Middle-Level Representation for Human Activities Recognition: The Role of Spatio-Temporal Relationships

Fei Yuan[1], Véronique Prinet[1], and Junsong Yuan[2]

[1] LIAMA & NLPR, CASIA, Chinese Academy of Sciences, Beijing, China 100190
{fyuan,prinet}@nlpr.ia.ac.cn
[2] School of EEE, Nanyang Technological University, Singapore 639798
jsyuan@ntu.edu.sg

Abstract. We tackle the challenging problem of human activity recognition in realistic video sequences. Unlike local features-based methods or global template-based methods, we propose to represent a video sequence by a set of *middle-level parts*. A part, or *component*, has *consistent spatial structure* and *consistent motion*. We first segment the visual motion patterns and generate a set of middle-level components by clustering keypoints-based trajectories extracted from the video. To further exploit the interdependencies of the moving parts, we then define *spatio-temporal relationships* between pairwise components. The resulting descriptive middle-level components and pairwise-components thereby catch the essential motion characteristics of human activities. They also give a very compact representation of the video. We apply our framework on popular and challenging video datasets: Weizmann dataset and UT-Interaction dataset. We demonstrate experimentally that our middle-level representation combined with a χ^2-SVM classifier equals to or outperforms the state-of-the-art results on these dataset.

1 Introduction

Human activity patterns extraction and categorization sparked considerable interest in Computer Vision community these recent years. Recent literature has shifted from actions classification in controlled acquisition conditions [1–3], to complex activities classification in more realistic scenarios[4–6]. In the latter situation, the challenge stems from structured property of activity themselves; specifically, the complicated spatio-temporal relationships between a set of body parts or multiple persons often exhibit low intra-similarity and large inter-variability. In addition, it suffers from the free acquisition setting for realistic scenarios, which makes possible camera motion, illumination changes or occlusions.

In terms of activity representation, most models fall into the following three categories: *local feature-based* [2, 7–9], *part-based* [10, 11], and *global template-based* [12]. Local features-based models, *e.g.* bag-of-features model [7], and global template-based models [12] are widely used; they achieve impressive results in

K.N. Kutulakos (Ed.): ECCV 2010 Workshops, Part I, LNCS 6553, pp. 168–180, 2012.

certain situations. Local features-based models and global template-based models, however, have limitations to represent complicated activities. For the former, local features only contain limited spatio-temporal information, insufficient for representing complex activities. For the latter, the global templates are not flexible enough to capture intra-class variations of activities. Middle-level features, which connect local features and global features, are apparently suitable to represent complex activities. Nevertheless, part-based models are not widely employed. The possible reason is that it is difficult to generate middle-level features from low-level features, and the problem will become extremely complicated if we try to decompose a global template into a set of interactive parts.

In this work, we propose to represent complex activities by a set of middle-level features, which we call *spatio-temporal parts*. We define a part as a component which has consistent spatial structure and consistent motion in temporal domain, *e.g.* an extending arm. Unlike previous part-based models, such as Hidden Part Model [10] or Constellation model [13], in which the parts are hidden or abstract, without physical meaning, our middle-level components are concrete: they correspond to certain physical entities of the body and embed semantic information. For example, *an extending arm* means that it is an arm and the arm is extending. Furthermore, our model is free from complex and computationally heavy learning and inference on a graph.

Trajectory-based representation of activity got encouraging results [5, 14, 15]. We go further in this direction. We generate middle-level components by grouping similar trajectories. First, motion salient keypoints are selected and tracked, to form a set of trajectories. Then middle-level components are obtained by clustering trajectories of similar appearance and motion. A component is *a bundle of trajectories*. To exploit the structural and dynamic property of activities, we define spatio-temporal relationships that encode co-dependence between components. As a result, a video sequence is represented by a set of middle-level components and their spatio-temporal relationships.

Our contribution is threefold: i) we propose to represent a video by a set of middle-level components, or motion-parts; to this end, we introduce a hierarchical clustering approach (section 3.1); ii) we develop a new motion descriptor computed on components (section 3.2); iii) we model the spatio-temporal relationships between pairwise components (section 3.3).

The rest of this paper is organized as follows. Section 2 discusses related work. Section 3 gives a detailed description of our approach for generating and representing middle-level components and their relationships, and classifying activity. We illustrate and interpret the experimental results in Section 4. Finally we present the conclusion in Section 5.

2 Related Work

Video representation is of great interests in computer vision. A large panoply of work are dedicated to this problem, proposing new detectors or descriptors. To cite only a few: Spatio-Temporal Interest Points (STIPs) [16], shape descriptor [3], motion descriptor [17], shape-motion [18], static features [19], or hybrid features [20]. Among these, local detectors such as STIPs are very popular. Impressive results are reported in realistic video sequences with low resolution, camera motion, and illumination changes. However, STIPs-like feature only represents local and limited information in a spatio-temporal volume. Although much effort has been put to add spatio-temporal information between these local features [6, 21], the problem is still far from being solved.

The second, different, trend is the global approach. In these methods, a set of global templates are first built, then they are matched with testing video sequences. To improve the flexibility of templates, [12] presented a deformable action templates model by learning a best set of primitives and weights. [22] proposed to split the entire template into a set of parts, then match each part individually.

At an intermediate level between local and global approaches, several middle-level feature-based classification models have been proposed. [10] introduced a discriminative part-based approach, in which a human action is modelled by a flexible constellation of parts. In [13], the authors proposed a hierarchical action model, in which the lower layer corresponds to local features, the higher layer describes a constellation of P parts; each part is associated to a bag of features, and the relative positions of parts are taken into account. A human body part-based approach was developed in [11]. The human body is divided into elementary points from which a decomposable triangulated graph is built. The graph structure represents the spatial configuration of body parts and the velocity distribution of nodes encodes the temporal variation of activity.

Different from simple periodic actions, activities often correspond to one person performing a series of elementary actions or multiple persons interacting with each other. The spatio-temporal relations among local or middle-level features are important to help understand the human motion and recognizing activities. Related to our work, the authors of [6] proposed a spatio-temporal matching kernel to measure structural similarity between sets of features. They first define seven temporal relationships and four spatial relationships between local features based on their 3-D coordinates. Then a 3-D spatial histogram and a 3-D temporal histogram, in which each bin contains designated pairs of two feature points from a video, are used to capture the appearance and relationship information of a video. Histogram intersection is finally used to match two histograms.;

3 Our Approach

We propose to extract middle-level components to describe and classify human activities in videos. In the first stage, we segment the motion into ‘meaningful parts’, or *components*. To this end, we take a bottom-up strategy and cluster moving parts which are spatially and temporally consistent. We proceed by hierarchical clustering : linking key-points into trajectories, then clustering trajectories into middle-level components. In the second stage, we represent each video by a histogram of quantized *motion descriptor* computed at each component. *Co-components*, or pairwise components, are also considered by taking into account spatio-temporal relationships. Classification is performed by SVM classifier, using this global video representation as input features. Figure 1 gives an overview of our method.

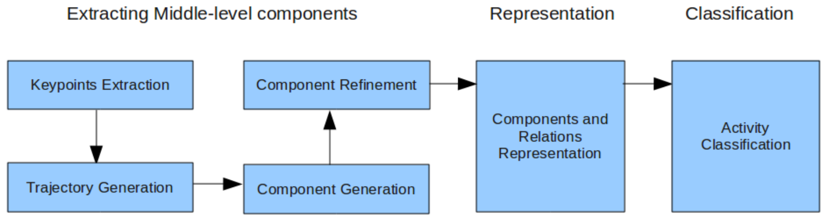


Fig. 1. From keypoints to middle-components and the global representation of videos

3.1 From Trajectories to Middle-Level Components

We aim to extract descriptive motion-parts, or components, of human activities from videos. We use a hierarchical approach that progressively generates features at different levels of semantic : *keypoint*, *trajectory*, *component*. *Keypoints* $\{p_1, p_2, \ldots, p_m\}$ are described by their 2D coordinates and associated local features computed on local patches; *trajectories* $\{t_1, t_2, \ldots, t_n\}$ are generated by tracking local keypoints in the image sequences; *components* $\{c_1, c_2, \ldots, c_l\}$ are bundles of trajectories, resulting from a graph-clustering algorithm.

In order to create reliable and robust trajectories, it is critical to extract good keypoints. In [14, 15], trajectories are produced by tracking spatially salient points, such as SIFTs or corners. However, in low-resolution videos with cluttered background and fast motion, the number of resulting keypoints is far from sufficient and trajectories are not reliable. Instead, we propose to extract densely sampled points from each frame. Then the correspondence between keypoints in successive frames is established by nearest neighbour distance ratio matching [23].

Several heuristics are proposed to obtain reliable and useful trajectories. For a keypoint p in the frame t, we only match it with keypoints in the frame $t+1$ which are located within a $N * N$ spatial window around it. The window size depends on the maximal velocity. We also discard short trajectories of limited length. For a trajectory with N keypoints with coordinates $(x_1, y_1), ..., (x_N, y_N)$, the average displace $disp = \frac{1}{N}\sum_t (x_t - \overline{x})$ is computed, and trajectories whose displace $disp < th_d$ are removed.

We turn now to the key problem of clustering trajectories into descriptive consistent motion-parts, namely *components*. Current work on motion segmentation from trajectories often only use the 2D or 3D location information of points on the paths [24]. However, in complex scenes, *e.g.* with low motion contrast or cluttered background, this approach cannot lead to satisfying segmentation. We propose an alternative and more robust solution, which describe keypoints, hence trajectories, by multi-features: location $L = (x_t, y_t)$, displacement $D = (x_t - x_{t+1}, y_t - y_{t+1})$, and intensity histogram I computed on a patch centred at each keypoint. We then define the distance between trajectory i and trajectory j as follows :

$$d_{ij} = \frac{1}{|t_2 - t_1|} \sum_{t=t_1}^{t_2} d_L(L_t(i), L_t(j)) \;\; (\alpha_1 \, d_I(I_t(i), I_t(j)) \; + \; \alpha_2 \, d_D(D_t(i), D_t(j))) \quad (1)$$

where, t_1 and t_2 are the start-frame and end-frame of the two overlapping trajectories i and j; $d_L(L_t(i), L_t(j))$ is the spatial distance between two points from two trajectories in frame t, $d_I(I_t(i), I_t(j)) = \exp(-\frac{1}{2\sigma^2} \sum_{k=1}^{r} \min(I_t^k(i), I_t^k(j)))$ and $d_D(D_t(i), D_t(j)) = ||D_t(i) - D_t(j)||_2$ are distances between two intensity histograms $(I_t(i), I_t(j))$ and displacements $(D_t(i), D_t(j))$ respectively. α_1 and α_2 are weights that balance the contribution of each features. The role of the spatial distance measure $d_L(L_t(i), L_t(j))$ is to favor the clustering of trajectories that are close to each other spatially.

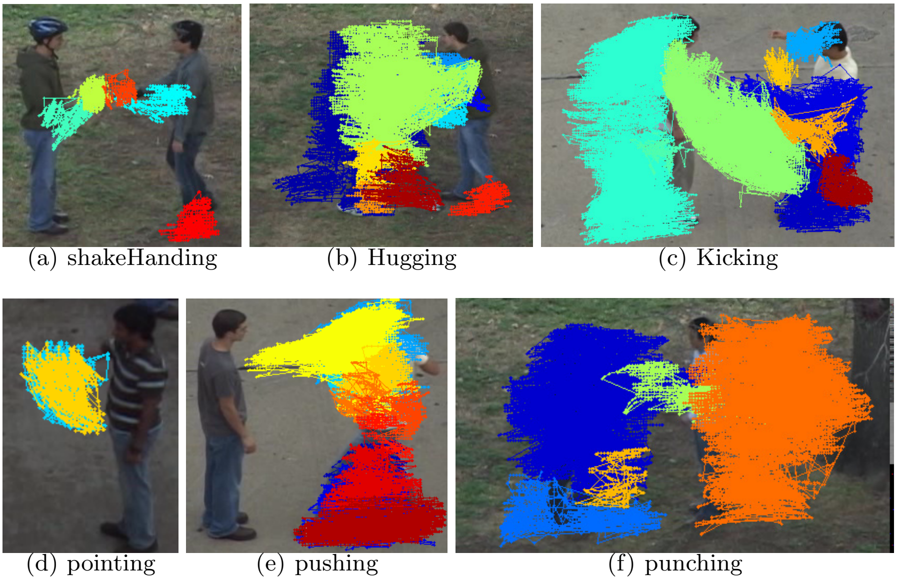
Once the distance matrix among trajectories is computed, we use an efficient graph-based partitioning algorithm to cluster trajectories [25]. The main principle of this algorithm is to disconnect nodes of the graph (trajectories are associated to nodes in our case) which are linked by a 'weak' edge. Hence, the approach relies on a threshold parameter T to cut the edges. In order to alleviate the sensibility of the clustering to this parameter T, we proceed in a cut-and-merge manner: we first over-segment the graph with a 'hard' parameter value; then use the resulting partitions as nodes of a new graph, and apply the graph-based partitioning algorithm again to merge these partitions which are similar. The dissimilarity measure between two partitions c_k and c_l that we use is the following :

$$d_{c_k c_l} = ||LC_{c_k} - LC_{c_l}|| \cdot ||MC_{c_k} - MC_{c_l}|| \qquad (2)$$

where LC_p is the centroid of partition p, MC_p is a matrix motion descriptor of partition p (which we will describe in section 3.2); $||.||$ is the Euclidean distance (or equivalently, the Frobenus distance for MC descriptor).

This two-phase partitioning procedure alleviates the dilemma of choosing the parameter T and produces reliable and robust *middle-level components.* It is worth mentioning that the entire hierarchical clustering method, from keypoints to components, is very fast because the number of trajectories and motion-parts is relatively small.

Figures 2 and 3 show 2D and 3D visualizations of the components extracted from image sequences of 6 activity classes in HT-Interaction dataset. Different components are shown by different colours, keypoints in the same component share the same colour, and keypoints in the same trajectory are connected by white lines.



(a) shakeHanding (b) Hugging (c) Kicking

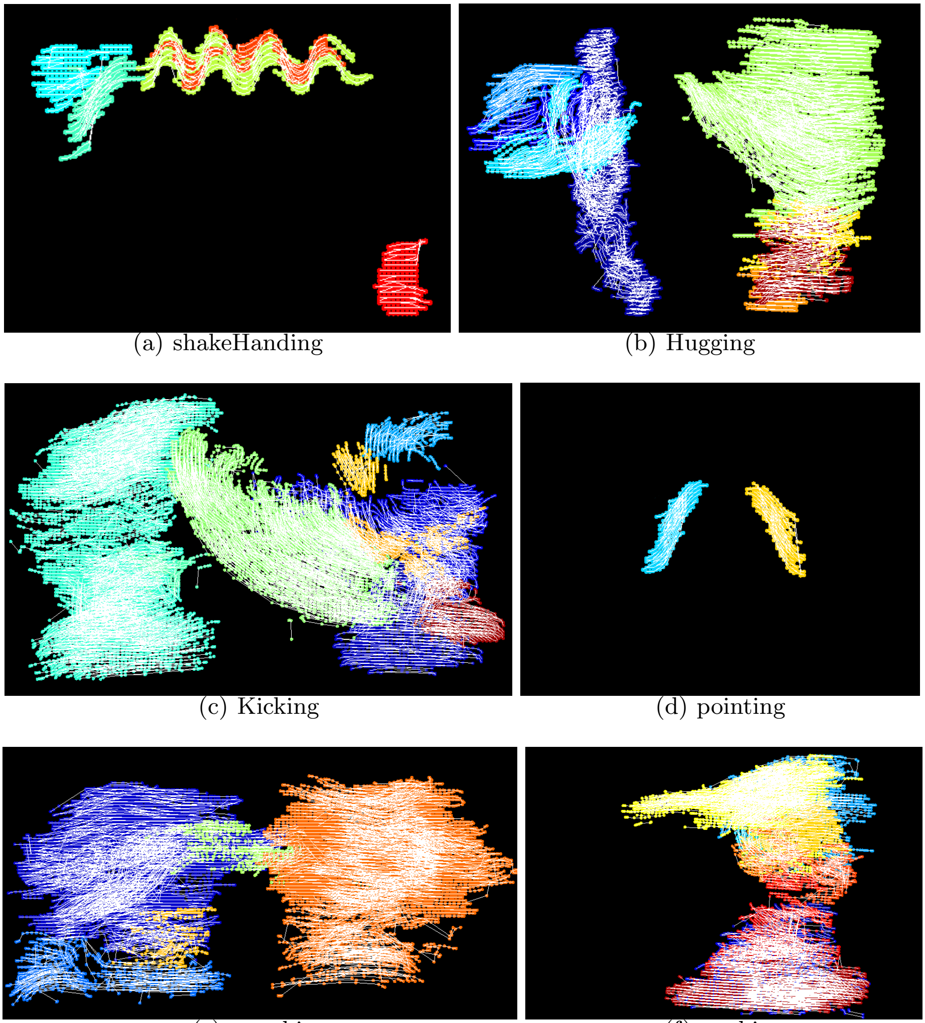
(d) pointing (e) pushing (f) punching

Fig. 2. Illustration of the components. Each component contains a collection of trajectories that are of consistent motion. Different colors correspond to different components. Only the largest components are shown.

3.2 Components Motion Descriptor

Inspired by Johansson's work on visual interpretation of biological motion [26], which showed that humans can recognize actions merely from the motion of a few moving lights attached to the human body, and subsequent works for action recognition and detection [27], we argue that motion is the most critical feature for recognizing activities. Thus, similar to the idea of 'Trajectory translation descriptor' proposed by Sun [14], we introduce a robust *motion descriptor*. This new descriptor is based on a piecewise linear approximation of the whole trajectory/component.

Each trajectory in a component defines a parametric curve of the 3D spatio-temporal space. Following Rao' approach to segment a curve into line segments [28], we first smooth each trajectory-curve using anisotropic diffusion [29], then compute its spatio-temporal curvature. The extrema of the curvature, which capture both the speed and direction changes, divide a trajectory into several line segments. The orientations of line segments are quantized into S states, including a 'no-motion' state. Thereby, we compute a $S \times S$ state transition matrix for each trajectory, and sum over all trajectories. If a trajectory contains only a single segment, the matrix contains a single non-zero element on its diagonal. We use the resulting matrix MC to describe the overall motion of a component.



(a) shakeHanding

(b) Hugging

(c) Kicking

(d) pointing

(e) punching

(f) pushing

Fig. 3. Spatio-temporal visualization of components in the 6-class HT-Interaction dataset. Only the largest components are shown.

3.3 Spatio-temporal Relationships between Components

Individual components alone are not sufficient for classification, because they do not integrate information of neighborhood components. Instead of treating components as independent ones, we propose to incorporate spatial and temporal relationships between components.

We describe the spatial and temporal relationships between components respectively, and quantize them into discrete states. For spatial relationship, we project the coordinates of keypoints of a component into the $x-y$ axis, and use the average coordinates $(\overline{x}_c, \overline{y}_c)$ to characterise the spatial location of a component C. Then the distance between the average coordinates of two components is computed. If the distance is small, it is quantized into a bin; otherwise, we compute the arc-tangent value of the spatial locations of two components, and quantize it into K_s bins. For temporal dependencies, similar to [6], we define $K_t = 5$ types of relationship: *before*, *meeting*, *overlapping*, *equalling*, *inclusion*.

A pairwise component unit, or *co-components* is defined as two components and a directed edge between them. The type of an edge is decided by the spatio-temporal relationship between components, which belongs to one of $(K_s+1) \times K_t$ spatio-temporal relationships, and the direction of an edge is mainly decided by the precedence relationship of the centroids of two components $\overline{t_1}$ and $\overline{t_2}$. If $\overline{t_2} > \overline{t_1}$, the direction of the edge is from component C_1 to C_2, which means that C_1 happens before C_2. For inclusion relationship, the direction of the edge is from component C_1 to C_2 if component C_1 includes component C_2. After that, we represent a pairwise component unit with a descriptor by concatenating the descriptors of two components. For the equal relationship, a bidirectional edge is used, and two descriptors, concatenating C_1 to C_2 and C_2 to C_1, are used.

In total, there are $K = (K_s + 1) \times K_t$ spatio-temporal relationships. We consider all possible pairs of components. Because the number of components in a video sequence is small, for example, from one to several dozen, it is fast to perform the above operations. In our experiments, we found that this representation encodes robustly complicated spatial and temporal interactions between components.

3.4 Global Representation of Activity

We follow a *bag-of-features* approach to represent the video sequences, using motion matrix of co-component as features.

During the learning phase, we first vectorize the motion matrices of components extracted from the videos, then quantize the resulting motion vectors into

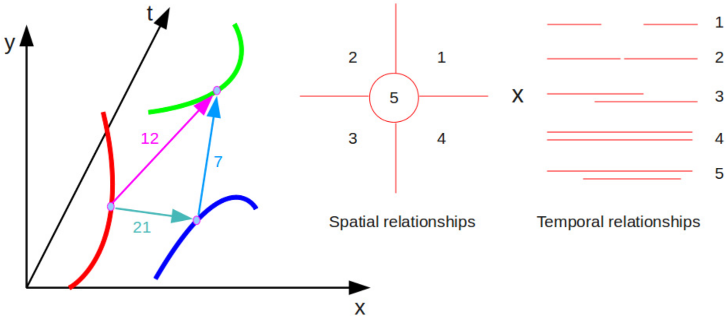


Fig. 4. Examples of pairwise components and their spatio-temporal relationships. Each bold curve stands for a component, and each line with an arrow stands for the edge of a co-component. The relationship between a pairwise component unit is quantized into one of several types.

B bins, using a k-means clustering algorithm. This enables to create a codebook. During the test phase, we assign the closest codebook entry to each component motion descriptor of a new video sequence, and generate a histogram of quantised descriptors.

When using individual components alone (without taking into account the spatial and temporal relationships), the whole video is represented by a histogram of B bins. When using co-components, during the training phase, for each of the K spatio-temporal relationships, we generate a codebook of dimension B'. During the testing phase, we assign the pairwise components descriptors to their closest entry of the codebook associated to the same type of edge. We then generate K histograms, and concatenate these histograms to represent the video.

3.5 Activity Recognition

Based on a bag-of-features representation of video sequences with motion descriptors as features, we use the SVM with χ^2 kernel to classify activities. One-against-all strategy is adopted for the multi-classes classification task .

4 Experiments

4.1 Practical Details

Keypoints sampling. Keypoints are densely sampled at regular (space) interval L: every $L \times L$ patch is represented by a keypoint. L might vary according to the resolution of the input video. We chose $L = 3$ in our experiments.

Weights (α_1, α_2). Their values are fixed to $(0.5, 0.5)$.

Trajectories. We set the size of searching window for matching keypoints to $N = 15$. Short trajectories whose length is less than 5 frames are discarded, and trajectories with average displacement $disp < 4$ are removed.

Keypoints descriptor. We compute an intensity histogram over a 12×12 patch centred at each keypoint. Image intensity level is quantized into 20 bins.

Clustering parameter (t_1, t_2). Their values are fixed to $(0.55, 0.5)$.

Component motion descriptor. The orientation of line segments is quantized into S states (bins) including a no-motion state. The velocity of the line segments can also be quantized. We analyse in the next section the impact of the quantization on the classification rate.

Codebook size. Motion descriptor of single components are quantized in a codebook of size $B = 300$. For co-components, the size of total codebook is 4280.

4.2 Dataset

We test our algorithm on two datasets: Weizmann dataset [3] and UT-Interaction dataset [6]. The Weizmann dataset comprises videos with an homogeneous and static background. It consists in ten action classes: bending, running, walking, skipping, jacking, jumping forward, jumping in place, galloping sideways, waving with two hands, and waving with one hand. Each video contains a single periodic action, performed by a single person. The UT-Interaction dataset contains six classes of human-human interactions: shake-hands, point, hug, push, kick and punch. In this dataset, 6 participants performed activities with 10 different clothings in different background, scale, and illumination conditions.

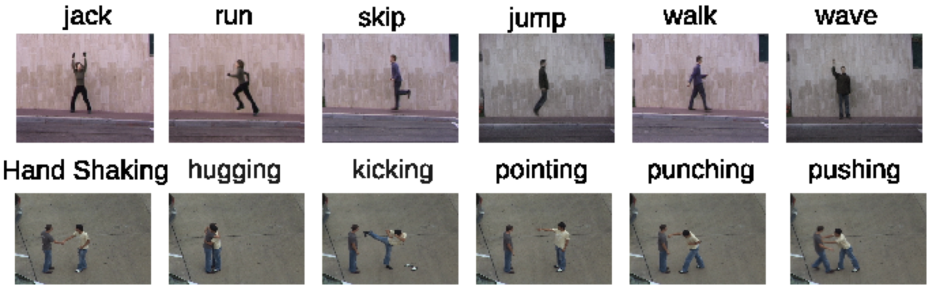


Fig. 5. Snapshots of video sequences of Weizmann (top) and UT-Interaction (bottom) datasets

4.3 Experimental Setting and Results

During the learning phase, in Weizmann dataset, training is done on 8 subjects and testing is done on all video sequences of the remaining subjects. In UT-Interaction dataset, we use 15 video sequences for training and the rest for testing.

Table 1. Classification accuracy on 6 classes of Weizmann dataset. We use here motion descriptor of independent components.

Representation	Bend	Run	Jack	Jump	Wave 1	Wave 2	Ave.
[30]	100	100	100	77.8	100	100	96.3
Components	100	100	100	100	100	100	100

We first evaluate the performance of our motion descriptor on individual components, without taking into account the spatio-temporal relations between them (components are assumed to be independent). The rate of correct classification computed on each class from Weizmann dataset is given in Table 1. For the six classes we tested, our approach reaches 100% of correct classification.

We then evaluate the role of pairwise relationships, *e.g.* co-component motion descriptors (see section 3.3). The UT human-interaction dataset is particularly well suited for this task because it comprises complex inter-actions between people and therefore is a very challenging dataset. Accuracy results are given in Table 2. In this table, we evaluate our middle-level representation with motion features computed on individual components and co-components. For comparison, we show classification results obtained from bag-of-features generated from point-wise descriptors: histograms of gradient (HOG) and histograms of optical-flow (HOF) computed around points extracted with STIP detectors [16]. We give also the recent results of Ryoo [6]. In [6] the spatio-temporal relationships among local STIPs are represented by a three-dimensional spatial histogram and a three-dimensional temporal histogram; histogram intersection is used to measure the similarity between spatial histograms and temporal histograms.

From Table 2, we observe that STIP-based features give the best results for 3 out of 6 classes. Comparatively, [6] gives relatively poor performances –*i.e.* no winning class– compared to STIP- or component- based classification results. On this dataset, bag-of-co-components representation combined with our motion descriptor improves on the state-of-the-art results.

Table 2. Classification accuracy results on UT-interaction dataset. Our approach is compared with others. In bold are highlighted best classification results for each class.

Detector/Descriptor	HandShake	Hug	Kick	Point	Punch	Push	Ave.
STIPs / HOG	65	60	65	90	25	75	63.25
STIPs / HOF	65	85	**90**	85	55	65	74.2
STIPs / HOG+HOF	75	**95**	70	70	**75**	70	75.8
[6]	**75**	87.5	75	62.5	50	75	70.8
Components / motion	**75**	80	75	94.7	50	40	69.1
Co-components / motion	70	80	85	**100**	55	**80**	**78.2**

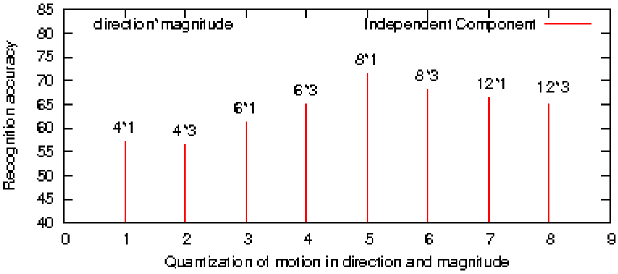


Fig. 6. Impact of quantization of motion descriptor: accuracy results for varying number of orientation bins (from 4 to 12) and amplitude bins (from 1 to 3) in the quantization process

Finally, we analyse the impact of parameter setting of our motion descriptor on the classification results. Our motion descriptor relies on the quantisation of orientation angle and displacement amplitude. We vary the number of bins used in this quantisation procedure, from 4 to 12 for the former, from 1 to 3 for the latter. Results on UT database are illustrated in Figure 6.

5 Conclusion and Future Work

In this paper, we introduced a new middle-level representation, namely *components*, to describe human activities in videos, and proposed a motion descriptor that captures essential characteristics of human motions. We also have shown that taking into account the spatial and temporal relationships between components improves significantly the performance of the classification. Experimental results show that recognition results based on our middle-level components and motion descriptor improve the state-of-the-art results on the most recent UT-interaction database. In future work, we will add more information for our middle-level components, and extend the work to activity detection task.

Acknowledgments. This work is supported by the 863 program of the Ministry of Science and Technology of China. Junsong Yuan is supported in part by the Nanyang Assistant Professorship (SUG M58040015).

References

1. Sullivan, J., Carlsson, S.: Recognizing and Tracking Human Action. In: Heyden, A., Sparr, G., Nielsen, M., Johansen, P. (eds.) ECCV 2002, Part I. LNCS, vol. 2350, pp. 629–644. Springer, Heidelberg (2002)
2. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: Proc. of Int. Computer Vision and Pattern Recognition, CVPR (2004)
3. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proc. of Int. Conf. on Computer Vision, ICCV, pp. 1395–1402 (2005)
4. Li, R., Chellappa, R.: Recognizing coordinated multi-object activities using a dynamic event ensemble model. In: Proc. of Int. Acoustics, Speech, and Signal Processing, pp. 3541–3544 (2009)
5. Turaga, P., Chellappa, R.: Locally time-invariant models of human activities using trajectories on the grassmannian. In: Proc. of Int. Computer Vision and Pattern Recognition, CVPR, pp. 2435–2441 (2009)
6. Ryoo, M.S., Aggarwal, J.K.: Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In: Proc. of Int. Conf. on Computer Vision, ICCV, pp. 1593–1600 (2009)
7. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: Proc. of Conf. on Computer Vision and Pattern Recognition, CVPR, pp. 1–8 (2008)
8. Yuan, J., Liu, Z., Wu, Y.: Discriminative video pattern search for efficient action detection. In: Proc. of Int. Computer Vision and Pattern Recognition, CVPR (2009)
9. Cao, L., Liu, Z., Huang, T.S.: Cross-dataset action recognition. In: Proc. of Int. Computer Vision and Pattern Recognition, CVPR (2010)
10. Wang, Y., Mori, G.: Learning a discriminative hidden part model for human action recognition. In: Advances in Neural Information Processing Systems, NIPS, vol. 21 (2008)
11. Thi, T.H., Lu, S., Zhang, J., Cheng, L., Wang, L.: Human body articulation for action recognition in video sequences. In: Proc. of Int. Conf. on Advanced Video and Signal Based Surveillance, pp. 92–97 (2009)

12. Yao, B., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: Proc. of Int. Conf. on Computer Vision, ICCV, pp. 1507–1514 (2009)
13. Niebles, J., Fei-Fei, L.: A hierarchical model of shape and appearance for human action classification. In: Proc. of Conf. in Computer Vision and Pattern Recognition, CVPR (2007)
14. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: Proc. of Conf. on Computer Vision and Pattern Recognition, CVPR, pp. 2004–2011 (2009)
15. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: Proc. of Int. Conf. on Computer Vision, ICCV, Washington, DC, USA (2009)
16. Laptev, I., Lindeberg, T.: On space-time interest points. In: Proc. Int. Conf. on Computer Vision, ICCV, pp. 432–439 (2003)
17. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. Int. Conf. on Computer Vision, ICCV (2003)
18. Lin, Z., Jiang, Z., Davis, L.: Recognizing actions by shape-motion prototype trees. In: Proc. Int. Conf. on Computer Vision, ICCV, pp. 444–451 (2009)
19. Li, L.J., Fei-Fei, L.: What, where and who? classifying event by scene and object recognition. In: Proc. of IEEE Intern. Conf. in Computer Vision, ICCV (2007)
20. Liu, J., Luo, J., Shah, M.: Recognizing realistic actions from videos 'in the wild'. In: Proc. of Int. Computer Vision and Pattern Recognition, CVPR, pp. 1996–2003 (2009)
21. Wu, J., Osuntogun, A., Choudhury, T., Philipose, M., Rehg, J.M.: A scalable approach to activity recognition based on object use. In: Proc. of IEEE Intern. Conf. in Computer Vision, ICCV (2007)
22. Ke, Y., Sukthankar, R., Hebert, M.: Event detection in crowded videos. In: Proc. of IEEE Intern. Conf. in Computer Vision, ICCV (2007)
23. Mikolajczyk, K., Schmid, C.: A performance evaluation of local descriptors. IEEE Trans. on Pattern Analysis & Machine Intelligence 27, 1615–1630 (2005)
24. Rao, S., Tron, R., Vidal, R., Ma, Y.: Motion segmentation in the presence of outlying, incomplete, or corrupted trajectories. IEEE Trans. on Pattern Analysis and Machine Intelligence 99 (2009)
25. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. Int. J. Comput. Vision 59, 167–181 (2004)
26. Johansson, G.: Visual perception of biological motion and a model for its analysis. Perception and Psychophysics 14, 201–211 (1973)
27. Song, Y., Goncalves, L., Bernardo, E.D., Perona, P.: Monocular perception of biological motion in johansson displays. Comput. Vis. Image Underst. 81, 303–327 (2001)
28. Rao, C., Yilmaz, A., Shah, M.: View-invariant representation and recognition of actions. Int. J. Comput. Vision 50, 203–226 (2002)
29. Perona, P., Malik, J.: Scale-space and edge detection using anisotropic diffusion. IEEE Transactions on Pattern Analysis and Machine Intelligence 12, 629–639 (1990)
30. Junejo, I.N., Dexter, E., Laptev, I., Pérez, 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)