Trajectory-Based Modeling of Human Actions with Motion Reference Points

Yu-Gang Jiang[1], Qi Dai[1], Xiangyang Xue[1], Wei Liu[2], and Chong-Wah Ngo[3]

[1] School of Computer Science, Fudan University, Shanghai, China
[2] IBM T. J. Watson Research Center, NY, USA
[3] Department of Computer Science, City University of Hong Kong, China

Abstract. Human action recognition in videos is a challenging problem with wide applications. State-of-the-art approaches often adopt the popular bag-of-features representation based on isolated local patches or temporal patch trajectories, where motion patterns like object relationships are mostly discarded. This paper proposes a simple representation specifically aimed at the modeling of such motion relationships. We adopt global and local reference points to characterize motion information, so that the final representation can be robust to camera movement. Our approach operates on top of visual codewords derived from local patch trajectories, and therefore does not require accurate foreground-background separation, which is typically a necessary step to model object relationships. Through an extensive experimental evaluation, we show that the proposed representation offers very competitive performance on challenging benchmark datasets, and combining it with the bag-of-features representation leads to substantial improvement. On Hollywood2, Olympic Sports, and HMDB51 datasets, we obtain 59.5%, 80.6% and 40.7% respectively, which are the best reported results to date.

1 Introduction

The recognition of human actions in videos is a topic of active research in computer vision. Significant progress has been made in recent years, particularly with the invention of local invariant features and the bag-of-features framework. For example, currently a common solution that shows state-of-the-art 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., [3]).

However, the typical bag-of-features approach does not capture the motion relationships among objects and the background scene. We argue that such motion patterns are important and thus should be incorporated into a recognition system, especially when the target videos are captured under unconstrained environment with severe camera motion. This paper proposes an approach to model the motion relationships among moving objects and the background. In particular, we introduce two kinds of reference points to characterize complex motions in the unconstrained videos, in order to alleviate the effect incurred by camera

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part V, LNCS 7576, pp. 425–438, 2012.

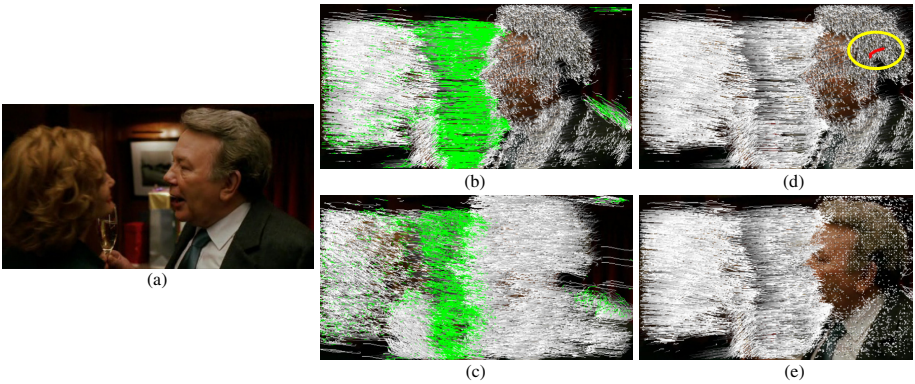

Fig. 1. (a) A video frame of a *kissing* action. (b) Local patch trajectories, with the largest trajectory cluster shown in green. (c) Amended trajectories by using the mean motion of the green cluster as a global reference point; See details in Section 4.1. (d) The original patch trajectories, with 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 4.2. This figure is best viewed in color.

movement. Figure 1 illustrates our proposed approach. Tracking of local frame patches is firstly performed to capture the motion of the local patches. With the trajectories, we use a simple clustering method to identify the dominant motion of the scene, which is used as a *global* motion reference point to calibrate the motion of each trajectory. In addition, to capture the relationships of moving objects, we treat each trajectory as a *local* motion reference point for motion characterization, which leads to a rich representation that encapsulates 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. Although simple in its form, our approach holds the following advantages.

First, it has been realized that motion patterns, particularly the interaction of moving objects, are critical for recognizing human actions (e.g., the proximity changes between two people in action "kissing"), and the modeling of such motion interactions in unconstrained videos is not easy due to camera motion. Two intuitive ways to cancel the camera motion are to perform foreground-background separation or video stabilization, which are still difficult research problems, however. Therefore 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. Moreover, we show that the proposed motion representation works well with efficient classifiers, producing very competitive action recognition accuracy on several challenging benchmarks.

The rest of this paper is structured as follows. Section 2 discusses related works. Section 3 briefly introduces the tracking of local patches, which is the basis of our representation. Section 4 elaborates the proposed approach and Section 5 presents extensive experimental validations. Finally, we conclude in Section 6.

2 Related Works

Local features, coupled with the bag-of-features framework, are the most popular way to represent images [4,5] and videos [2,6]. Most of the recent works on video representation belong to two categories. The first category extracts/learns spatial-temporal local features where many efforts have been devoted to the design of good detectors/descriptors [1,7,8,9,10,11] or feature learning algorithms [12,13,14]. 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 patch trajectories [15,16,17,18,19,20,3]. In this section we focus our discussion on the trajectory-based approaches, which are more related to this work. Readers are referred to [21,22] for comprehensive surveys of action recognition techniques.

In [15], Uemura et al. [15] extracted trajectories of SIFT patches with the KLT tracker [23]. Mean-Shift based frame segmentation was used to estimate dominate plane in the scene, which was used for motion compensation. Messing et al. [16] computed velocity histories of the KLT-based trajectories for action recognition. The work of [20] also adopted the KLT tracker, and proposed representations to model inter-trajectory proximity. Wang et al. [17] modeled the motion between KLT-based keypoint trajectories, without considering trajectory locations. Spatial and temporal context of trajectories was explored in [19] with an elegant probabilistic formulation. In addition, Raptis and Soatto [18] proposed tracklet, which emphasizes more on the local casual structures of action elements (short trajectories), not the pairwise motion patterns. A recent work by Wang et al. [3] 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) on several human action recognition benchmarks. To cope with camera motion, they extended Dalal's motion boundary histogram (MBH) [24] as an 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.

This paper presents a new video representation that 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. By simply using global and local reference points to suppress the noise caused by camera motion, we avoid the use of expensive and unreliable foreground-background separation or video stabilization algorithms.

3 Dense Trajectories

Our proposed representation is grounded on local patch trajectories. In this work, we adopt the dense trajectory approach by Wang et al. [3], which is briefly introduced as follows. Note that our approach can be applied on top of any local patch trajectories.

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

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

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 [25] is adopted, which is available from the OpenCV library. A maximum value of trajectory length is set to avoid a drifting problem that often occurs when trajectories are long, and 15 frames were found to be a suitable choice. Also, trajectories with sudden large displacements are removed from the final set.

Several descriptors can be computed to encode either the shape of a trajectory or the local motion and appearance within a space-time volume around the trajectory. In [3], the shape of a trajectory is described 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 changes, the concatenated vector is normalized by the overall magnitude of motion displacements:

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

where $L = 15$ is the length of the trajectories.

The local motion and appearance around a trajectory are described by Histograms of Oriented Gradients (HoG) [26], Histograms of Optical Flow (HOF), and the MBH. HOG encodes local appearance information, while HOF and MBH capture local motion pattern. 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. HOF has 9 bins in total, with one more zero bin compared to HOG. Therefore the final representation has 96 dimensions for HOG and 108 dimensions for HOF. MBH computes a histogram based on the derivatives of optical flow separately on both horizontal and vertical components. Like HOG, 8 bins are used to quantize orientations, and since there are two motion boundary maps based on derivatives along two directions, the MBH descriptors are of $96 \times 2 = 192$ dimensions. By using derivatives of optical flow, MBH is able to cancel global motion and only captures local relative motion of pixels. This is quite appealing for the analysis of realistic videos with severe camera motion, but the pairwise motion relationships are not captured in MBH. The parameter choices for computing these descriptors are based on an empirical study conducted in [3]. All the three descriptors have been shown to be effective for action recognition in unconstrained videos [2,27,6,3].

4 Trajectory-Based Motion Modeling

This section elaborates our trajectory-based motion modeling approach for human action recognition. We first describe a method that utilizes global reference points to cancel camera motion specifically for improving the TrajShape descriptor. After that we introduce a trajectory-based motion representation which uses each trajectory as a local reference point. This representation incorporates location and motion relationships of the patch trajectories as well as their local descriptors, and is not sensitive to camera motion. Between the two, the latter representation is considered as a more important contribution. We discuss the details of our approach in the following.

4.1 Trajectory Shape Descriptor with Global Reference Points

Uncovering the global motion pattern in complex unconstrained videos is not an easy process. Typical solutions like foreground-background separation [15] and video stabilization [28] are very expensive to compute. We therefore pursue a simple solution by clustering the motion patterns of trajectories on the scene. The dominant pattern is treated as a reference point to characterize motion, so that the effect of global motion can be alleviated.

Given a trajectory $\mathcal{T}$ with start position P_t on frame t, the overall motion displacement of the trajectory is $\Delta\mathcal{T} = (P_{t+L-1} - P_t) = (x_{t+L-1} - x_t, y_{t+L-1} - y_t)$. Since the length of dense trajectories has been limited to only 15 frames (0.5 seconds for a video of 30 fps), we do not further split it and only use the overall displacement to represent the motion of the trajectory. 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[1], we cluster trajectories starting within each 5-frame window of a video, and empirically generate five trajectory clusters per

[1] Note that the TrajShape descriptor also can be used to generate the trajectory clusters, but we have observed that the two dimensional displacement vectors show similar results at a much faster speed.

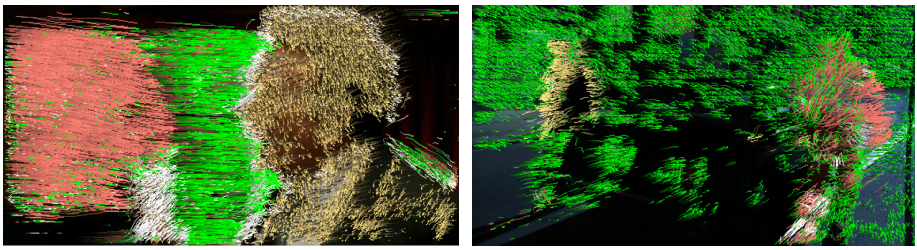

Fig. 2. Trajectory clustering results. Trajectories in the top-three largest clusters are visualized in green, light red and yellow, while the remaining ones are in white. **Left.** two people kissing; **Right.** two people getting out of a car. This figure is best viewed in color.

window. Since it is difficult to predict which cluster contains trajectories on the background and which one refers to a moving object, we use the top-three largest clusters and compute the mean motion displacement of each cluster as a candidate dominant motion direction. We have found that this is more reliable than using a single cluster. Figure 2 visualizes the trajectory clustering results on two example frames, where the top-three clusters are visualized in different colors.

Denote the mean motion displacement of a trajectory cluster $\mathcal{C}$ as $\Delta\mathcal{C} = (\Delta\bar{x}_c, \Delta\bar{y}_c)$. The displacement of a trajectory between two nearby frames within the corresponding 5-frame window is adjusted to $\Delta P'_t = \Delta P_t - \Delta\mathcal{C}/15$. We then proceed to update the displacement of all the trajectories in the next 5-frame window until the end of the video. With this compensation by the dominant motion, the TrajShape descriptor can be updated following Equation (2): TrajShape$' = (\Delta P'_t, \ldots, \Delta P'_{t+L-1}) / \sum_{i=t}^{t+L-1} \|\Delta P'_i\|$, where TrajShape$'$ is the amended descriptor. Using the mean motion displacements of the three largest clusters, a trajectory is now associated with a set of three TrajShape$'$ descriptors, each adjusted by the motion pattern of one cluster. The method of converting sets of TrajShape$'$ for video representation will be introduced later in Section 4.3.

4.2 Motion Representation with Local Reference Points

We now introduce the pairwise motion representation. Since the number of trajectories varies across different videos, a common way to generate fixed-dimensional video representations is to use *visual codewords*, which are cluster centers of the trajectory descriptors. This is in the same spirit to the classical bag-of-features framework based on static SIFT descriptors [4]. In our representation, we also adopt visual codewords as the basic units to encode the pairwise motion relationships. For each type of trajectory descriptor (e.g., HOF), a codebook of n codewords is constructed by clustering a subset of the descriptors using k-means.

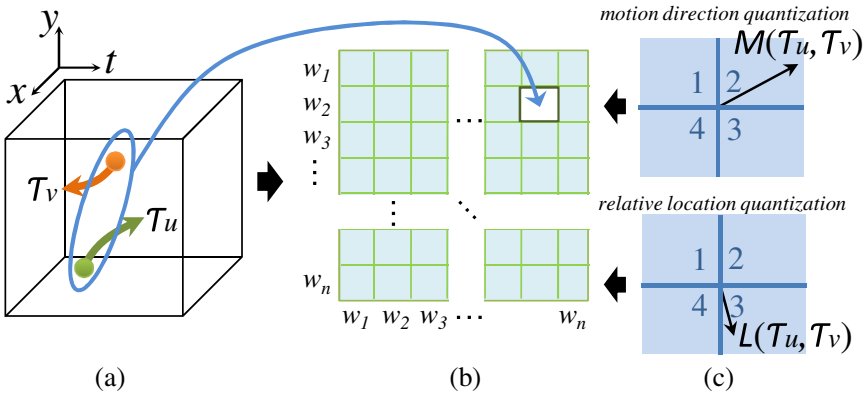

Fig. 3. An illustration of our trajectory-based motion feature representation—TrajMF. Each trajectory pair in (a) is mapped to an entry of a codeword-based representation (b), according to the local descriptors of the two trajectories. The motion between each codeword pair, i.e., an entry in (b), is further described by a 16-d vector, based on the relative motion direction and relative location of the trajectory pairs falling into that entry. The quantization maps for generating the 16-d vector are shown in (c). See texts for more explanations.

Given two trajectories $\mathcal{T}_u$ and $\mathcal{T}_v$, their relative motion (with $\mathcal{T}_v$ as the local reference point) can be computed by

$$\mathcal{M}(\mathcal{T}_u, \mathcal{T}_v) = \Delta\mathcal{T}_u - \Delta\mathcal{T}_v. \tag{3}$$

Note that in this representation there is no need to use the dominant motion $\Delta\mathcal{C}$ to suppress global motion, since the pairwise relative motion is naturally robust to camera movement. Later in the experiments we will show that the improved trajectory shape descriptor TrajShape′ can be used in combination with this pairwise motion representation to achieve better action recognition performance.

Figure 3 illustrates the generation of the trajectory-based motion feature representation, dubbed TrajMF. The motion $\mathcal{M}(\mathcal{T}_u, \mathcal{T}_v)$ between two trajectories is quantized in a way that integrates very rich information, including trajectory neighborhood descriptors, motion direction and magnitude, as well as the relative location of the two trajectories. The local information is encoded in TrajMF by using the trajectory descriptor codewords. Specifically, we only consider the overall motion between two codewords in the final representation, and all the pairwise trajectory motion patterns are mapped to the corresponding codeword pairs. Because a visual codeword may generally represent a (moving) local pattern of an object or a background scene, the TrajMF representation implicitly captures object-object or object-background relationships.

The motion pattern between two codewords is quantized into a vector, according to both the relative motion direction and the relative location of each

trajectory pair that belongs to the codeword pair. Formally, let $Q(\cdot)$ be the quantization function based on motion direction and relative location (see the quantization maps in Figure 3(c)), which returns 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

$$\mathbf{f}(w_p, w_q) = \sum_{\forall(\mathcal{T}_u,\mathcal{T}_v)\rightarrow(w_p,w_q)} Q(\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v), \mathcal{L}(\mathcal{T}_u,\mathcal{T}_v)) \cdot ||\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v)||, \quad (4)$$

where $\mathcal{L}(\mathcal{T}_u,\mathcal{T}_v) = (\bar{P}_{\mathcal{T}_u} - \bar{P}_{\mathcal{T}_v}) = (\bar{x}_{\mathcal{T}_u} - \bar{x}_{\mathcal{T}_v}, \bar{y}_{\mathcal{T}_u} - \bar{y}_{\mathcal{T}_v})$ indicates the relative location of the mean positions of two trajectories, "$\rightarrow$" denotes the trajectory-to-codeword mapping, and $||\mathcal{M}(\mathcal{T}_u,\mathcal{T}_v)||$ is the magnitude of the relative motion. In our experiments we use four bins to quantize both the motion direction and the relative location direction, and therefore $\mathbf{f}$ is 16-d. We have evaluated several choices for the number of areas used in quantization and found 4 is suitable choice (cf. Section 5). Concatenating $\mathbf{f}$ of all the codeword pairs, our final TrajMF representation has $\frac{n\times n}{2} \times 4 \times 4$ dimensions (n is the number of codewords).

4.3 Classification

This subsection briefly discusses the classifier choices for the augmented trajectory shape descriptor and the TrajMF representation. For TrajShape$'$, we use the standard bag-of-features approach to convert a set of descriptors from a video into a fixed-dimensional vector. Following [2,3], we construct a visual codebook of 4,000 codewords using k-means. We quantize all the three TrajShape$'$ descriptors of each trajectory together into a single 4,000-d histogram for each video, which is used as the final representation. This feature is classified by the popular χ^2 kernel SVM.

The TrajMF can be applied on top of any basic trajectory descriptors. In this work we adopt all the three well-performing descriptors used in [3]: HOG, HOF, and MBH. For each trajectory descriptor, a separate TrajMF representation is generated. Like many previous works that modeled pairwise feature relationships, the dimension of the TrajMF is high, making non-linear classifiers like the χ^2 SVM unsuitable due to speed limitation. Existing methods of pairwise feature modeling often adopt data mining techniques for feature selection [29,30]. We have experimented these techniques but found them ineffective—reducing the dimension always results in degraded performance. We therefore seek another solution by using fast classification techniques. The most simple option is linear SVM, which is extremely fast, even for such high-dimensional representations. We will also test Maji's fast Histogram Intersection (HI) kernel SVM [31], with which classification can be executed in logarithmic complexity to the number of support vectors.

5 Experiments

Considering the simplicity of our approach, one question that naturally arises is how well it performs on popular benchmark datasets. We conduct extensive experiments using three challenging datasets of realistic videos: Hollywood2

Fig. 4. Video frames of example action classes in Hollywood2 (top), Olympic Sports (middle) and HMDB51 (bottom) datasets

dataset [32], Stanford Olympic Sports dataset [33], and HMDB51 dataset [28]. Many videos in these datasets contain camera motion and their contents are very diverse (see Figure 4).

The Hollywood2 dataset [32] contains 1,707 video clips, collected from 69 Hollywood movies. The clips are divided into a training set of 823 samples and a test set of 884 samples. There are 12 action classes in this dataset: 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 modeled by a one-versus-all SVM classifier. Performance is measured by average precision (AP) for each single class and mean AP (mAP) is used to measure the overall performance of all the classes.

The Olympic Sports dataset [33] contains 783 clips and 16 action classes (around 50 clips per class): 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 train/test split from Niebles et al. [33]. AP/mAP is reported as the performance measure.

The HMDB51 dataset was recently collected by Kuehne et al. [28], containing 6,766 video clips in total. There are 51 action classes, each with at least 101 positive samples. We adopt the official setting of [28] to use three train–test splits. Each split has 70 training and 30 testing clips for each class. Following [28], we report mean classification accuracy over the three splits.

5.1 Results and Comparison

We first evaluate the performance of our proposed representations and compare with the state-of-the-art approaches. We set the number of codewords n as 300,

Table 1. Performance of baselines, our representations, and their combination on Hollywood2, Olympic Sports and HMDB51 datasets. The "4 combined" baseline results (using four features TrajShape, HOG, HOF and MBH) are based on the approach of [3], where the features are represented by the standard bag-of-features. The amended TrajShape′ descriptor shows better performance than its original version on all the three datasets. The combination of TrajShape′ and three TrajMF representations ("Our 4 combined") shows better results than the baseline on Olympic Sports and HMDB51. Moreover, the combination of our four representations with the four baseline bag-of-features leads to very competitive results (indicated by "All combined"). To our knowledge, the numbers shown in the bottom row are to-date the best reported performance on all the three datasets.

	Approach	**Hollywood2**	**Olympic Sports**	**HMDB51**
Baseline results	TrajShape	49.3%	59.5%	24.0%
	4 combined [3]	58.4%	74.3%	37.7%
Our results	TrajShape′	50.2%	59.6%	26.7%
	TrajMF-HOG	39.4%	66.7%	24.3%
	TrajMF-HOF	42.3%	56.0%	25.0%
	TrajMF-MBH	46.9%	74.6%	34.0%
	Our 4 combined	55.6 %	77.6%	39.8%
	All combined	**59.5%**	**80.6%**	**40.7%**

and use 4 bins to quantize both the motion direction and the relative location as shown in Figure 3. The linear kernel SVM is adopted for the three TrajMF representations (each based on a different trajectory descriptor) and the χ^2 kernel SVM is used for the others. We will evaluate the number of codewords and quantization bins in TrajMF later.

Table 1 summarizes the results on the three datasets. In addition to evaluating our proposed representations, we also report the results of bag-of-features baselines using the same dense trajectory descriptors. Following [3], in the bag-of-features, we use a codebook of 4000 codewords for each type of trajectory descriptor[2]. As shown in the table, we see that the amended trajectory shape descriptor TrajShape′ outperforms the original TrajShape, which demonstrates the effectiveness of using the simple clustering-based method to cancel global motion. More importantly, the TrajMF representation shows fairly competitive performance—combining our TrajShape′ and TrajMF representations ("Our 4 combined") generates better results than the "4 combined" baseline of [3] on Olympic Sports and HMDB51 datasets. Here the combination is done by simply averaging the kernels computed from different representations. In addition, we also observe that further combining our representations with the baseline ("All combined") gives substantial improvements on all the three datasets. This indicates that the TrajMF representations are quite complementary to the standard bag-of-features.

[2] Source codes for generating dense trajectories and computing the basic descriptors are available online (`http://lear.inrialpes.fr/people/wang/dense_trajectories`). The bag-of-features is based on our own implementation.

Table 2. Performance of different kernels. "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**
Our 4 combined	χ^2	58.1%	77.7%	38.3%
	HI	58.6%	76.9%	37.7%
	Linear	55.6%	77.6%	39.8%
All combined	χ^2	**60.1%**	79.2%	38.8%
	HI	**60.3%**	78.9%	38.4%
	Linear	**59.5%**	**80.6%**	**40.7%**

Table 3. Comparison with the state-of-the-art approaches. Our results are listed in the bottom row. The performance of Laptev et al. on Olympic Sports is from [33].

Hollywood2		**Olympic Sports**		**HMDB51**	
Taylor et al. [12]	46.6%	Laptev et al. [2]	62.0%	Kuehne et al. [28]	
Gilbert et al. [30]	50.9%	Niebles et al. [33]	72.1%	– HOG-HOF	20.4%
Ullah et al. [34]	53.2%	Liu et al. [35]	74.4%	– C2	22.8%
Le et al. [13]	53.3%	Brendel et al. [36]	77.3%		
Wang et al. [3]	58.3%				
59.5%		**80.6%**		**40.7%**	

Next, we compare the results of different classification kernels in Table 2, where χ^2, HI, and linear kernel SVMs are used to classify the TrajMF representations. We only list the results of the combined representations in the table due to space limitation. The linear kernel SVM offers strong performance except for the "Our 4 combined" approach on the Hollywood2 dataset, for which HI and χ^2 are better. For the "All combined" approach, linear kernel contributes to the top or near-top accuracy on all the three datasets, which is quite appealing since it is more efficient.

In Table 3, we compare our results with the state-of-the-art approaches. On Hollywood2, we obtain 1.2% gain over [3] (linear kernel for TrajMF features; 2% gain when using the HI kernel), which used bag-of-features on dense trajectories. This performance gain is nontrivial considering that our result is based on the same set of trajectories and the only added information comes through the use of the two types of motion reference points. Compared with a recent hierarchical spatio-temporal feature learning approach [13], a significant gain of 6.2% is achieved. On Olympic Sports, we attain better results than all the state of the arts, including an attribute-based action learning method [35] and a graph-based action modeling approach [36]. Our best performance on HMDB51 almost doubles the two results reported in [28], where the HOG-HOF approach is based on the work of Laptev et al. [2] and C2 uses a biologically inspired system of Serre et al. [37].

5.2 Evaluation of TrajMF Parameters

In this subsection, we evaluate a few parameters in generating the TrajMF representation, including the size of the visual codebook and the number of

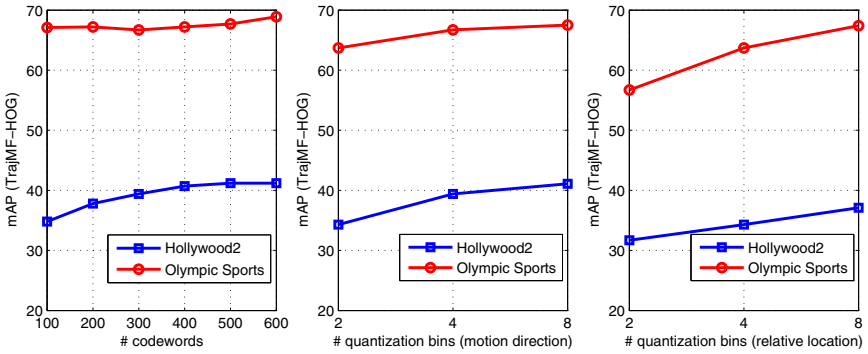

Fig. 5. Evaluation of TrajMF parameters on Hollywood2 and Olympic Sports datasets. (a) Codebook size. (b) Number of motion direction quantization bins. (c) Number of relative location quantization bins.

quantization bins (for both motion direction and relative location). We report performance of the TrajMF-HOG representation on both Hollywood2 and Olympic Sports datasets.

Number of Codewords. Figure 5(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. Therefore we conclude that a codebook of 200-300 codewords is preferred for TrajMF.

Number of Quantization Bins. Figure 5(b) and 5(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 may slightly improve the results, while resulting in representations of much higher dimensions.

6 Conclusion

In this paper, we have introduced an approach for motion-based action modeling, where two kinds of motion reference points are considered to alleviate the effect of camera movement and—more importantly—take object relationships into account in our action representation. The object relationships are encoded by the motion patterns among pairwise trajectory codewords, so that accurate object boundary detection or foreground-background separation is avoided. Extensive experiments on three challenging action recognition benchmarks (Hollywood2, Olympic Sports and HMDB51) 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. By simply combining our representations and the bag-of-features using kernel-level fusion, we attain to-date the best results on all the three benchmarks. For future work, we plan to compress the TrajMF representation and also explore this approach in other computer vision tasks.

Acknowledgments. This work was supported in part by two STCSM's Programs (No. 10511500703 & No. 12XD1400900), a National 863 Program (No. 2011AA010604), a National 973 Program (No. 2010CB327906), and a grant from the Research Grants Council of the Hong Kong S.A.R., China (CityU 119709).

References

1. Laptev, I.: On space-time interest points. IJCV 64, 107–123 (2005)
2. Laptev, I., Marszalek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
3. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
4. Sivic, J., Zisserman, A.: Video Google: A text retrieval approach to object matching in videos. In: ICCV (2003)
5. Lazebnik, S., Schmid, C., Ponce, J.: Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In: CVPR (2006)
6. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos "in the wild". In: CVPR (2009)
7. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: VS-PETS (2005)
8. Junejo, I.N., Dexter, E., Laptev, I., Perez, P.: Cross-View Action Recognition from Temporal Self-similarities. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 293–306. Springer, Heidelberg (2008)
9. Scovanner, P., Ali, S., Shah, M.: A 3-dimensional SIFT descriptor and its application to action recognition. In: ACM MM (2007)
10. Willems, G., Tuytelaars, T., van Gool, L.: An Efficient Dense and Scale-Invariant Spatio-Temporal Interest Point Detector. In: Forsyth, D., Torr, P., Zisserman, A. (eds.) ECCV 2008, Part II. LNCS, vol. 5303, pp. 650–663. Springer, Heidelberg (2008)
11. Knopp, J., Prasad, M., Willems, G., Timofte, R., van Gool, L.: Hough Transform and 3D SURF for Robust Three Dimensional Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 589–602. Springer, Heidelberg (2010)
12. Taylor, G., Fergus, R., LeCun, Y., Bregler, C.: Convolutional Learning of Spatio-temporal Features. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 140–153. Springer, Heidelberg (2010)
13. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR (2011)
14. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for human action recognition. In: CVPR (2010)
15. Uemura, H., Ishikawa, S., Mikolajczyk, K.: Feature tracking and motion compensation for action recognition. In: BMVC (2008)

16. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
17. Wang, F., Jiang, Y.G., Ngo, C.W.: Video event detection using motion relativity and visual relatedness. In: ACM MM (2008)
18. Raptis, M., Soatto, S.: Tracklet Descriptors for Action Modeling and Video Analysis. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 577–590. Springer, Heidelberg (2010)
19. Matikainen, P., Hebert, M., Sukthankar, R.: Representing Pairwise Spatial and Temporal Relations for Action Recognition. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part I. LNCS, vol. 6311, pp. 508–521. Springer, Heidelberg (2010)
20. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: CVPR (2009)
21. Turaga, P., Chellappa, R., Subrahmanian, V.S., Udrea, O.: Machine recognition of human activities: a survey. IEEE TCSVT 18, 1473–1488 (2008)
22. Poppe, R.: Survey on vision-based human action recognition. IVC 28, 976–990 (2010)
23. Lucas, B.D., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: IJCAI (1981)
24. Dalal, N., Triggs, B.: Human Detection Using Oriented Histograms of Flow and Appearance. In: Leonardis, A., Bischof, H., Pinz, A. (eds.) ECCV 2006. LNCS, vol. 3952, pp. 428–441. Springer, Heidelberg (2006)
25. Farneback, G.: Two-Frame Motion Estimation Based on Polynomial Expansion. In: Bigun, J., Gustavsson, T. (eds.) SCIA 2003. LNCS, vol. 2749, pp. 363–370. Springer, Heidelberg (2003)
26. Dalal, N., Triggs, B.: Histogram of oriented gradients for human detection. In: CVPR (2005)
27. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2008)
28. Kuehne, H., Jhuang, H., Garrote, E., Poggio, T., Serre, T.: HMDB: A large video database for human motion recognition. In: ICCV (2011)
29. Yuan, J., Wu, Y., Yang, M.: Discovery of collocation patterns: from visual words to visual phrases. In: CVPR (2007)
30. Gilbert, A., Illingworth, J., Bowden, R.: Action recognition using mined hierarchical compound features. IEEE TPAMI 33, 883–897 (2011)
31. Maji, S., Berg, A.C., Malik, J.: Classification using intersection kernel support vector machines is efficient. In: CVPR (2008)
32. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
33. Niebles, J.C., Chen, C.W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part II. LNCS, vol. 6312, pp. 392–405. Springer, Heidelberg (2010)
34. Ullah, M.M., Parizi, S.N., Laptev, I.: Improving bag-of-features action recognition with non-local cues. In: BMVC (2010)
35. Liu, J., Kuipers, B., Savarese, S.: Recognizing human actions by attributes. In: CVPR (2011)
36. Brendel, W., Todorovic, S.: Learning spatiotemporal graphs of human activities. In: ICCV (2011)
37. Serre, T., Wolf, L., Bileschi, S., Riesenhuber, M., Poggio, T.: Robust object recognition with cortex-like mechanisms. IEEE TPAMI 29, 411–426 (2007)