Human Action Recognition in Unconstrained Videos by Explicit Motion Modeling

Yu-Gang Jiang, Qi Dai, Wei Liu, Xiangyang Xue, and Chong-Wah Ngo

***Abstract*— Human action recognition in unconstrained videos is a challenging problem with many applications. Most state-of-the-art approaches adopted the well-known bag-of-features representations, generated based on isolated local patches or patch trajectories, where motion patterns, such as object–object and object-background relationships are mostly discarded. In this paper, we propose a simple representation aiming at modeling these motion relationships. We adopt global and local reference points to explicitly characterize motion information, so that the final representation is more robust to camera movements, which widely exist in unconstrained videos. Our approach operates on the top of visual codewords generated on dense local patch trajectories, and therefore, does not require foreground–background separation, which is normally a critical and difficult step in modeling object relationships. Through an extensive set of experimental evaluations, we show that the proposed representation produces a very competitive performance on several challenging benchmark data sets. Further combining it with the standard bag-of-features or Fisher vector representations can lead to substantial improvements.**

***Index Terms*— Human action recognition, trajectory, motion representation, reference points, camera motion.**

I. Introduction

HUMAN action recognition has received significant research attention in the field of image and video analysis. Significant progress has been made in the past two decades, particularly with the invention of local invariant features and the bag-of-features representation framework. For example, currently a popular and very common solution that produces competitive accuracy on popular benchmarks is to employ the bag-of-features representation on top of spatial-temporal interest points (STIP) [1], [2] or the temporal trajectories of frame-level local patches (e.g., the dense trajectories by Wang et al. [3], [4]).

Manuscript received December 4, 2014; revised May 12, 2015 and June 18, 2015; accepted June 19, 2015. Date of publication July 14, 2015; date of current version July 24, 2015. This work was supported in part by the National 863 Program of China under Grant 2014AA015101, in part by the National Science Foundation of China under Grant 61201387, in part by the Science and Technology Commission of Shanghai Municipality, China, under Grant 13PJ1400400, and in part by the EU FP7 QUICK Project under Grant PIRSES-GA-2013-612652. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Béatrice Pesquet-Popescu.

Y.-G. Jiang, Q. Dai, and X. Xue are with the School of Computer Science, Fudan University, Shanghai 200433, China (e-mail: ygj@fudan.edu.cn; daiqi@fudan.edu.cn; xyxue@fudan.edu.cn).

W. Liu is with the IBM T. J. Watson Research Center, NY 10598 USA (e-mail: weiliu@us.ibm.com).

C.-W. Ngo is with the Department of Computer Science, City University of Hong Kong, Hong Kong (e-mail: cscwngo@cityu.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2015.2456412

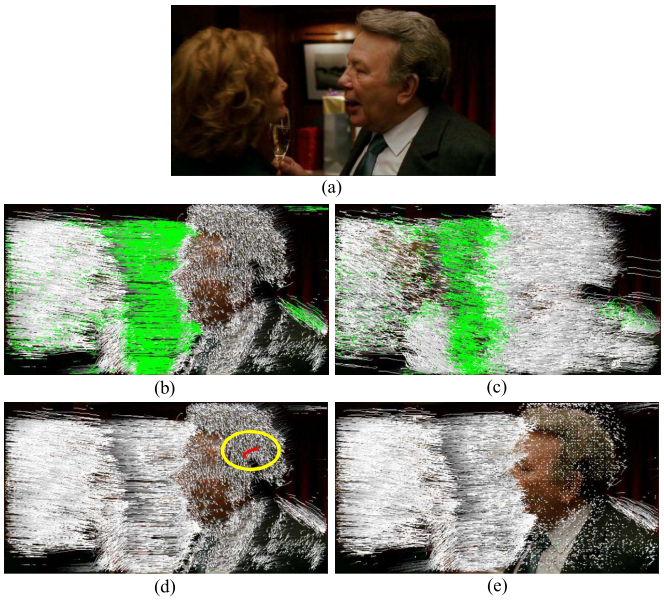


Fig. 1. Illustration of the proposed approach. (a) A video frame of a *kissing* action. (b) Local patch trajectories, with the largest trajectory cluster shown in green. (c) Amended trajectories by using the mean motion of the green cluster as a global reference point; See details in Section IV-A. (d) The original patch trajectories, with a trajectory on a person's head shown in red (circled). (e) Amended trajectories by using the motion of the red trajectory as a local reference point; The relative motion patterns w.r.t. the red trajectory (as visualized in (e)) are quantized into a pairwise trajectory-codeword representation; See details in Section IV-B. This figure is best viewed in color.

One disadvantage of the typical bag-of-features approach is that it ignores the motion relationships among foreground objects or between the objects and the background scene. Apparently such motion patterns are important for recognizing many human actions and thus should be incorporated into a recognition system. This is particularly necessary when the target videos are captured under *unconstrained* environment with severe camera motion, which often hinders the acquisition of the real motion of foreground objects (e.g., consider the case of a camera moving at the same pace with a person).

In this paper, we propose an approach to model the motion relationships among moving objects and the background. We adopt two kinds of reference points to explicitly characterize complex motion patterns in the unconstrained videos, in order to alleviate the effect incurred by camera movement. Figure 1 illustrates our proposed approach. Tracking of local frame patches is firstly performed to capture the pixel motion of the local patches. With the trajectories, we then adopt a simple clustering method to identify the dominant motion, which is used as a *global* motion reference point to calibrate

the motion of each trajectory. As will be discussed later, although the identified global motion reference may not be accurate, it helps uncover at least some motion relationships in the scene. In addition, to further capture the relationships of moving objects, we treat each trajectory as a *local* motion reference point, which leads to a rich representation that encapsulates both trajectory descriptors and pairwise relationships. Specifically, the trajectory relationships are encoded by trajectory codeword pairs in the final representation. Since each trajectory codeword represents a unique (moving) visual pattern (e.g., a part of an object), the motion among objects/background can be captured in this representation. With the local reference points, the resulted representation is naturally robust to camera motion as it only counts the relative motion between trajectories, which is considered as the main contribution of this work.

Although very simple in its form, our approach has the following advantages. First, it has been widely acknowledged that motion patterns, particularly the interaction of moving objects, are very important for recognizing human actions (e.g., the distance changes between two people in action "kissing"), and the modeling of such motion interactions in unconstrained videos is difficult due to camera motion. Using trajectory-based pairwise relative motion is a desirable solution to uncover the real object movements in videos. On the other hand, we notice that there have been several works exploring pairwise relationships of local features, where generally only one type of relationship such as co-occurrence or proximity was modeled, using methods like the Markov process. In contrast, our approach explicitly integrates the descriptors of patch trajectories as well as their relative spatial location and motion pattern. Both the identification of the reference points and the generation of the final representation are very easy to implement, and very competitive action recognition accuracy can be achieved on several challenging benchmarks. Moreover, we also show that the proposed motion representation can be reduced to very low dimensions for efficient classification with no performance degradation.

The rest of this paper is organized as follows. We first briefly discuss related works in Section II, and then introduce the tracking of local patches, which is the basis of our representation, in Section III. Section IV elaborates the proposed approach and Section V discusses an extensive set of experiments and results. Finally, Section VI concludes this paper.

II. Related Works

Human action recognition has been extensively studied in the literature, where most efforts have been devoted to the design of good feature representations. Local features, coupled with the bag-of-features framework, are currently the most popular way to represent videos [2], [5]. In addition to the bag-of-features, several alternative feature coding methods have been proposed, such as the Fisher Vectors [6], VLAD [7] and the super vectors [8], some of which have also been successfully used in human action recognition.

Recent works on video representation may be divided into the following two categories. The first category extracts or learns spatial-temporal local features, which are spatial-temporal volumes typically capturing representative regions like the boundary of a moving object. Many efforts in this category focused on the design of good local volume detectors/descriptors [1], [9]–[13] or feature learning algorithms [14]–[16]. A few others focused on the selection or sampling of more effective local volumes [17], [18] or higher-level attribute representations [19], [20]. Instead of directly using the spatial-temporal local features in the bag-of-features representation, the other category performs temporal tracking of local patches and then computes features on top of the local patch trajectories [3], [21]–[26]. In the following we mainly focus our discussion on the trajectory-based approaches, which are more related to this work. Readers are referred to [27]–[29] for comprehensive surveys of action recognition techniques, particularly those focusing on the design of recognition models.

In [21], Uemura et al. extracted trajectories of SIFT patches with the KLT tracker [30]. Mean-Shift based frame segmentation was used to estimate dominating plane in the scene, which was used for motion compensation. Messing et al. [22] computed velocity histories of the KLT-based trajectories for action recognition. The work of [26] also adopted the KLT tracker, and proposed representations to model inter-trajectory proximity. They used a different set of trajectories and did not specifically focus on alleviating the negative effect of camera motion. Wang et al. [23] modeled the motion between KLT-based keypoint trajectories, without considering trajectory locations. Spatial and temporal context of trajectories was explored in [25], where the authors adopted an elegant probabilistic formulation and focused on modeling context, not directly on alleviating the negative effect of camera motion. Raptis and Soatto [24] and Gaidon et al. [31] proposed tracklet, which emphasizes more on the local casual structures of action elements (short trajectories), not the pairwise motion patterns. In [32], the authors extended [24] to a mid-level representation by grouping trajectories based on appearance and motion information, leading to a set of discriminative action parts, which are essentially identified by the found trajectory clusters. The idea of grouping trajectories is similar to our method in the identification of the global reference points, but the way of using the trajectory clusters is totally different. Another work by Kliper-Gross et al. [33] proposed a representation called motion interchange pattern to capture local motions at every frame and image location. The authors also proposed a suppression mechanism to overcome camera motion, which—as will be shown later—offers much lower recognition accuracies than our approach. In addition, Wang et al. [34] performed trajectory-based modeling using Bayesian models and Wu et al. [35] proposed to use decomposed Lagrangian particle trajectories for action recognition. Several other authors also explored object-level trajectories [36], [37] for video content recognition.

A representative approach of trajectory-based motion modeling is from Wang et al. [3], [4], [38], who generated trajectories based on dense local patches and showed that the dense trajectories significantly outperform KLT tracking of sparse local features (e.g., the SIFT patches). Very promising

results have been observed on several human action recognition benchmarks. They found that long trajectories are often unstable and therefore adopt short trajectories that only last 15 frames. To cope with camera motion, they extended Dalal's motion boundary histogram (MBH) [39] as a very effective trajectory-level descriptor. MBH encodes the gradients of optical flow, which are helpful for canceling constant camera motion, but cannot capture the pairwise motion relationships. Jain et al. [40] extended the work by considering the compensation of dominant motion in both tracking and encoding stages, which is different as Wang's work only used the MBH to consider the issue in the encoding stage. A new descriptor called Divergence-Curl-Shear (DCS) was also proposed based on differential motion scalar features. In a recent work of Wang and Schmid [38], feature matching across frames was adopted to estimate a homography that helps cancel global motion, such that the effect of camera motion can be alleviated. This method is similar to our global reference point based method, which may fail when moving objects like humans dominate the scene [38]. In addition, it cannot explicitly capture the pairwise motion relationships between objects, which can be achieved by our local reference point based method. Furthermore, Jung et al. [41] also clustered trajectories for feature modeling, but did not adopt the idea of dense trajectories, which are more effective. Piriou et al. [42] explored a method for computing the dominant global image motion in the scene using probabilistic models.

Our representation integrates trajectory descriptors with the pairwise trajectory locations as well as motion patterns. It not only differs from the previous inter-trajectory descriptors in its design, but also generates competitive recognition accuracies compared to the state-of-the-art approaches on challenging benchmarks of realistic videos. This work extends upon a previous conference publication [43] by adding new experiments on a large dataset, more comparative analysis with alternative methods and baselines, and extra discussions throughout the paper. In addition, we also discuss and evaluate a solution to successfully reduce the dimensionality of the proposed representation, which is very important particularly when dealing with large datasets.

III. Generating Dense Trajectories

The proposed representation is generated based on local patch trajectories. In this paper, we adopt the dense trajectory approach by Wang et al. [3], [4] as it has been shown effective on several benchmarks. We briefly describe the idea of dense trajectories as follows. Notice that our approach is not limited to this specific trajectory generation method and can be applied on top of any local patch trajectories.

The first step is to sample local patches densely from every frame. We follow the original paper to sample patches in 8 spatial scales with a grid step size of 5 pixels. Tracking is then performed on the densely sampled patches by median filtering in a dense optical flow field. Specifically, a patch $P_t = (x_t, y_t)$ at frame number t is tracked to another patch P_{t+1} in the following frame by

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (F \times \omega)|_{(\bar{x}_t, \bar{y}_t)}, \qquad (1)$$

where F is the kernel of median filtering, $\omega = (u_t, v_t)$ denotes the optical flow field, and $(\bar{x}_t, \bar{y}_t)$ is the rounded position of P_t. To compute the dense optical flow, the algorithm of [44] is adopted, which is publicly available from the OpenCV library. A maximum value of trajectory length is set here to avoid a drifting problem that often occurs when trajectories are long, and 15 frames were found to be a suitable choice. According to the authors, this is considered as an effective strategy to make sure the trajectories are mostly correct. To further improve tracking accuracy, trajectories with sudden large displacements are removed from the final set.

After the trajectories are generated, we can compute several descriptors to encode either the trajectory shape or the local motion and appearance within a space-time volume around the trajectories. In [3], the shape of a trajectory is described in a very straightforward way by concatenating a set of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. In order to make the trajectory shape (TrajShape) descriptor invariant to scale, the shape vector is further normalized by the overall magnitude of motion displacements:

$$\text{TrajShape} = \frac{(\Delta P_t, \ldots, \Delta P_{t+L-1})}{\sum_{i=t}^{t+L-1} \| \Delta P_i \|}, \qquad (2)$$

where $L = 15$ is the length (frame number) of the trajectories.

Three descriptors are used to encode the local motion and appearance around a trajectory: Histograms of Oriented Gradients (HoG) [45], Histograms of Optical Flow (HOF), and the MBH. HOG captures local appearance information, while HOF and MBH encode local motion patterns. To get a fine-grained description of local structures, the space-time volumes (spatial size 32×32 pixels) around the trajectories are divided into 12 equal-sized 3D grids (spatially 2×2 grids, and temporally 3 segments). For HOG, gradient orientations are quantized into 8 bins, which is a standard setting used in the literature. HOF has 9 bins in total, with one additional zero bin compared to HOG. With these parameters the final representation has 96 dimensions for HOG and 108 dimensions for HOF. As described earlier, MBH computes a histogram based on the derivatives of optical flow. Specifically, the derivatives are computed separately on both horizontal and vertical components. Like HOG, 8 bins are used to quantize orientations, and as there are two motion boundary maps from the derivatives along two directions, the MBH descriptors have $96 \times 2 = 192$ dimensions. By using the derivatives of optical flow, MBH is able to cope with global motion and only captures local relative motion of pixels. This is quite useful for the analysis of realistic videos "in the wild" with severe camera motion, but the pairwise motion relationships are not captured in MBH. The parameters for computing the descriptors are chosen based on an empirical study conducted in [3]. All the three descriptors have been shown effective in human action recognition studies, particularly on benchmarks of unconstrained videos [2], [3], [5], [40], [46].

Notice that the method was recently augmented by Wang and Schmid in [38]. The general flow of computing the features remains the same, except that, as aforementioned in Section II, global motion is estimated and trajectories determined to be on the background are excluded from computing

the representations. In the experiments, we will show results of our approach on both the original trajectories [3] and the new improved trajectories [38].

IV. Trajectory-Based Motion Modeling

In this section, we introduce the proposed trajectory-based motion modeling approach. We first elaborate a method that utilizes global reference points to alleviate the effect of camera motion specifically for improving the TrajShape descriptor. After that we describe a trajectory-based motion representation that uses each individual trajectory as a local reference point. This representation integrates the location and motion relationships of the local patch trajectories as well as their local appearance descriptors. Because of the use of relative motion, it is not sensitive to camera movements. Between the two ideas using global and local reference points respectively, the latter representation is considered as a more important contribution. We elaborate both of them in the following.

A. Improved Shape Descriptor With Global Reference Points

Identifying the global motion in complex unconstrained videos is not an easy task. Typical solutions include foreground-background separation [21] and video stabilization [47], etc. In this paper we present a very simple solution by clustering the motion patterns of all the found trajectories on the scene. The dominant pattern from the largest clusters is treated as reference points to calibrate motion, so that the effect of global/camera motion can be alleviated. Specifically, given a trajectory $\mathcal{T}$ with start position P_t on frame t, the overall motion displacement of the trajectory is

$$\Delta\mathcal{T} = (P_{t+L-1} - P_t) = (x_{t+L-1} - x_t, y_{t+L-1} - y_t). \quad (3)$$

Notice that, because the length of the dense trajectories has been restricted to only 15 frames (0.5 seconds for a 30 fps video), most trajectories are fairly straight lines with small angle deviations from the overall motion direction. To verify this, we compute the angles between the moving directions of all the segments of each trajectory and the "overall" motion direction (between the starting and ending points) of the trajectory. Results are visualized in Figure 2. We see that almost all the segments are within 90 degrees and more than half of them are within 45 degrees, indicating that the "shape" of the trajectories is mostly very straight. Because of this observation, we do not need to further split a trajectory and only adopt the overall displacement to represent its motion.

The motion pattern similarity of two trajectories is computed by $\mathcal{S}(\mathcal{T}_u, \mathcal{T}_v) = ||\Delta\mathcal{T}_u - \Delta\mathcal{T}_v||$. With this similarity measure, we cluster trajectories starting within each 5-frame temporal window of a video, and empirically produce five trajectory clusters per window. Note that the TrajShape descriptor also can be used to compute similarities and generate the trajectory clusters, but we have observed that the 2D displacement vectors show similar results at a much faster speed.

It is difficult to predict which cluster contains trajectories on the background scene and which one refers to a moving object. For instance, if the foreground objects are small, then the largest cluster may refer to the background scene. However when the foreground objects are very large and occupy most area of a frame, trajectories from the largest cluster may mostly come from the objects. This problem was also found in the recent work of Wang and Schmid [38], who used a more complex method of feature matching to identify the global motion. In the experiments, we empirically choose the top-three largest clusters (out of a total of five clusters) and compute the mean motion displacement of each cluster as a candidate dominant motion direction. We found that this is more reliable than using a single cluster (see evaluations of this choice in Section V-D). Figure 3 visualizes the trajectory clustering results on two example frames, where the top-three clusters are shown in different color. Note that, for some special motions like camera zooming in or out, the induced image motion is a divergence field, and the resulting trajectories are straight segments but of any orientations. In this rare case using more clusters might be helpful, but three was just found to be a reliable number in general.

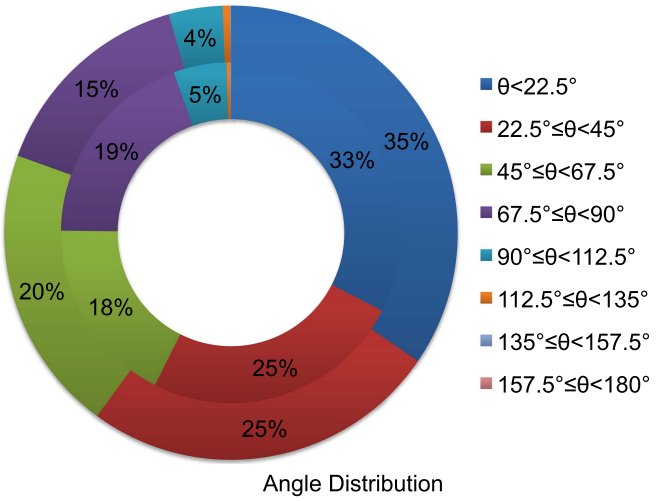
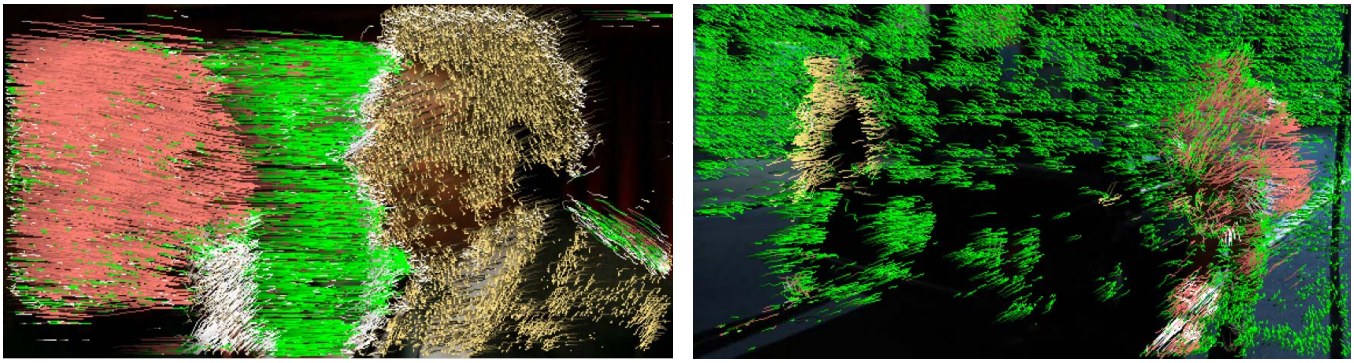


Fig. 2. Distribution of the angles between the motion directions of trajectory segments and the overall trajectory motion direction. Outer circle shows statistics of the Hollywood2 dataset and inner circle plots that of the Olympic Sports dataset. This figure is best viewed in color. See texts for more explanations.

Given a trajectory cluster $\mathcal{C}$, let the mean motion displacement be $\Delta\mathcal{C} = (\Delta\bar{x}_c, \Delta\bar{y}_c)$. The displacement of a trajectory between two nearby frames within the corresponding 5-frame window is adjusted to $\Delta P'_t = \Delta P_t - \Delta\mathcal{C}/15$, where $\Delta\mathcal{C}/15$ is the determined global motion. We then update the displacement of all the trajectories in the next 5-frame window and further proceed until the end of the video. With this compensation by the estimated dominant motion, the TrajShape descriptor in Equation (2) can be adjusted to:

$$\text{TrajShape}' = \frac{(\Delta P'_t, \ldots, \Delta P'_{t+L-1})}{\sum_{i=t}^{t+L-1} \|\Delta P'_i\|}, \quad (4)$$

where TrajShape$'$ is the improved descriptor. Using the mean motion displacements of the three largest clusters, a trajectory has a set of three TrajShape$'$ descriptors, each adjusted by the motion pattern of one cluster. The method of converting sets



(a) (b)

Fig. 3. Visualization of trajectory clustering results. Trajectories from the top-three largest clusters are visualized in green, light red and yellow respectively, while the remaining ones are shown in white. (a) Two people kissing; (b) Two people getting out of a car. This figure is best viewed in color.

of TrajShape′ to measure video similarity will be described later.

It is worth further explaining that, if the cluster corresponds to the background, the adjustment of $\Delta\mathcal{C}/15$ represents the canceling of the camera motion. While when the cluster corresponds to a large moving object such as a human subject dominating the scene, the adjustment can be explained as estimating the relative motion of all the other components to the subject, which can also alleviate the effect of camera motion, simply because of the use of relative motion. In this case, as the reference point (i.e., the mean motion of the cluster) corresponds to a large area of the scene, it can still be considered as a *global* reference point, in contrast to the local reference points discussed in the following.

B. Motion Modeling With Local Reference Points

The global reference points can be used to alleviate the effect of camera motion. However, the resulted representation can hardly capture the motion relationships between moving objects, which motivates the proposal of local reference points in this subsection, which is considered as the main contribution of this work.

We start from discussing the quantization of the appearance descriptors, and will elaborate the use of local reference points afterwards. Since the number of trajectories varies across different videos, a common way to generate fixed-dimensional video representation is to use the well-known *visual codewords*, which are cluster centers of the trajectory descriptors. This is the same with the classical bag-of-features framework based on static SIFT descriptors [48]. In our representation, we also use visual codewords as the abstract units to encode the pairwise motion relationships. For each type of trajectory descriptor (e.g., HOF), a codebook of n codewords is generated by clustering the descriptors using k-means.

We use every trajectory as a local reference point to characterize *relative motion*, so that camera motion may be canceled and the motion relationships between objects can be encoded. Specifically, given two trajectories $\mathcal{T}_u$ and $\mathcal{T}_v$, the relative motion (with $\mathcal{T}_v$ as the local reference point) can be computed by

$$\mathcal{M}(\mathcal{T}_u, \mathcal{T}_v) = \Delta\mathcal{T}_u - \Delta\mathcal{T}_v, \tag{5}$$

where $\Delta\mathcal{T}$ can be computed by Equation 3. Note that for most cases it is not needed to use the dominant motion $\Delta\mathcal{C}$ to further cancel global motion here, since the relative motion is already robust to camera movement. However, for some special types of camera movements like zoom in or out, or when the objects are with different depth in the scene, computing relative motion in the above form is not sufficient to fully cancel camera motion, and therefore using the global reference points is still helpful. We will show in the experiments that the improved trajectory shape descriptor TrajShape′ is complementary to this pairwise motion representation and can be combined to achieve higher recognition accuracies.

Figure 4 visualizes the generation of the motion feature representation with local reference points, named as TrajMF. The relative motion $\mathcal{M}(\mathcal{T}_u, \mathcal{T}_v)$ of two trajectories is quantized in a way that incorporates very rich information, including trajectory neighborhood appearance descriptors, motion direction and magnitude, as well as the relative location of the two trajectories. The neighborhood appearance information is encoded in TrajMF because this representation is constructed based on the trajectory codewords, which are generated using the appearance descriptors like HOG. In the final representation as shown in the middle of Figure 4, we only consider the overall relative motion between codeword pairs, so that the dimension of TrajMF is fixed. All the pairwise trajectory motion patterns are mapped/accumulated to their corresponding codeword pairs. In other words, given a pair of trajectories, we first find their corresponding codeword pair, and then add the quantized motion vector (explained in the next paragraph) to that particular entry. Because a visual codeword may represent a (moving) local pattern of an object or a part of the background scene, the final TrajMF representation implicitly encodes object-object or object-background motion relationships.

The motion pattern between two trajectories is quantized into a compact vector, according to both the relative motion direction and the relative location of the trajectory pair. Formally speaking, let $Q(\cdot)$ be the quantization function

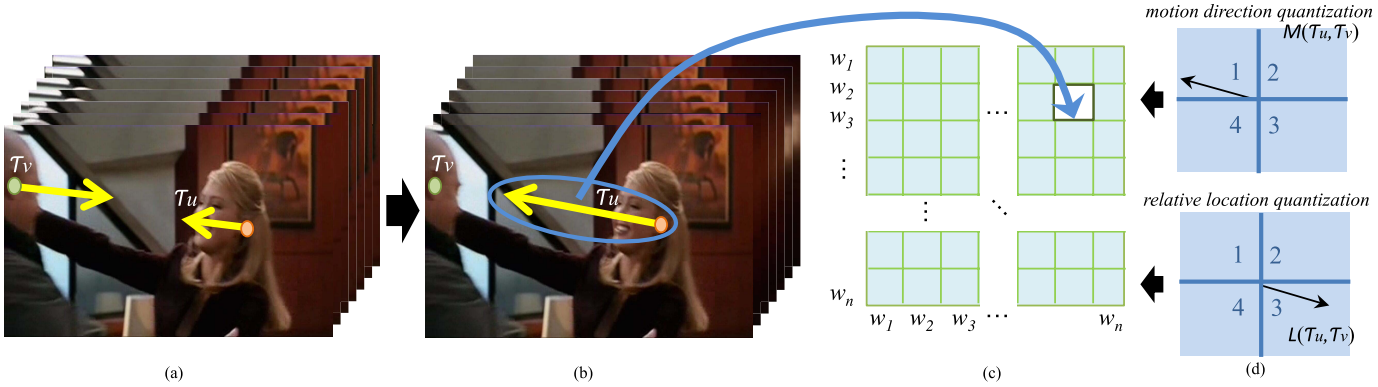


Fig. 4. An illustration of the trajectory-based motion feature representation, named as TrajMF. The motion of two trajectories in (a) is converted to relative motion ($\mathcal{T}_u$ relative to $\mathcal{T}_v$ in this example) in (b), which is then mapped to an entry of a codeword-based matrix representation (c), by quantizing the local descriptors of the two trajectories. The motion pattern between each codeword pair, i.e., an entry in (c), is described by a 16-d vector, based on the relative motion direction and relative location of all the trajectory pairs falling into that entry. The quantization maps for generating the 16-d vector are shown in (d). The resulted representation is a vector that concatenates all the values in (c), which is in very high dimension but can be mapped into a compact space using dimension reduction techniques. See texts for more explanations.

according to motion direction and relative location (see the quantization maps in Figure 4(c)), which outputs a quantization vector with all zeros except the bit that an input trajectory pair should be assigned to. The motion vector of a codeword pair (w_p, w_q) is then defined as the summation of the motion vectors of all the trajectory pairs that fall into the codeword pair:

$$\mathbf{f}(w_p, w_q) = \sum_{\forall(\mathcal{T}_u,\mathcal{T}_v)\rightarrow(w_p,w_q)} Q(\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v), \mathcal{L}(\mathcal{T}_u,\mathcal{T}_v)) \cdot ||\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v)||, \tag{6}$$

where "$\rightarrow$" denotes the trajectory-to-codeword mapping, $||\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v)||$ is the magnitude of the relative motion, and

$$\mathcal{L}(\mathcal{T}_u,\mathcal{T}_v) = (\bar{P}_{\mathcal{T}_u} - \bar{P}_{\mathcal{T}_v}) = (\bar{x}_{\mathcal{T}_u} - \bar{x}_{\mathcal{T}_v}, \bar{y}_{\mathcal{T}_u} - \bar{y}_{\mathcal{T}_v})$$

indicates the relative location of the mean positions of two trajectories. In the experiments we use four bins to quantize both the motion direction and the relative location direction, and therefore $\mathbf{f}$ is 16-d. Evaluations of these parameters can be found in a later Section V. Concatenating $\mathbf{f}$ of all the codeword pairs, the final TrajMF representation has $\frac{n\times n}{2} \times 4 \times 4$ dimensions (n is the number of codewords), which is obviously very high. We discuss techniques to reduce the dimensions of TrajMF in the following subsection.

C. Dimension Reduction of TrajMF

The goal of dimension reduction is to improve the efficiency of recognition and reduce the usage of memory. We experimented with several dimension reduction methods to reduce the dimension of TrajMF. The first method came into our mind was to use data mining techniques [49], [50] for feature selection, which choose a subset of the entries in the TrajMF based on an estimation of discriminativeness in recognition. Another work we considered is [51], where the authors proposed product quantization to map high-dimensional inputs into low compact spaces. However, as will be shown in the experiments, our results indicate that both options are ineffective and the performance is always degraded.

We therefore decided to reduce the feature dimension with the simple principal components analysis (PCA). Naive PCA cannot be deployed in our case due to the high computational needs arised from the high dimensionality of the original features. We therefore adopt the EM-PCA approach proposed by Roweis [52], which was designed to be suitable for high dimensional data and large collections. We briefly introduce it below.

Consider a linear model that assumes an observed data sample $\boldsymbol{y} \in \mathbf{R}^p$ is generated by

$$\boldsymbol{y} = \boldsymbol{C}\boldsymbol{x} + \boldsymbol{v}, \tag{7}$$

where the k-dimensional latent variables $\boldsymbol{x} \in \mathbf{R}^k$ follow the unit normal distribution with zero mean ($p \geq k$). $\boldsymbol{C} \in \mathbf{R}^{p\times k}$ is the transformation matrix, and $\boldsymbol{v}$ is the noise vector.

We can view PCA as a limiting case when the noise covariance becomes infinitely small. So the model can be rewritten as $\boldsymbol{Y} = \boldsymbol{C}\boldsymbol{X}$ where $\boldsymbol{Y}$ is a matrix of the observed data and $\boldsymbol{X}$ is a matrix of the latent variables. The first k principal components can then be learned through the following EM algorithm [52]:

$$\mathbf{e}-\mathbf{step}: \quad \boldsymbol{X} = (\boldsymbol{C}^T\boldsymbol{C})^{-1}\boldsymbol{C}^T\boldsymbol{Y}$$
$$\mathbf{m}-\mathbf{step}: \quad \boldsymbol{C}^{new} = \boldsymbol{Y}\boldsymbol{X}^T(\boldsymbol{X}\boldsymbol{X}^T)^{-1}$$

It is an iterative process and the required storage space is $O(kp) + O(k^2)$, which is much smaller than the naive PCA solution.

D. Classification

The proposed representations can be used to convert videos to feature vectors, which are then used for action model learning and prediction. In this subsection we briefly discuss classifier choices for both the augmented trajectory shape descriptor and the TrajMF representation. For TrajShape′, we adopt the standard bag-of-features approach to convert a set of descriptors into a fixed-dimensional vector. Following [2] and [3], we construct a codebook of 4,000 codewords using k-means. All the three TrajShape′ descriptors of every trajectory are quantized together into a single 4,000-d histogram for

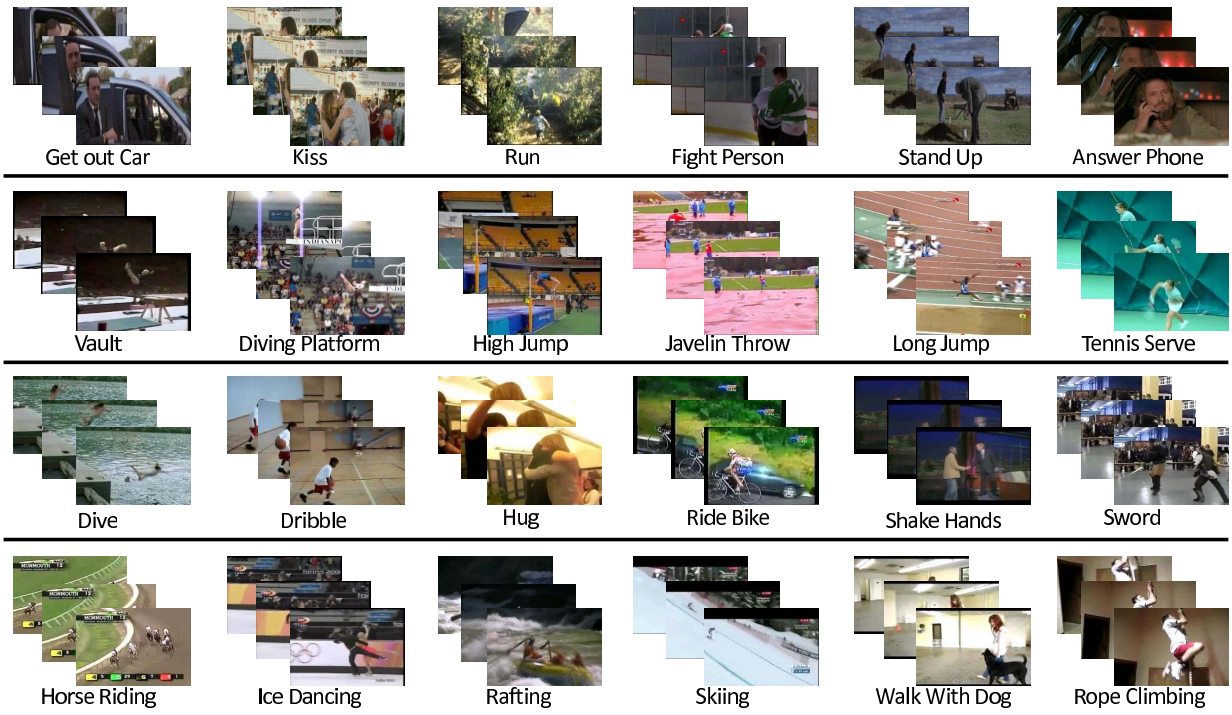


Fig. 5. Example frames of a few action classes in Hollywood2 (first row), Olympic Sports (second row), HMDB51 (third row) and UCF101 (bottom row) datasets. Videos in all the datasets were mostly captured under unconstrained environments with camera motion.

each video, which is used as the final representation. This is classified by the popular χ^2 kernel Support Vector Machines (SVM) due to its consistently good performance on histogram-like representations.

The TrajMF can be computed on top of any basic trajectory descriptors. We adopt all the three descriptors used in [3]: HOG, HOF, and MBH. For each type of trajectory descriptor, a separate TrajMF representation is computed. We evaluate both the original TrajMF and its dimension reduced version. As the dimension of the original TrajMF is very high, non-linear classifiers such as the χ^2 SVM are unsuitable due to speed limitation, and thus more efficient alternatives like the linear SVM are preferred. We will evaluate these popular kernel options in the experiments.

V. Experiments

A. Datasets and Evaluation

We conduct extensive experiments using four challenging datasets of realistic videos: Hollywood2 dataset [53], Stanford Olympic Sports dataset [54], HMDB51 dataset [47], and UCF101 dataset [55]. Many videos in these datasets contain camera motion and their contents are very diverse. Figure 5 gives some example frames from each of the datasets.

The first dataset is the widely adopted Hollywood2 [53], which contains 1,707 video clips collected from 69 Hollywood movies. The dataset is divided into a training set of 823 samples and a test set of 884 samples. 12 action classes are defined and annotated in this dataset, including answering phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. Each class is learned by a one-versus-all SVM classifier. Recognition performance is measured by average precision (AP) for a single class and mean AP (mAP) for the overall performance of all the classes.

The Olympic Sports dataset [54] has 783 clips and 16 action classes. So on average there are around 50 clips per class. The classes are high jump, long jump, triple jump, pole vault, gymnastics vault, shot put, snatch, clean jerk, javelin throw, hammer throw, discus throw, diving platform, diving springboard, basketball layup, bowling, and tennis serve. We adopt the provided train/test split by Niebles et al. [54], and use one-versus-all SVM for classification. Like Hollywood2, mAP is used as the performance measure.

The HMDB51 dataset was recently collected by Kuehne et al. [47], containing 6,766 video clips in total. There are 51 action classes, each with at least 101 positive samples. The action names can be found in Figure 6. We adopt the official setting of [47] to use three train/test splits and also the one-versus-all classifiers. Each split has 70 training and 30 test clips for each action class. Also following [47], we report mean classification accuracy over the three splits.

The last dataset is the UCF101 [55], which was collected by Soomro et al. and is currently the largest publicly available dataset for action recognition. The dataset has 101 action classes and 13320 video clips in total. Each category is grouped into 25 groups, with each group containing 4-7 videos. We adopt one-versus-all SVMs and the leave-one-group-out strategy, i.e., each time 24 groups are used for training and 1 for testing. We report the mean classification accuracy over the 25 train/test splits.

B. Results and Discussion

First, we report the performance of the proposed representations. We set the number of codewords n to 300, and use 4 bins to quantize both the motion direction and

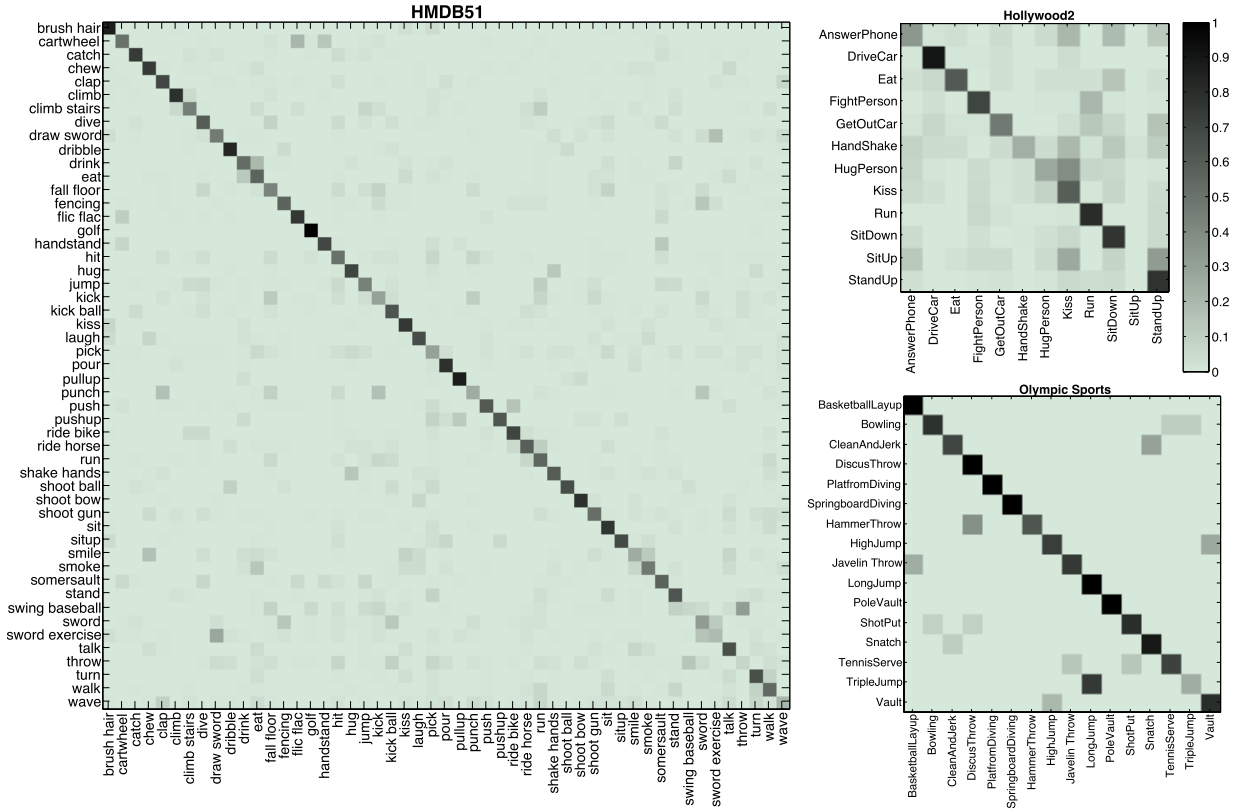


Fig. 6. Confusion matrices of the fusion results ("All combined") on the Hollywood2 (upper right), Olympic Sports (lower right) and HMDB51 (left) datasets.

TABLE I

PERFORMANCE OF BASELINES, OUR REPRESENTATIONS, AND THEIR COMBINED FUSION ON HOLLYWOOD2, OLYMPIC SPORTS, HMDB51 AND UCF101 DATASETS, USING THE **ORIGINAL DENSE TRAJECTORIES [3]**. THE "4 COMBINED" BASELINE RESULTS (USING FOUR FEATURES TRAJSHAPE, HOG, HOF AND MBH) ARE COMPUTED BASED ON THE **STANDARD BAG-OF-FEATURES**. "OUR 4 COMBINED" INDICATES THE FUSION RESULTS OF THE TRAJSHAPE′ AND THE THREE TRAJMF REPRESENTATIONS. "ALL COMBINED" INDICATES RESULTS FROM THE FUSION OF OUR REPRESENTATIONS AND THE BASELINE. NOTE THAT BETTER RESULTS ARE REPORTED THAN THE CONFERENCE VERSION [43] ON HMDB51 BECAUSE ONE-VS.-ALL SVM (NOT MULTI-CLASS SVM) IS ADOPTED FOLLOWING THE LITERATURES USING THIS BENCHMARK. FUSION IS DONE BY SIMPLY AVERAGING THE PREDICTIONS OF SEPARATE CLASSIFIERS

	Approach	**Hollywood2**	**Olympic Sports**	**HMDB51**	**UCF101**
Baseline results (bag-of-features)	TrajShape	49.3%	59.5%	28.3%	57.1%
	4 combined [3]	58.4%	74.3%	46.5%	78.3%
Our results	TrajShape′	50.2%	59.6%	29.8%	59.0%
	TrajMF-HOG	39.4%	66.7%	28.0%	58.9%
	TrajMF-HOF	42.3%	56.0%	27.6%	59.6%
	TrajMF-MBH	46.9%	74.6%	39.3%	72.6%
	Our 4 combined	55.6 %	77.6%	46.3%	79.2%
	All combined	**59.5%**	**80.6%**	**49.8%**	**80.7%**

the relative location, as depicted in Figure 4. The linear kernel SVM is adopted to classify the three original TrajMF representations before dimension reduction (each based on a different trajectory descriptor) and the χ^2 kernel SVM is used for the other representations. Later on we will evaluate the dimension reduced TrajMF, kernel choices, and also several key parameters.

Table I gives the results on the four datasets, using the original dense trajectory features [3]. In addition to discussing our proposed representations, we also present the results of the bag-of-features baselines using the same set of dense trajectory descriptors. Following the work of Wang et al. [3], in the bag-of-features representation, we use a codebook of 4000 words for each type of the trajectory descriptor. We use the source codes released by the authors to generate the dense trajectories and compute the basic descriptors, while the bag-of-features representation is based on our own implementation. As shown in the table, the amended trajectory shape descriptor TrajShape′ outperforms the original TrajShape, which validates the effectiveness of using the simple

TABLE II

PERFORMANCE OF BASELINES, OUR REPRESENTATIONS, AND THEIR COMBINED FUSION ON HOLLYWOOD2, OLYMPIC SPORTS, HMDB51 AND UCF101 DATASETS, USING THE **IMPROVED DENSE TRAJECTORIES [38]**. THE "4 COMBINED" BASELINE RESULTS ARE COMPUTED BASED ON THE **FISHER VECTOR CODING**. "OUR 4 COMBINED" INDICATES THE FUSION RESULTS OF THE TRAJSHAPE′ AND THE THREE TRAJMF REPRESENTATIONS. "ALL COMBINED" INDICATES RESULTS FROM THE FUSION OF OUR REPRESENTATIONS AND THE BASELINE. FUSION IS DONE BY SIMPLY AVERAGING THE PREDICTIONS OF SEPARATE CLASSIFIERS

	Approach	**Hollywood2**	**Olympic Sports**	**HMDB51**	**UCF101**
Baseline results (Fisher vector)	TrajShape	46.8%	75.5%	31.6%	63.8%
	4 combined [38]	63.3%	89.5%	55.8%	86.7%
Our results	TrajShape′	49.5%	77.0%	32.8%	65.1%
	TrajMF-HOG	38.0%	67.7%	27.4%	60.7%
	TrajMF-HOF	43.9%	70.3%	34.8%	66.8%
	TrajMF-MBH	48.6%	77.1%	41.6%	74.4%
	Our 4 combined	56.7 %	80.5%	48.6%	80.1%
	All combined	**65.1%**	**91.0%**	**57.0%**	**87.3%**

TABLE III

PERFORMANCE OF THE DIMENSION REDUCED FEATURES ON HOLLYWOOD2, OLYMPIC SPORTS, HMDB51 AND UCF101 DATASETS, USING BOTH THE ORIGINAL AND THE IMPROVED DENSE TRAJECTORIES. OVERALL THE RESULTS ARE VERY CLOSE TO THAT OF THE HIGH DIMENSIONAL FEATURES

	Approach	**Hollywood2**	**Olympic Sports**	**HMDB51**	**UCF101**
Original dense trajectories [3]	Our 4 combined	54.6%	79.2%	46.7%	77.1%
	All combined	**59.5%**	**81.2%**	**49.4%**	**80.2%**
Improved dense trajectories [38]	Our 4 combined	55.2%	80.6%	48.4%	78.5%
	All combined	**65.4%**	**91.0%**	**57.3%**	**87.2%**

clustering-based method to cancel global motion. On the large UCF101 dataset, the performance is boosted from 57.1% to 59.0%, which is very encouraging considering simplicity of our method and the complexity of the dataset. In contrast, recently Jain et al. [40] proposed ω-Trajdesc descriptor based on a different motion compensation method and achieved 51.4% on Hollywood2 and 32.9% on HMDB51, which are slightly higher to ours.

For the TrajMF representation, we also observe very promising performance. Combining our TrajShape′ and TrajMF representations ("Our 4 combined") generates better results than the "4 combined" baseline of [3] on the Olympic Sports and UCF101 datasets. On Hollywood2 and HMDB51 the performance is similar or slightly lower than the bag-of-features baseline. We underline that the TrajMF representation is not a direct replacement of the baseline bag-of-features. In fact they are complementary because they emphasize on different aspects of the visual contents. More specifically, TrajMF encodes in particular the motion relationship information and the bag-of-features captures visual appearances. As shown in the table, further combining our representations with the baseline ("All combined") gives substantial improvements on all the four datasets. This confirms the fact that the TrajMF representations are very complementary to the standard bag-of-features, and should be used together for improved action recognition performance. Note that in Table I, the results on HMDB51 are based on one-vs.-all SVMs following existing works, which are found to be better than that reported in the previous conference paper [43], where multi-class SVMs were used. This is probably due to the fact that popular multi-class SVMs use a top-down hierarchical classification scheme, which is less optimal compared with the binary one-vs.-all SVMs that train an optimal separation plane solely for each class.

We also evaluate our approach on the improved dense trajectories [38]. Results are summarized in Table II. The improved version uses feature matching to estimate camera motion, so that the effect from global camera movement can be alleviated. This is similar to our goal of using the motion reference points, but our TrajMF has an additional capability of modeling the motion relationships as discussed earlier. As shown in the table, it is interesting to observe that the TrajShape′ still outperforms the baseline with clear margin. This is probably because we use three global reference points instead of one as [38], which also confirms the fact that global camera motion is very difficult to be estimated accurately. The combination of our TrajMF representations with the baseline offers similar performance gains to that on the original dense trajectories, leading to very competitive results on all the four evaluated datasets (row "All combined"). This again verifies the effectiveness of our proposed representations. Note that in this experiment, the well-known Fisher vector coding is adopted for the baseline, which is significantly better than the bag-of-features [38].

Next we evaluate the performance of the dimension reduced TrajMF using EM-PCA. For the Hollywood2, HMDB51 and UCF101, the dimensionality is reduced to 1,500, while for the Olympic Sports, we use 500 because there are only 783 videos in this dataset. We will evaluate the effect of dimensionality later. Linear kernel SVM is also adopted in this experiment. Table III summarizes the results. Compared with the results in Table I and Table II, we can see that the performance remains almost the same after dimension reduction. For the "4 combined" results, we even observe better performance in several cases, which is probably because the PCA process is able to remove noises from the original features. These results confirm that EM-PCA is suitable for compressing the TrajMF features. Although very simple, we consider this as an

TABLE IV

PERFORMANCE OF SEVERAL KERNEL OPTIONS FOR THE TRAJMF REPRESENTATION, USING THE ORIGINAL DENSE TRAJECTORIES. "OUR 4 COMBINED" DENOTES THE COMBINATION OF THE 4 REPRESENTATIONS DERIVED FROM USING THE MOTION REFERENCE POINTS, AND "ALL COMBINED" IS THE COMBINATION OF OUR 4 REPRESENTATIONS AND THE BASELINE BAG-OF-FEATURES

	Kernels	**Hollywood2**	**Olympic Sports**	**HMDB51**	**UCF101**
Our 4 combined	χ^2	58.1%	77.7%	47.4%	78.8%
	HI	58.6%	76.9%	46.8%	78.6%
	Linear	55.6%	77.6%	46.3%	79.2%
All combined	χ^2	60.1%	79.2%	49.0%	80.4%
	HI	**60.3%**	78.9%	48.4%	80.2%
	Linear	59.5%	**80.6%**	**49.8%**	**80.7%**
Dimension Reduced our 4 combined	RBF	56.2%	77.9%	46.5%	77.6%
	Linear	54.6%	79.2%	46.7%	77.1%
Dimension Reduced all combined	RBF	59.4%	79.0%	48.5%	80.0%
	Linear	**59.5%**	**81.2%**	**49.4%**	**80.2%**

TABLE V

PERFORMANCE OF VARIOUS DIMENSION REDUCTION METHODS ON HOLLYWOOD2, OLYMPIC SPORTS, HMDB51 AND UCF101 DATASETS. FOR BOTH MUTUAL INFORMATION AND PRODUCT QUANTIZATION, THE TRAJMF FEATURES ARE REDUCED TO 2,000 DIMENSIONS

	Approach	**Hollywood2**	**Olympic Sports**	**HMDB51**	**UCF101**
EM-PCA	Our 4 combined	55.2%	80.6%	48.4%	78.5%
	All combined	**65.4%**	**91.0%**	**57.3%**	**87.2%**
Mutual Information	Our 4 combined	40.1%	76.8%	38.6%	75.6%
	All combined	63.4%	89.4%	55.3%	86.3%
Product Quantization	Our 4 combined	49.2%	74.5%	40.2%	74.2%
	All combined	65.0%	88.0%	55.9%	86.6%

important ingredient of the approach as the original TrajMF features are in high dimensions which may prevent its use in some applications. Figure 6 further shows the confusion matrices of the fusion results on Hollywood2, Olympic Sports and HMDB51. Errors mostly occur between classes that are visually similar, like "drink" and "eat" in HMDB51, and "HugPerson" and "Kiss" in Hollywood2.

We also report the performance of several popular classifier kernels, in order to identify the most suitable kernel for the proposed TrajMF representation. We only discuss results on the original dense trajectories in this experiment, as the observations from the improved trajectories are mostly the same. Specifically, we evaluate χ^2, HI (Histogram Intersection) and Linear kernel SVMs for the original TrajMF representations, and the χ^2 kernel is replaced by RBF kernel for the dimension reduced TrajMF representation that has negative values, on which the χ^2 kernel is not applicable. The HI kernel is also dropped for the dimension reduced TrajMF since it is no longer a histogram. Instead of reporting the performance of each single TrajMF classified by different kernels, we report the fusion performance due to space limitation. Fusion performance is also more important as we care more on the best possible results that can be attained on these challenging datasets. Table IV shows the results. Across all the fusion results, we use fixed kernel options for the baseline bag-of-features representation and the trajectory shape descriptors, and deploy different kernels on the TrajMF. We see that the performance of these kernels does not differ significantly under all the settings. More interestingly, the linear kernel is observed to be very robust for both the original and the dimension reduced TrajMF representations, offering similar or better results than the nonlinear kernels on all the datasets. This is very appealing as the linear kernel is much more efficient.

C. Comparative Studies

In this subsection, we first compare our results with alternative solutions for dimension reduction and for alleviating the effect of camera motion, followed by a comparison with recent state-of-the-art results.

We first compare results of a few dimension reduction methods. For this, we consider two alternative methods as discussed in Section IV-C. One is using mutual information to select a subset of discriminative dimensions, and the other method is Product Quantization [51], which decomposes the input space into a Cartesian product of low dimensional subspaces that can be quantized separately, where the number of the subspaces is equal to the number of the target dimensions. In our implementation, we use 8 binary values to quantize each subspace which is converted to an integer between 0 and 255 in the dimension-reduced representation. We fix the final dimension of both methods to 2,000, which is higher than 1,500 from the EM-PCA as we found 2,000 is a better number for both the compared methods.

Results are summarized in Table V, where we show both our results of "Our 4 combined" and the "All combined" which further includes fusion with the Fisher Vector baseline on the improved trajectories. We see that for all the datasets EM-PCA is clearly better. This is probably because PCA can preserve most valuable information from the original feature, while Mutual Information incurs significant information loss by selecting only a small fraction of the dimensions. Product Quantization is better than Mutual Information but its way of quantizing the features into binary vectors also loses

TABLE VI
SPEED AND MEMORY COST BEFORE AND AFTER DIMENSION REDUCTION, ON THE HOLLYWOOD2 DATASET USING THE TRAJMF-HOG FEATURE. DIMENSION REDUCTION HELPS REDUCE BOTH COST SIGNIFICANTLY. THE TRAINING PROCESS OF THE EM-PCA COSTS 885S, AND REDUCING THE DIMENSION OF ONE FEATURE ONLY REQUIRES 0.035S. SPEED IS MEASURED AS THE SINGLE THREAD RUNNING TIME ON A REGULAR MACHINE WITH INTEL CORE i7 4770 3.4GHz CPU AND 32 GB RAM

	Before	After
Model training time (12 classes)	42269s	9.9s
Prediction time (per test sample)	8.2s	0.002s
Memory usage (prediction with 12 models)	16GB	9.9MB

TABLE VII
COMPARISON WITH A VIDEO STABILIZATION-BASED APPROACH, USING THE HOLLYWOOD2 DATASET AND THE ORIGINAL DENSE TRAJECTORIES. OUR APPROACH GENERATES SIMILAR PERFORMANCE TO THE DENSE TRAJECTORY BASELINE ON STABILIZED VIDEOS, BUT IS MORE EFFICIENT

Approach	Performance
Baseline on original HMDB51	46.5%
Baseline on the stabilized version of HMDB51	50.5%
Our approach ("All combined") on original HMDB51	49.4%

more information. Table VI further compares the speed and memory cost before and after using dimension reduction, where we can clearly see the advantages of reducing the feature dimensions.

To alleviate the effect of camera motion, we consider an expensive yet very powerful stabilization-based method. We experiment with the HMDB51 dataset, which has a stabilized version obtained by applying a standard image stitching method [47], [56], where camera motion is basically fully canceled. We re-run the dense trajectory baseline on the stabilized HMDB51 dataset. The results are shown in Table VII. We see that our method gives very close performance to the new baseline on stabilized videos, which are extremely expensive to be generated using the method of [56]. This is very encouraging and clearly proves the effectiveness of our method in dealing with camera motion.

In Table VIII, we further compare our results with several state-of-the-art approaches. On Hollywood2, we obtain 2.4% gain over [38] (w/o Human Detection, HD), which used Fisher vectors on the improved dense trajectories. This performance gain is nontrivial considering that our result is based on the same set of trajectories and [38] already has a function of canceling global motion based on homography estimation. In other words, the only added information comes through the modeling of relative motion relationships in the TrajMF representation. Compared to [38] using human detection (i.e., w/HD) to compensate camera motion, our result is still 1.1% higher, which is very encouraging as the HD process is very expensive. Compared with a recent hierarchical spatio-temporal feature learning approach [15], a significant gain of 12.1% is achieved. The approach of Jain et al. [40] considered motion compensation in both tracking and feature encoding stages, which is very interesting. A very high-dimensional descriptor called VLAD was included in their best result, where more features were also used. However, it is still around 3% lower than ours.

On Olympic Sports, we attain better results than most of the compared approaches, including an attribute-based action learning method [19], a graph-based action modeling approach [61], a new sparse coding-based representation [62], a method modeling the dynamics of action attributes over time [64], the approach of Jain et al. [40], and a mid-level representation called motion atom and phrase [65]. Compared with [38], we achieve better result than the without HD approach and similar performance to the HD based approach. Our best performance on HMDB51 is much higher than the baseline result reported [47], where a biologically inspired system of Serre et al. [68] was used. Our approach is also much better than recent approaches like the Action Bank [20], the Motion Interchange Pattern [33], and the new sparse coding-based representation [62]. Compared with a recent work on the sampling of local features [18], a new VLAD encoding approach [60], and the approach of Jain et al. [40], we also achieve better performance. For the approach of Narayan and Ramakrishnan [66], the authors used the Granger causality to model the temporal cause and effect relationships of dense trajectories for action recognition. The result of 58.7% reported in the table is from the fusion of their causality descriptor and the improved dense trajectory baseline. Compared with [38], like the observations on the Olympic Sports, we obtain better performance than the without HD approach and similar result to that of the HD based approach. In addition, a recent work by Peng et al. [67] used a new Fisher vector encoding method and achieved very strong results. As their method focuses on a very different aspect of the problem, our method is expected to be complementary.

Since the UCF101 is relatively a new benchmark, there are not many published results. The original baseline [55] is based on the simple and standard HOG/HOF descriptors, which is much worse than our approach. Compared with recent works on fusing multiple super vectors [58] and improved VLAD encoding [60], we also achieve better result with clear margins. Our result is also better than the without HD performance of Wang and Schmid [38], which was reported in the THUMOS action recognition challenge as the best result [57]. This again verifies the effectiveness of our approach by explicitly modeling the motion relationships, even when the global motion calibration was already used in the improved dense trajectory baseline [38]. Notice that the baseline result of [55] was produced by a multi-class SVM, which we found is generally around 10% lower than using multiple one-vs-all SVMs. All the other results reported in the table are based on the latter.

D. Evaluation of Parameters

In this subsection, we evaluate a few important parameters including the number of clusters in TrajShape′, and the size of the visual codebook, the number of quantization bins

TABLE VIII

COMPARISON WITH THE STATE-OF-THE-ART METHODS. OUR RESULTS ARE GIVEN IN THE BOTTOM ROW. THE PERFORMANCE OF LAPTEV et al. ON THE OLYMPIC SPORTS DATASET IS OBTAINED FROM [54], AND THE PERFORMANCE OF WANG AND SCHMID [38] ON THE UCF101 IS REPORTED IN THE THUMOS ACTION RECOGNITION CHALLENGE 2013 [57]

Hollywood2		Olympic Sports		HMDB51		UCF101	
Taylor et al. [14]	46.6%	Laptev et al. [2]	62.0%	Kuehne et al. [47]	22.8%	Soomro et al. [55]	44.5%
Gilbert et al. [50]	50.9%	Niebles et al. [54]	72.1%	Sadanand et al. [20]	26.9%	Cai et al. [58]	83.5%
Ullah et al. [59]	53.2%	Liu et al. [19]	74.4%	Kliper-Gross et al. [33]	29.2%	Wu et al. [60]	84.2%
Le et al. [15]	53.3%	Brendel et al. [61]	77.3%	Gopalan [62]	34.1%	Wang et al.	
Kantorov et al. [63]	56.7%	Li et al. [64]	78.2%	Wang et al. [4]	46.6%	(w/o HD) [38]	85.9%
Wang et al. [4]	58.2%	Gopalan [62]	78.6%	Shi et al. [18]	47.6%		
Jain et al. [40]	62.5%	Jain et al. [40]	83.2%	Jain et al. [40]	52.1%		
Wang et al. (w/o HD) [38]	63.0%	L. Wang et al. [65]	84.9%	Wu et al. [60]	56.4%		
Wang et al. (w/ HD) [38]	64.3%	Wang et al. (w/o HD) [38]	90.2%	Narayan et al. [66]	58.7%		
		Wang et al. (w/ HD) [38]	91.1%	Wang et al. (w/o HD) [38]	55.9%		
				Wang et al. (w/ HD) [38]	57.2%		
				Peng et al. [67]	66.8%		
65.4%		**91.0%**		**57.3%**		**87.2%**	

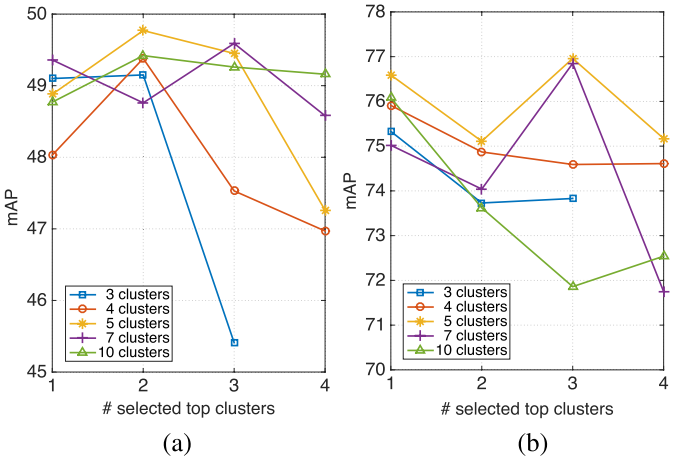


Fig. 7. Performance of TrajShape′ on Hollywood2 (a) and Olympic Sports (b) using different *total* numbers of clusters and different numbers of *selected* clusters for motion compensation.

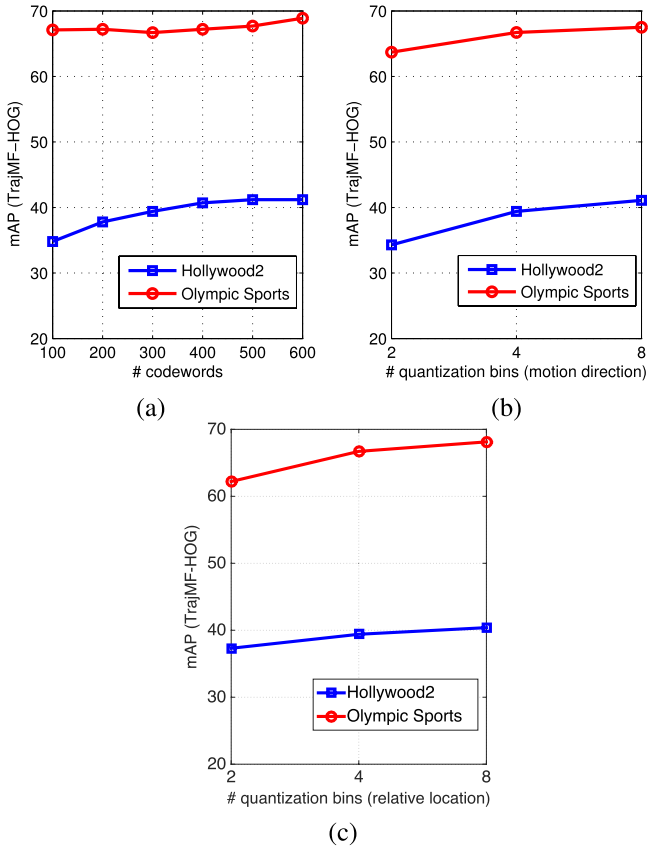


Fig. 8. Evaluation of TrajMF parameters on Hollywood2 and Olympic Sports datasets, using only the TrajMF-HOG feature. (a) Codebook size. (b) Number of motion direction quantization bins. (c) Number of relative location quantization bins.

(for both motion direction and relative location) and the number of dimensions used in the dimension reduced TrajMF. Results of the TrajMF representations are based on the original dense trajectories, which are overall a bit lower than that of the improved trajectories. For most experiments, we report performance on both Hollywood2 and Olympic Sports datasets. For the number of dimensions of TrajMF, we use Hollywood2 and UCF101, as Olympic Sports has too few videos to evaluate a wide range of feature dimensions.

1) Number of Clusters: We first evaluate the performance of TrajShape′ on Hollywood2 and Olympic Sports datasets, using different numbers of clusters and different numbers of selected clusters for motion compensation. Results are shown in Figure 7, where we see that it is consistently good to group all the trajectories into five clusters and then use the top-three largest clusters as references to adjust the trajectories. Using more clusters may bring noise into the representation as the "small" clusters are not always meaningful, and thus the results of selecting four clusters are generally worse than that of three.

2) Number of Codewords: Figure 8(a) shows the results w.r.t. visual codebook size. We use 4 quantization bins for both motion direction and relative location. We see that the performance on both datasets is fairly stable over various codebook sizes. Using a codebook of 600 codewords, we obtain 41.2% on Hollywood2 and 68.9% on Olympic Sports. Since the dimension of TrajMF is quadratic to the number of codewords, the minor gain over smaller codebooks does not justify the use of a much higher dimensional representation. Even using dimension reduction, if the original dimension is

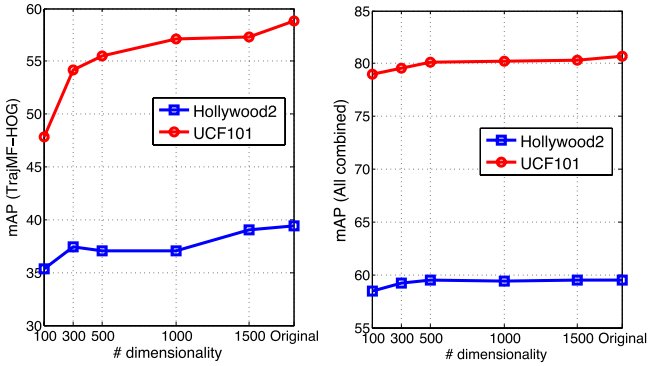


Fig. 9. Evaluation of TrajMF dimensionality on Hollywood2 and UCF101 datasets. (a) TrajMF-HOG feature only. (b) Combination of multiple features.

too high, the reduction process requires more computational workload. Therefore we conclude that a codebook of 200-300 codewords is preferred for TrajMF.

3) Number of Quantization Bins: Figure 8(b) and 8(c) plot the results w.r.t. the number of quantization bins, respectively for motion direction and relative location. We use 300 codewords and fix the number of relative location quantization bins at 4 for (b) and motion direction quantization bins at 2 for (c). 4 bins are consistently better than 2 bins on both datasets. Further using more bins can improve the results slightly.

4) Number of Dimensions: In Figure 9 we further show the results of different dimensionality ranging from 100 to 1500, on the Hollywood2 and UCF101 datasets. We show results of both individual feature (TrajMF-HOG) and the combination of multiple features. We see that the performance of the single feature drops with less dimensions. However, for the fusion result, there is no performance degradation at all when the reduced dimension is as low as 500. These results confirmed that dimension reduction can be reliably used on TrajMF with no performance drop.

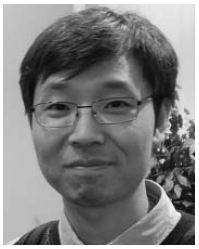
VI. Conclusion

We have introduced an approach for human action recognition in unconstrained videos, where extensive camera motion exists, which affects the performance of many existing features. Our proposed solution explicitly models motion information in videos. Two kinds of motion reference points are considered to alleviate the effect of camera movement and also take object relationships into account in action representation. The object relationships are encoded by the relative motion patterns among pairwise trajectory codewords, so that accurate object boundary detection or foreground-background separation is avoided. Extensive experiments on four challenging action recognition benchmarks (Hollywood2, Olympic Sports, HMDB51 and UCF101) have shown that the proposed approach offers very competitive results. This single approach already outperforms several state-of-the-art methods. We also observed that it is very complementary to the standard bag-of-features and Fisher vectors. In addition, we have shown that the dimension of our proposed TrajMF can be reduced by simple EM-PCA with no performance degradation. Overall, we believe that approaches explicitly modeling motion information are needed in a robust human action recognition system, particularly when dealing with unconstrained videos such as those on the Internet. One promising future work is to further explore higher order relationships instead of just pairwise motion patterns, which may be very helpful for recognizing highly complex actions.

References

[1] I. Laptev, "On space-time interest points," *Int. J. Comput. Vis.*, vol. 64, pp. 107–123, Sep. 2005.
[2] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *Proc. IEEE Conf. CVPR*, Jun. 2008, pp. 1–8.
[3] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Action recognition by dense trajectories," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3169–3176.
[4] H. Wang, A. Klaser, C. Schmid, and C.-L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 60–79, 2013.
[5] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos 'in the wild,"' in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 1996–2003.
[6] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher kernel for large-scale image classification," in *Proc. 11th ECCV*, 2010, pp. 143–156.
[7] H. Jegou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 3304–3311.
[8] X. Zhou, K. Yu, T. Zhang, and T. S. Huang, "Image classification using super-vector coding of local image descriptors," in *Proc. 11th ECCV*, 2010, pp. 141–154.
[9] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *Proc. 2nd Joint IEEE Int. Workshop VS-PETS*, Oct. 2005, pp. 65–72.
[10] I. N. Junejo, E. Dexter, I. Laptev, and P. Pérez, "Cross-view action recognition from temporal self-similarities," in *Proc. 10th ECCV*, 2008, pp. 293–306.
[11] P. Scovanner, S. Ali, and M. Shah, "A 3-dimensional SIFT descriptor and its application to action recognition," in *Proc. 15th Int. Conf. MM*, 2007, pp. 357–360.
[12] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *Proc. 10th ECCV*, 2008, pp. 650–663.
[13] J. Knopp, M. Prasad, G. Willems, R. Timofte, and L. Van Gool, "Hough transform and 3D SURF for robust three dimensional classification," in *Proc. 11th ECCV*, 2010, pp. 589–602.
[14] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler, "Convolutional learning of spatio-temporal features," in *Proc. 11th ECCV*, 2010, pp. 140–153.
[15] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng, "Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3361–3368.
[16] A. Kovashka and K. Grauman, "Learning a hierarchy of discriminative space-time neighborhood features for human action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2010, pp. 2046–2053.
[17] E. Vig, M. Dorr, and D. Cox, "Space-variant descriptor sampling for action recognition based on saliency and eye movements," in *Proc. 12th ECCV*, 2012, pp. 84–97.
[18] F. Shi, E. Petriu, and R. Laganiere, "Sampling strategies for real-time action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2595–2602.
[19] J. Liu, B. Kuipers, and S. Savarese, "Recognizing human actions by attributes," in *Proc. IEEE Conf. CVPR*, Jun. 2011, pp. 3337–3344.
[20] S. Sadanand and J. J. Corso, "Action bank: A high-level representation of activity in video," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1234–1241.
[21] H. Uemura, S. Ishikawa, and K. Mikolajczyk, "Feature tracking and motion compensation for action recognition," in *Proc. BMVC*, 2008, pp. 1–10.

[22] R. Messing, C. Pal, and H. Kautz, "Activity recognition using the velocity histories of tracked keypoints," in *Proc. IEEE 12th ICCV*, Sep./Oct. 2009, pp. 104–111.
[23] F. Wang, Y.-G. Jiang, and C.-W. Ngo, "Video event detection using motion relativity and visual relatedness," in *Proc. 16th Int. Conf. MM*, 2008, pp. 239–248.
[24] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," in *Proc. 11th ECCV*, 2010, pp. 577–590.
[25] P. Matikainen, M. Hebert, and R. Sukthankar, "Representing pairwise spatial and temporal relations for action recognition," in *Proc. 11th ECCV*, 2010, pp. 508–521.
[26] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, and J. Li, "Hierarchical spatio-temporal context modeling for action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2004–2011.
[27] P. Turaga, R. Chellappa, V. S. Subrahmanian, and O. Udrea, "Machine recognition of human activities: A survey," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1473–1488, Nov. 2008.
[28] R. Poppe, "A survey on vision-based human action recognition," *Image Vis. Comput.*, vol. 28, no. 6, pp. 976–990, 2010.
[29] Y.-G. Jiang, S. Bhattacharya, S.-F. Chang, and M. Shah, "High-level event recognition in unconstrained videos," *Int. J. Multimedia Inf. Retr.*, vol. 2, no. 2, pp. 73–101, 2013.
[30] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th IJCAI*, 1981, pp. 674–679.
[31] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *Proc. BMVC*, 2012, pp. 30.1–30.13.
[32] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering discriminative action parts from mid-level video representations," in *Proc. IEEE Conf. CVPR*, Jun. 2012, pp. 1242–1249.
[33] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf, "Motion interchange patterns for action recognition in unconstrained videos," in *Proc. 12th ECCV*, 2012, pp. 256–269.
[34] X. Wang, K. T. Ma, G.-W. Ng, and W. E. L. Grimson, "Trajectory analysis and semantic region modeling using nonparametric hierarchical Bayesian models," *Int. J. Comput. Vis.*, vol. 95, no. 3, pp. 287–312, 2011.
[35] S. Wu, O. Oreifej, and M. Shah, "Action recognition in videos acquired by a moving camera using motion decomposition of Lagrangian particle trajectories," in *Proc. ICCV*, 2011, pp. 1419–1426.
[36] F. I. Bashir, A. A. Khokhar, and D. Schonfeld, "Object trajectory-based activity classification and recognition using hidden Markov models," *IEEE Trans. Image Process.*, vol. 16, no. 7, pp. 1912–1919, Jul. 2007.
[37] A. Hervieu, P. Bouthemy, and J.-P. Le Cadre, "A statistical video content recognition method using invariant features on object trajectories," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1533–1543, Nov. 2008.
[38] H. Wang and C. Schmid, "Action recognition with improved trajectories," in *Proc. IEEE ICCV*, Dec. 2013, pp. 3551–3558.
[39] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," in *Proc. 9th ECCV*, 2006, pp. 428–441.
[40] M. Jain, H. Jegou, and P. Bouthemy, "Better exploiting motion for better action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2555–2562.
[41] C. R. Jung, L. Hennemann, and S. R. Musse, "Event detection using trajectory clustering and 4D histograms," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 11, pp. 1565–1575, Nov. 2008.
[42] G. Piriou, P. Bouthemy, and J.-F. Yao, "Recognition of dynamic video contents with global probabilistic models of visual motion," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3417–3430, Nov. 2006.
[43] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo, "Trajectory-based modeling of human actions with motion reference points," in *Proc. 12th ECCV*, 2012, pp. 425–438.
[44] G. Farnebäck, "Two-frame motion estimation based on polynomial expansion," in *Proc. 13th Scandin. Conf. Image Anal.*, 2003, pp. 363–370.
[45] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Conf. CVPR*, Jun. 2005, pp. 886–893.
[46] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *Proc. BMVC*, 2008, pp. 124.1–124.11.
[47] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," in *Proc. IEEE ICCV*, Nov. 2011, pp. 2556–2563.
[48] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE ICCV*, Oct. 2003, pp. 1470–1477.
[49] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Conf. CVPR*, Jun. 2007, pp. 1–8.
[50] A. Gilbert, J. Illingworth, and R. Bowden, "Action recognition using mined hierarchical compound features," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 5, pp. 883–897, May 2011.
[51] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.
[52] S. Roweis, "EM algorithms for PCA and SPCA," in *Proc. NIPS*, 1998, pp. 626–632.
[53] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *Proc. IEEE Conf. CVPR*, Jun. 2009, pp. 2929–2936.
[54] J. C. Niebles, C.-W. Chen, and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *Proc. 11th ECCV*, 2010, pp. 392–405.
[55] K. Soomro, A. R. Zamir, and M. Shah. (2012). "UCF101: A dataset of 101 human actions classes from videos in the wild." [Online]. Available: http://arxiv.org/abs/1212.0402
[56] M. Brown and D. G. Lowe, "Automatic panoramic image stitching using invariant features," *Int. J. Comput. Vis.*, vol. 74, no. 1, pp. 59–73, 2007.
[57] Y.-G. Jiang *et al.* (2013). *THUMOS Challenge: Action Recognition With a Large Number of Classes*. [Online]. Available: http://crcv.ucf.edu/ICCV13-Action-Workshop/
[58] Z. Cai, L. Wang, X. Peng, and Y. Qiao, "Multi-view super vector for action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 596–603.
[59] M. M. Ullah, S. N. Parizi, and I. Laptev, "Improving bag-of-features action recognition with non-local cues," in *Proc. BMVC*, 2010, pp. 95.1–95.11.
[60] J. Wu, Y. Zhang, and W. Lin, "Towards good practices for action video encoding," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2577–2584.
[61] W. Brendel and S. Todorovic, "Learning spatiotemporal graphs of human activities," in *Proc. IEEE ICCV*, Nov. 2011, pp. 778–785.
[62] R. Gopalan, "Joint sparsity-based representation and analysis of unconstrained activities," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2738–2745.
[63] V. Kantorov and I. Laptev, "Efficient feature extraction, encoding, and classification for action recognition," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2593–2600.
[64] W. Li, Q. Yu, H. Sawhney, and N. Vasconcelos, "Recognizing activities via bag of words for attribute dynamics," in *Proc. IEEE Conf. CVPR*, Jun. 2013, pp. 2587–2594.
[65] L. Wang, Y. Qiao, and X. Tang, "Mining motion atoms and phrases for complex action recognition," in *Proc. IEEE ICCV*, Dec. 2013, pp. 2680–2687.
[66] S. Narayan and K. R. Ramakrishnan, "A cause and effect analysis of motion trajectories for modeling actions," in *Proc. IEEE Conf. CVPR*, Jun. 2014, pp. 2633–2640.
[67] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action recognition with stacked fisher vectors," in *Proc. 13th ECCV*, 2014, pp. 581–595.
[68] T. Serre, L. Wolf, S. Bileschi, M. Riesenhuber, and T. Poggio, "Robust object recognition with cortex-like mechanisms," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 3, pp. 411–426, Mar. 2007.



Yu-Gang Jiang received the Ph.D. degree in computer science from the City University of Hong Kong, Hong Kong, in 2009. From 2008 to 2011, he was with the Department of Electrical Engineering, Columbia University, New York, NY. He is currently an Associate Professor of Computer Science with Fudan University, Shanghai, China. His research interests include multimedia retrieval and computer vision. He is one of the organizers of the annual THUMOS Challenge on Large Scale Action Recognition, and is currently serving as a Program Chair of ACM ICMR 2015. He was a recipient of many awards, including the prestigious ACM China Rising Star Award in 2014.

Qi Dai received the B.Sc. degree in computer science from the East China University of Science and Technology, Shanghai, China, in 2011. He is currently pursuing the Ph.D. degree with the School of Computer Science, Fudan University, Shanghai. His research interests include multimedia retrieval and computer vision.

Wei Liu received the M.Phil. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, USA, in 2012. He is currently a Research Staff Member with the IBM T. J. Watson Research Center, Yorktown Heights, NY, USA, and holds an adjunct faculty position with the Rensselaer Polytechnic Institute, Troy, NY, USA. He has been the Josef Raviv Memorial Post-Doctoral Fellow with the IBM T. J. Watson Research Center for one year since 2012. His research interests include machine learning, data mining, computer vision, pattern recognition, and information retrieval. He was a recipient of the 2011-2012 Facebook Fellowship and the 2013 Jury Award for best thesis of the Department of Electrical Engineering, Columbia University.

Xiangyang Xue received the B.S., M.S., and Ph.D. degrees in communication engineering from Xidian University, Xi'an, China, in 1989, 1992, and 1995, respectively. He joined the Department of Computer Science, Fudan University, Shanghai, China, in 1995, where he has been a Full Professor since 2000. He has authored over 100 research papers in these fields. His current research interests include multimedia information processing and retrieval, pattern recognition, and machine learning. He is currently an Associate Editor of the IEEE TRANSACTIONS ON AUTONOMOUS MENTAL DEVELOPMENT. He is also an Editorial Board Member of the *Journal of Computer Research and Development*, and the *Journal of Frontiers of Computer Science and Technology*.

Chong-Wah Ngo received the M.Sc. and B.Sc. degrees in computer engineering from Nanyang Technological University, Singapore, and the Ph.D. degree in computer science from The Hong Kong University of Science and Technology. He was a Post-Doctoral Scholar with the Beckman Institute, University of Illinois in Urbana-Champaign, and a Visiting Researcher with Microsoft Research Asia. He is currently a Professor with the Department of Computer Science, City University of Hong Kong. His recent research interests include video computing and multimedia mining. He is currently an Associate Editor of the IEEE TRANSACTIONS ON MULTIMEDIA.