τ so as to achieve, for instance, a 99 % ($\alpha = 0.99$) significance level in deciding whether the given values are in the extreme (right) tails of the distribution. This approach is based on the assumption that in the target video, most cubes do not contain the action of interest (in other words, action of interest is a relatively rare event), and therefore, the few matches will result in values which are in the tails of the distribution of $f(\rho_i)$.

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

Algorithm 1 Training-free generic action detection

Q : Query video, T : Target video, τ_o : Overall threshold, α : Confidence level, P : Size of space-time local steering kernel (3-D LSK) cube.

Stage1 : **Compute Descriptors**

Construct $\mathbf{W}_Q, \mathbf{W}_T$ which are a collection of normalized 3-D LSK associated with Q, T.

Stage2 : **Feature Representation**

1) Apply PCA to $\mathbf{W}_Q$ and obtain projection space $\mathbf{A}_Q$ from its top d eigenvectors.

2) Project $\mathbf{W}_Q$ and $\mathbf{W}_T$ onto $\mathbf{A}_Q$ to construct $\mathbf{F}_Q$ and $\mathbf{F}_T$.

Stage3

1) **Compute Matrix Cosine Similarity**

for every target cube T_i, where $i \in [0, \cdots, M-1]$ **do**

$$\rho_i = < \frac{\mathbf{F}_Q}{\|\mathbf{F}_Q\|_F}, \frac{\mathbf{F}_{T_i}}{\|\mathbf{F}_{T_i}\|_F} >_F \text{ and (RV)} : f(\rho_i) = \frac{\rho_i^2}{1-\rho_i^2}.$$

end for

Then, find $\max f(\rho_i)$.

2) **Significance tests**

i) If $\max f(\rho_i) > \tau_o$, go on to the next test. Otherwise, there is no action of interest in T.

ii) Threshold RV by τ which is set to achieve 99 % confidence level (α = 0.99) from the empirical PDF of $f(\rho_i)$.

3) **Non-maxima suppression**

Apply non-maxima suppression [4] to RV until the local maximum value is below τ.

ing cubes with resemblance volume embedded around detected actions. Note that no background/foreground segmentation is required in the proposed method. This method can also handle modest amount of variations in rotation (up to ± 15 degree), and spatial and temporal scale change (up to $\pm 20\%$). In practice, once given Q and T, we downsample both Q and T by some factor of (3, here) in order to reduce the time-complexity. We then compute 3-D LSK of size 3×3 (space) $\times 7$ (time) as descriptors so that every space-time location in Q and T yields a 63-dimensional local descriptor $\mathbf{W}_Q$ and $\mathbf{W}_T$ respectively. The smoothing parameter h for computing 3-D LSKs was set to 2.1. We end up with $\mathbf{F}_Q, \mathbf{F}_T$ by reducing dimensionality from 63 to $d = 4$ and then, we obtain RV by computing the MCS measure between $\mathbf{F}_Q, \mathbf{F}_T$. The threshold τ for each test example was determined by the confidence level $\alpha = 0.99$. We applied our method to 3 different examples : *i.e.* detecting 1) walking people, 2) ballet turn actions, and 3) multiple actions in one video. Shechtman and Irani [8] have tested their method on these videos using the same query and [5, 7] also tested their methods on some of these videos. We achieved similar (or even better) performance as compared to the methods in [5, 7, 8]. It is worth noting here that the other action detection methods [5, 7, 8] did not provide either threshold values or describe how they selected threshold values in reporting detection performance. On the other hand, the threshold values are automatically chosen in our algorithm with respect to the confidence level as explained earlier.

Fig. 6(A) shows the results of searching for instances of walking people in a target beach video (460 frames of 180×360 pixels). The query video contains a very short walking action moving to the right (14 frames of 60×70 pixels) and has a background context which is not the beach scene. In order to detect walking actions in either direction, we used two queries (Q and its mirror-reflected version) and generated two RVs. By voting the higher score among values from two RVs at every space-time location, we ended up with one RV which includes correct locations of walking people in the correct direction. Fig. 6(A) (a) shows a few sampled frames from Q. In order to provide better illustration of T, we divided T into 3 non-overlapping sections. Fig. 6(A) (b) and (c) represent each part of T and its corresponding RV respectively. Red color represents higher resemblance while blue color denotes lower resemblance values. Fig. 6(A) (d) and (e) show a few frames from T, with RV and bounding boxes superimposed on them respectively.

Fig. 6 (B) shows the results of detecting ballet turning action in a target ballet video (284 frames of 144×192 pixels). The query video contains a single turn of a male dancer (13 frames of 90×110 pixels). Fig. 6(B) (a) shows a few sampled frames from Q. Next, Fig. 6(B) (b) and (c) represent each part of T and its corresponding RV respectively. Fig. 6(B) (d) and (e) show a few frames from T with resemblance volumes superimposed on it respectively. Most of the turns of the two dancers (a male and a female) were detected even though this video contains very fast moving parts and relatively large variability in spatial scale and appearance (the female dancer wearing a skirt) as compared to the given query Q. We observed that one of the female dancer turning actions was missed because of large spatial scale variation as compared to the given Q. However, we can easily deal with this problem by either adjusting the significance level or using multi-scale approach as done in [6]. The detection result of the proposed method on this video outperforms that in [5, 8] and compares favorably to that in [7].

Fig. 6(C) shows the results of detecting 4 different actions ("walk", "wave", "clap", and "jump") which occur simultaneously in a target video (120 frames of 288×360 pixels). Four query videos were matched against the target video independently. Fig. 6(C) (a) and (b) show a few sampled frames from Q and T respectively. White boxes in Fig. 6(C) (a) represent actual regions used for the query. The resulting RVs are shown in Fig. 6(C) (c). In all the above examples, we used the same parameters. It is evident, based on all the results above, that the proposed training-free action detection based on 3-D LSK works well and is robust to modest variations in spatio-temporal scale.

Our system is designed with detection accuracy as a high priority. A typical run of the object detection system 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 3-D LSKs from Q and T respectively, which needs to be computed

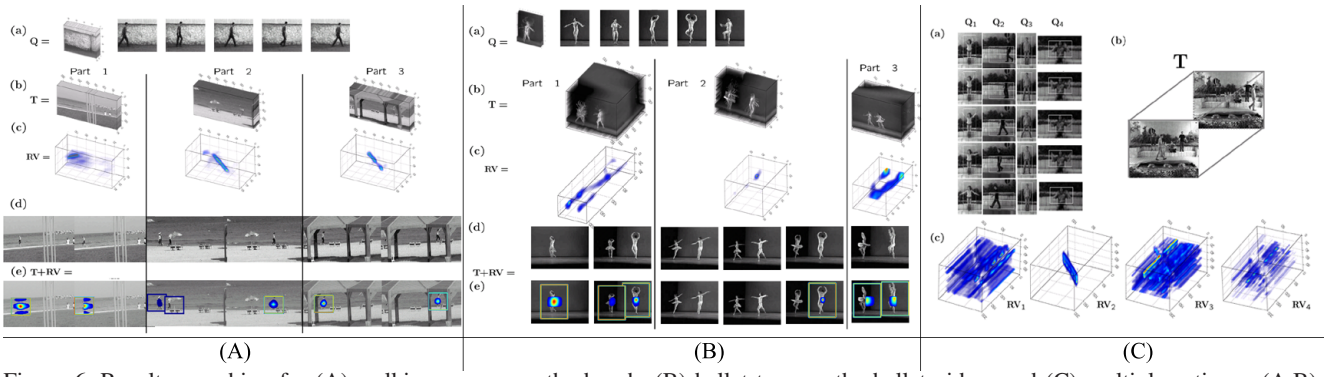

Figure 6. Results searching for (A) walking person on the beach, (B) ballet turn on the ballet video, and (C) multiple actions. (A,B): (a) query video (a short walk clip) (b) target video (c) resemblance volumes (RV) (d) a few frames from T (e) frames with resemblance volume on top of it. (C): (a) four different short video queries. Note that white boxes represent actual query regions (b) target video T (c) resemblance volumes (RV)s with respect to each query.

only once.) There are many factors that affect the precise timing of the calculations, such as query size, complexity of the video, and LSK size. Our system runs in Matlab but could be easily implemented using multi-threads or parallel programming as well as General Purpose GPU for which we expect a significant gain in speed.

4. Conclusion and Discussion

In this paper, we have proposed a novel action detection algorithm by employing *space-time local steering kernels* (3-D LSKs); and by using a training-free nonparametric detection scheme based on "Matrix Cosine Similarity" (MCS). The proposed method can automatically detect in the target video the presence, the number, as well as location of actions similar to the given query video. The proposed method is practically appealing because it is nonparametric. The proposed framework is general enough as to be extendable to action categorization using a nearest neighbor classifier along with an automatic action cropping method as similarly done in [5]. Improvement of the computational complexity of the proposed method is also a direction of future research worth exploring.

5. Acknowledgment

This work was supported by AFOSR Grant FA 9550-07-01-0365.

References

[1] S. Ali and M. Shah. Human action recognition in videos using kinematic features and multiple instance learning. *Accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2008.

[2] J. Boulanger, C. Kervrann, and P. Bouthemy. Space-time adaptation for patch-based image sequence restoration. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:1096–1102, June 2005.

[3] A. Buades, B. Coll, and J. M. Morel. Nonlocal image and movie denoising. *International Journal of Computer Vision*, 76(2):123–139, 2008.

[4] F. Devernay. A non-maxima suppression method for edge detection with sub-pixel accuracy. *Technical report, INRIA*, (RR-2724), 1995.

[5] H. Ning, T. Han, D. Walther, M. Liu, and T. Huang. Hierarchical space-time model enabling efficient search for human actions. *IEEE Transactions on Circuits and Systems for Video Technology, in press*, 2008.

[6] H. J. Seo and P. Milanfar. Training-free, generic object detection using locally adaptive regression kernels. *Accepted for publication in IEEE Transactions on Pattern Analysis and Machine Intelligence*, June 2009.

[7] E. Shechtman and M. Irani. Matching local self-similarities across images and videos. *In Proc. of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1–8, June 2007.

[8] 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 Transactions on Pattern Analysis and Machine Intelligence*, 29:2045–2056, November 2007.

[9] H. Takeda, S. Farsiu, and P. Milanfar. Kernel regression for image processing and reconstruction. *IEEE Transactions on Image Processing*, 16(2):349–366, February 2007.

[10] H. Takeda, S. Farsiu, and P. Milanfar. Deblurring using regularized locally-adaptive kernel regression. *IEEE Transactions on Image Processing*, 17:550–563, April 2008.

[11] H. Takeda, P. Milanfar, M. Protter, and M. Elad. Super-resolution without explicit subpixel motion estimation. *Accepted for publication in IEEE Transactions on Image Processing*, 2009.

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

[13] P. Turaga, R. Chellappa, V. Subrahmanian, and O. Udrea. Machine recognition of human activities: A survey. *IEEE Transactions on Circuits and Systems for Video Technology*, 18:1473–1488, November 2008.