Bag of Graphs with Geometric Relationships Among Trajectories for Better Human Action Recognition

Manel Sekma[1](✉), Mahmoud Mejdoub[1,2], and Chokri Ben Amar[1]

[1] REGIM: Research Groups on Intelligent Machines, University of Sfax,
National School of Engineers (ENIS), Sfax 3038, Tunisia
manel_sekma@ieee.org

[2] Department of Computer Science, College of AlGhat, Majmaah University,
P.O. BOX 445, Al Majmaah, Riyadh 11914, Kingdom of Saudi Arabia

Abstract. This paper presents a new video representation that exploits the geometric relationships among trajectories for human action recognition. Geometric relationships are provided by applying the Delaunay triangulation method on the trajectories of each video frame. Then, graph encoding method called bag of graphs (BOG) is proposed to handle the geometrical relationships between trajectories. BOG considers local graph descriptors to learn a more discriminative graph-based codebook and to represent the video with a histogram of visual graphs. The graph-based codebook is composed of the centers of graph clusters. To define graph clusters, a classification graph technique based on the Hungarian distance is proposed. Experiments using the human action recognition datasets (Hollywood2 and UCF50) show the effectiveness of the proposed approach.

1 Introduction

Recognizing human actions in realistic uncontrolled video is a challenging problem in computer vision due to the numerous applications such as video-surveillance [1], human computer interaction [2], video indexing and retrieving [1,3,4]. In these years, many methods have been proposed on the recognition of human actions in video. Local features, coupled with the bag-of-words model (BOW), have recently become a very popular video representation for action recognition [5–19]. The BOW model [20–23] use a codebook to create a descriptor based on the visual content of a video, where the codebook is a set of visual words that represents the distribution of local features of all video. This concept can be summarized on two successive steps: coding, which assigns the local descriptors according to a codebook; and spatial pooling, which aggregates the assigned words into a single feature vector. The standard BOW model encodes only the global distribution of features, by computing a disordered histogram of occurrences of visual words, which ignore the local structural organization of features. While it is obvious that using such local structure of features should help in video action classification.

V. Murino and E. Puppo (Eds.): ICIAP 2015, Part I, LNCS 9279, pp. 85–96, 2015.
DOI: 10.1007/978-3-319-23231-7_8

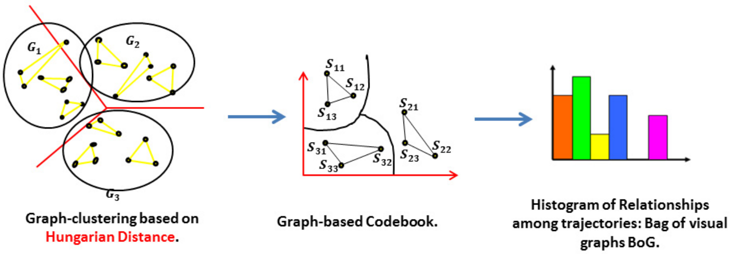
G_1
G_2
G_3
Graph-clustering based on Hungarian Distance.
S_{11}
S_{12}
S_{13}
S_{21}
S_{22}
S_{23}
S_{31}
S_{32}
S_{33}
Graph-based Codebook.
Histogram of Relationships among trajectories: Bag of visual graphs BoG.

Fig. 1. Visual graph quantization.

To overcome this limitation and to provide the spatial relationships between local features, we propose in this paper an approach that exploits the relationships among video trajectories based on graph encoding. To build the graphs, we apply Delaunay triangulation method [24] to link spatially near trajectories. Each graph vertex is labeled by four low level descriptors computed for its corresponding trajectory shape. Then, a graph descriptor is defined by the set of descriptors that label the graph vertices. Afterwards, to handle the relationships between trajectories, we propose a graph encoding method called bag of graphs (BOG). The BOG as illustrated in Figure 1, is an extension of the traditional BOW method that represents the video using a codebook of visual graphs instead of a visual words one.

In the first step, we quantize the graph descriptors of the video into graph clusters based on the Hungarian distance [25]. Then, the graph-based codebook is formed by the centers of the graph clusters named as visual graphs. In the next step, to describe the video we build a histogram that counts the occurrences of each visual graph of the graph-based codebook (See Figure1).

The contribution of this work can be summarized as follows: we exploit the geometric relationships of trajectories by presenting them as graph descriptors, and then using the BOG model, we build a graph-based codebook by quantifying these graph descriptors. We show how the rich information embedded in graph descriptors can improve performance over standard histogram encoding method. The rest of this paper is structured as follows.

Section 2 reviews related work. In section 3 we briefly introduce the dense trajectories that we will consider through this paper. Section 4 we give a detailed description of our approach. The experimental results are given in section 5. Finally, we conclude in section 6.

2 Related Work

Related to our work, more works perform temporal tracking of local patches and encode the temporal trajectories of frame-level local patches [8,13,15,16,26] whose performance highly depends on the performance of the local descriptors of trajectories. As has been shown in the mentioned works, when local descriptors are computed over trajectories, the performance improved considerably compared to when computed over spatio-temporal features [6]. Trajectories are widely used as features to construct the codebook of visual words. Wang et al. [8] proposed a method for trajectory shape extraction by tracking densely sampled points using the optical flow fields. To encode the shape of the trajectory, the local motion and appearance around a trajectory, four types of descriptors are computed, namely $TrajectoryShape$ [8], Histograms of Optical Flow (HOF) [5], Motion Boundary Histograms (MBH) [27] and Histograms of Oriented Gradients (HOG) [28]. To encode features, the standard BOW is used. A codebook for each descriptor was built using K-means to assign each feature to the closest visual word. Based on trajectories extraction methods, several extensions to the standard BOW model have been proposed in order to build a more compact codebook. Vig et al. [29] and Mathe et al [30] have proposed to use saliency-mapping algorithms to prune background features and they focus on BOW spatio-temporal computer-based action recognition pipelines. This results in a more compact video representation. Jain et al. [16] have proposed to decompose visual motion into dominant and residual motions both for extracting trajectories and computing descriptors. They have designed a motion Divergence-Curl-Shear descriptor (DCS) to capture additional information on the local motion patterns. To encode features, they have applied the vectors of locally aggregated descriptors (VLAD) [31].

More recently, Wang et al. [26], improve dense trajectories performance [8] by taking into account camera motion to correct them. A human detector is employed to remove the inconsistent matches generated by human motion. Then, given the estimated camera motion, trajectories consistent with it are removed. To encode features, they use two features encoding methods: the standard BOW and the Fisher vector (FV) [32]. These approaches mentioned above, improve action recognition accuracy using trajectory shape methods but do not take into account the relationships between trajectories. Indeed, in order to model the relationships between trajectories, pairing techniques have been proposed. Among approaches modeling the relationships between features of dense trajectories [8], Matikainen et al. [13] have presented a method for expressing pairwise relationships between quantized features obtained by the application of the K-means algorithm. Jiang et al. [15] have proposed an approach to model the motion relationships among moving objects and the background. Their method consists in clustering the dense trajectories, and then using the cluster centers as reference points. Afterwards, the pairwise trajectory relationships are encoded by pairs of reference points.

The major problem of these methods [13,15] is that pairing visual words can leads to a quadratic number of possible pairs of visual words. Our work is different from the above techniques in that we apply quantification in the joint feature space of local graph descriptors. Then a compact graph-based codebook can be built by discovering clusters that encode the relationships between spatially close descriptors. Besides, we model more higher order spatial relationships of the video trajectories by presenting them as graph descriptors using a multi-scales triangulation.

3 Trajectory Extraction

We adopt in our approach the dense trajectory descriptors [8,26] since they showed to be an efficient video representation for action recognition. The trajectories are obtained by densely tracking sampled points using optical flow fields. First, feature points are sampled from a dense grid. Then, each feature point P_t $= (x_t,\ y_t)$ at frame t is tracked to the next frame by median filtering in a dense optical flow field $F = (ut,\ v_t)$ as equation 1.

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

where F is the kernel of median filtering and $(\overline{x}_t, \overline{y}_t)$ is the rounded position of $(x_t,\ y_t)$. The tracking is limited to $L = 15$ frames to avoid any drifting effect.

To encode the shape of the trajectory, the local motion and appearance around a trajectory, four types of descriptors are computed: Trajectory Shape, HOF, MBH and HOG. The Trajectory Shape descriptor encodes the shape of the trajectory represented by the normalized relative coordinates of the successive points forming the trajectory. It directly depends on the dense flow used for tracking points. HOF is computed using the orientations and magnitudes of the flow field. HOG encodes the appearance by using the intensity gradient orientations and magnitudes. MBH is designed to capture the gradient of horizontal and vertical components of the flow. The motion boundaries encode the relative pixel motion and therefore eliminate the movement of the camera, but only to some extent. Wang et al. [26] have recently improved trajectories performance by taking into account camera motion. A human detector is used to remove the inconsistent matches generated by human motion.

4 Description of the Proposed Approach

A trajectory is the path that a person moves as a function of time. The trajectory of a tracked person in a scene is often used to analyze the action or activity of the person. We propose to use graphs to encode geometric relationships among the trajectories obtained as described in section 3. The premise is that things (person or object) that are related spatially are usually dependent on each other. For instance, in the Figure 2.1, by taking into consideration the relationship between the person and the car door, we can incorporate useful scene context information in our description. For "Eat action" (Figure 2.2), by considering the relationship between the motion of the hand and the head of the person, we can enhance the person motion description. Figure 2 (b) represents the dense trajectories extraction and Figure 2 (c) represents the feature trajectories as a set of connected graphs of trajectories. In the rest, we present the overview of our proposed approach (Figure 3).

Firstly, for each video, low level descriptors (i.e., HOG, HOF, MBH and Trajectory Shape) are extracted. Afterwards, these descriptors are used to model the input video with graph descriptors. After graph extraction, a graph encoding method called bag of graphs (BOG) is performed. Finally, for classification a

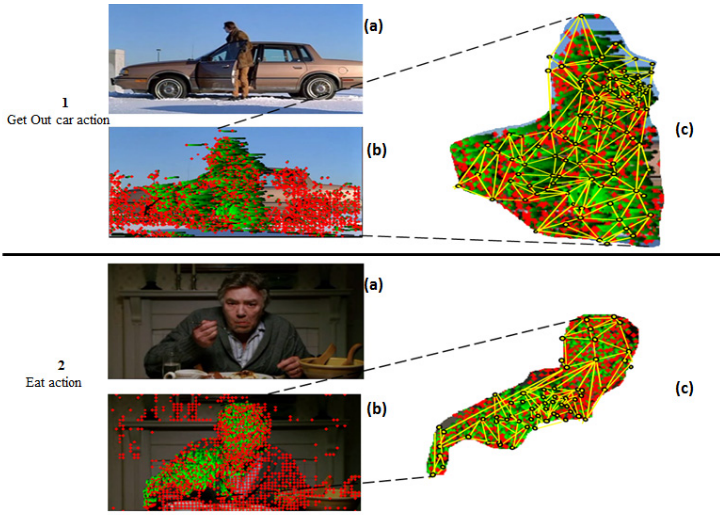

Fig. 2. Graph trajectories extraction (a) The current video frame (b) Dense trajectories; red dots indicate the trajectory shape positions in the current frame. (c) Graph of trajectories.

linear support vector machine (SVM) is used. The extraction of graph descriptors and BOG encoding method are detailed in the next subsections.

4.1 Extraction of the Graph Descriptors

The geometric relationships are provided by applying the Delaunay triangulation method [24] on the start points of the trajectories. Indeed, for each trajectory shape start point, Delaunay triangulation is applied to link neighbour trajectory points (See Figure 4). Delaunay triangulation is invariant with regard to affine transformations of image plane preserving angles: scaling, rotation and translation [33]. A collection of local graph of trajectories is constructed. Each graph vertex is labeled by the low level descriptors (i.e., HOG, HOF, MBH and Trajectory Shape) computed for its corresponding trajectory. The graph edges reflect the geometric relationships between the trajectories. A graph descriptor is defined by the set of descriptors that label the graph vertices. We notice that for each kind of low level descriptor we build a corresponding graph descriptor. Thus we obtain four graph descriptors i.e., HOG graph descriptor, HOF graph descriptor, MBH graph descriptor and Trajectory Shape graph descriptor. In this work, we adopt a multi-scale graph descriptor construction, where each scale adds more structural information. Each scale has his own set of trajectories neighbours around each given trajectory shape and then the triangulation is run separately on each scale. One scale will always contain the trajectories of all the lower scales (See Figure 4). The number of trajectories added from one scale to the upper one is fixed to three. Therefore, we define three scales, the bottom one containing only the three-nearest neighbours graph, the median

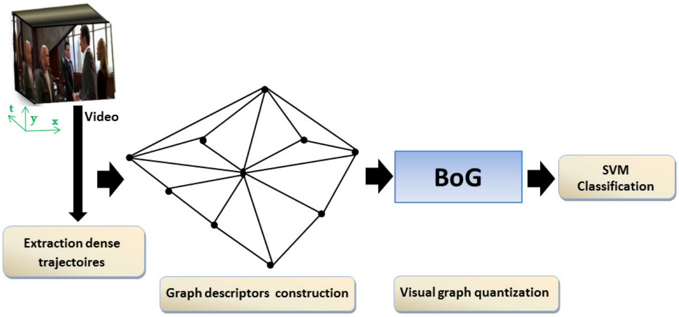
t
y
x
Video
Extraction dense trajectoires
Graph descriptors construction
BoG
Visual graph quantization
SVM Classification

Fig. 3. Description of the proposed framework.

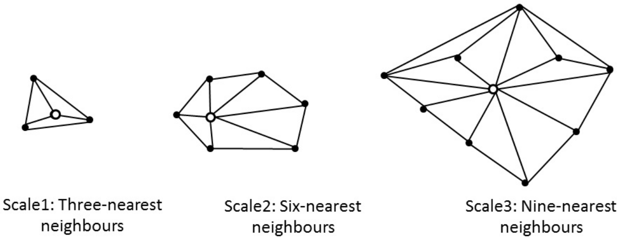
Scale1: Three-nearest neighbours
Scale2: Six-nearest neighbours
Scale3: Nine-nearest neighbours

Fig. 4. The multi-scales graphs of trajectories construction: for each trajectory start point (white node), Delaunay triangulation is applied to link neighbour trajectory points (Neighbours are in black).

containing the six-nearest neighbours graph and the top one containing a graph built upon nine-nearest neighbours (See Figure 4), resulting in a more complete local structure.

4.2 Bag of Graphs

The bag of graphs (BOG) is based on the quantification of the graph descriptors using k-means clustering method. Each obtained graph cluster represents a group of similar spatial structures of trajectories. The centers of the graph clusters correspond to the visual graphs forming the graph-based codebook. The process to generate the video description uses the graph-based codebook to compute a histogram, which counts the occurrences of the visual graphs within the video (Figure 1). For either assignment or clustering, to measure the distance of each candidate graph of a video to the visual graphs of the codebook, we consider a graph matching method based on Hungarian distance [25]. The Hungarian method is an algorithm which finds an optimal assignment between two graphs, running in $O(n^3)$ time [25], where n is the size of the graph. It is used in this work to compute the distance between two graph descriptors.

Firstly, the distances between every pair of vertices in the two graph descriptors are computed. These distances form a cost matrix which defines a vertex-to-vertex assignment for a pair of graph descriptors. Then the assignment problem is solved by the Hungarian method. Considering two graph descriptors Gi and Gj, the distance $D_{Hungarian}$ between them is given by the equation 2.

$$D_{Hungarian}(Gi, Gj) = \frac{\overline{C}}{|C|} \tag{2}$$

where $\overline{C}$ is the cost of the optimum graph matching of two graph descriptors, and $|C|$ is a normalization constant that refers to the number of matching vertices. The optimum matching cost $\overline{C}$ of a pair of graph descriptors is computed by applying the Hungarian method on the distance matrix, where each element of this matrix corresponds to the distance between two vertices of graph descriptors (described in section 4.1). For instance, in the case of three sized graphs, the distance matrix is given by the equation 3, $d(v_1^i, v_2^j)$ represent the distance between vertex 1 of the graph descriptor G_i and vertex 2 of the graph descriptor G_j. This distance is computed in terms of the euclidean distance.

$$C = \begin{pmatrix} d(v_1^i, v_1^j) & d(v_1^i, v_2^j) & d(v_1^i, v_3^j) \\ d(v_2^i, v_1^j) & d(v_2^i, v_2^j) & d(v_2^i, v_3^j) \\ d(v_3^i, v_1^j) & d(v_3^i, v_2^j) & d(v_3^i, v_3^j) \end{pmatrix} \tag{3}$$

The procedure of clustering algorithm based on graph matching method is summarized as follows: 1) Defining from candidate graph descriptors initial graphs as graph centers of the graph clusters. 2) Assigning each graph descriptor to a given graph center by applying the Hungarian distance to compute the distance between the candidate and the center of the graph cluster. 3) Updating the graph centers by averaging the assigned descriptors with the Hungarian method to each graph center vertex 4) Repeating Steps 2) and 3) until the graph centers no longer move. We note that for each low level trajectory descriptor (HOG, HOF, MBH and Trajectory Shape) and for each graph scale, we apply the BOG pipeline. Thus, 12 final graph descriptor histograms are generated derived from the 4 kinds of low level descriptors and the 3 kinds of graph scales. We apply sum pooling and L_1 normalization for each histogram, and then we horizontally concatenate them to form the final histogram.

5 Experimental Study

In our experiments, we adopt the well-performing trajectory features used in [26]. To implement the BOG model, we train a codebook for each type of low level descriptor and for each type of graph scales separately using $100,000$ randomly selected training graphs. The size of the codebook is set to 4000. The resulting histograms of visual graphs occurrences are used as video sequence representation. An SVM with RBF χ^2 kernel is used for classification.

5.1 Action Recognition Datasets

The Hollywood2 dataset [34] was collected from 69 different Hollywood movies. There are 12 action classes. In total, there are 1707 action samples divided into a training set and a testing set.

The UCF50 dataset [35] has 50 action categories, consisting of real-world videos taken from YouTube. The actions range from general sports to daily life exercises. For all 50 categories, the videos are split into 25 groups. For each group, there are at least 4 action clips. In total, there are $6,618$ video clips.

5.2 Results

In this section we present the experimental results using Hollywood2 and UCF50 datasets. Performance values are reported as recommended by the dataset authors using the already predefined training and test split, Mean Average Precision (MAP) for Hollywood2 dataset and Average Accuracy (AA) obtained by leave-one-group-out cross-validation for UCF50 dataset.

Comparison with Different Methods of Trajectory Extraction. We evaluate the performance of our approach using different methods of trajectory extraction and using BOG and BOW models. We report the results in Table 1. *Dense Traj* shows the results using basic dense trajectories [8]. Whereas *Improved Traj* shows the results of improved trajectories [26] without human detection and *Improved Traj+HD* corresponds to the results with the human detection. We denote by *combined* the results obtained by combining BOW and BOG models.

On Hollywood2, using basic dense trajectories, we obtain a MAP equal to 58.3% with BOW model and a MAP equal to 59.6% with BOG model. We can see that this result is improved by combining BOG and BOW models, thus we achieve 60%. BOW can be considered as a specific zero scale of the BOG model where none neighbour trajectories are used. Combining BOW with BOG add richer information in the representation. This performance gain mainly owns to complementarity of these two encoding methods.

With *Improved Traj* and combination of BOW with BOG, we achieve on UCF50 89.8% and 62.8% on Hollywood2. *Improved Traj* improves the dense tra-

Table 1. Influence of different methods of trajectory extraction. (Performance values are reported in the form of MAP percentages for Hollywood2 dataset and AA percentages for UCF50 dataset.)

Trajectory methods	Hollywood2	UCF50
Dense Traj+BOW	58.3	84.8
Dense Traj+BOG	59.6	86.9
Dense Traj+combined	60	87.7
Improved Traj+combined	62.8	89.8
Improved Traj+HD+combined	64.1	90.5

jectories using camera motion estimation method, which consists to remove trajectories generated by camera motion by thresholding the displacement vectors of the trajectories in the warped flow field. If the displacement is too small, the trajectory is considered to be too similar to camera motion, and thus removed.

The results are more improved by *Improved Traj+HD* on these two datasets i.e., 64.1% and 90.5% respectively. These improvements are due to the stabilized trajectories obtained using the human detector (HD), which is employed to remove the inconsistent matches generated by human motion.

Comparison with the State-of-the Art Methods. Table 2 compare our results to the state of the art approaches. On Hollywood2 dataset, Chakra et al. [36] have introduced a spatial interest points (SIP) feature. Only distinctive SIP features are kept by suppressing unwanted background features and imposing local and temporal constraints. Their approach follows a spatial pyramid based BOW improved by a vocabulary compression technique. They have reached a MAP equal to 58.46%. Cho et al. [37] have proposed a method that selects a small number of descriptors corresponding to local motion using group sparsity and emphasizes them by the multiple kernel method. This method gives 60.5%. Wang et al. [8] have reached a MAP equal to 58.3% using the dense trajectories coupled with the BOW model. In [38] the Spatio-Temporal Pyramid (STP) representation is applied on the dense trajectories giving a MAP equal to 59.9%. The presented results [15,16,26,29,30] improve basic dense trajectories [8] in different ways. Both Vig et al. [29] and Mathe et al. [30] prune background features based on saliency-mapping algorithms, they has achieved respectively 59.4% and 61%. Jiang et al. [15] have achieved 59.5% by modeling the relationship between dense trajectory clusters. Jain et al. [16] have given 62.5% using VLAD representation to encode the dense trajectories. Wang et al. [26] have improved dense trajectories by removing the camera motion in video to correct basic trajectories. They have reached 62.2% with the BOW model and

Table 2. Comparison with the state-of-the-art. (Performance values are reported in the form of MAP percentages for Hollywood2 dataset and AA percentages for UCF50 dataset.)

Hollywood2	MAP	UCF50	AA
Chakra et al. [36]	58.46	Kliper et al. [39]	72.7
Vig et al. [29]	59.4	Solmaz et al. [40]	73.7
Jiang et al. [15]	59.5	Reddy et al. [35]	76.9
Wang et al. (BOW) [8]	58.3	Shi et al. [42]	83.3
Cho et al. [37]	60.5	Wang et al. [38]	85.6
Mathe et al. [30]	61	Wang et al. (BOW)[26]	87.2
Jain et al. [16]	62.5	Wang et al. (FV)[26]	91.2
Wang et al. [38]	59.9		
Wang et al. (BOW)[26]	62.2		
Wang et al. (FV)[26]	64.3		
Our	**64.1**	Our	**90.5**

64.3% with FV model. The improved dense trajectories coupled with FV model outperforms BOW-based techniques. The FV is based on fitting the Gaussian Mixture Model (GMM) to the features. The obtained representation records, for each Gaussian component, mean and variance statistics along each dimension. Thus, more information is stored per visual word. But, the major inconvenient of the FV encoding method is that it provides a very large histogram ($2 \times k \times d$, where d, where d is the descriptor dimension and k is the number of Gaussian components). Despite that, we have obtained a comparable results (64.1%).

The obtained result proves that tackling the relationships between trajectories can significantly enhance the action recognition performance and thus BOG can be a good alternative to the BOW model usually used in the action recognition methods. Besides, this can encourage the investigation in future works of the possible FV encoding scheme extension by considering the topological relationships between trajectories.

On UCF50 dataset, Kliper-Gross et al. [39] have reported 72.7% by designing descriptors that capture local changes in motion directions. Solmaz et al. [40] have reached 73.7% with a GIST3D video descriptor which is an extension of the GIST descriptor [41] to video. Reddy and Shah. [35] have given 76.9% by combining the MBH descriptor with scene context information. Shi et al. [42] have reported 83.3% using randomly sampled HOG, HOF, HOG3D and MBH descriptors. Wang et al. [38] have reached 85.6% using the dense trajectories and the BOW model coupled with the STP representation. Using the improved dense trajectories method, Wang et al. [26] have reached 87.2% with the standard BOW and 91.2% with the FV encoding. As shown in table 2, our approach (90.5%) outperforms the approaches proposed in [35,38–42] by a significant margin and gives close result to Wang et al. method [26] (91.2%) which is based on FV encoding.

6 Conclusion

In this work, we have presented a new video representation that exploits the structural information from features for human action recognition. Our approach models more higher order spatial relationships of the video trajectories by presenting them as graph descriptors using a multi-scales triangulation. We have applied the BOG encoding method in order to build a compact graph-based codebook by quantifying graph descriptors. Our experimentation on two challenging datasets shows that exploiting the relationships between trajectories is important to enhance the BOW models. In future works, we expect to further improve our approach by incorporating the spatial constraints between trajectories in the encoding process of the more sophisticated FV method.

References

1. Ben Aoun, N., Elghazel, H., Ben Amar, C.: Graph modeling based video event detection. In: IIT, pp. 114–117 (2011)
2. Bouchrika, T., Zaied, M., Jemai, O., Ben Amar, C.: Neural solutions to interact with computers by hand gesture recognition. MTA, 1–27 (2013)

3. Mejdoub, M., Fonteles, L., Ben Amar, C., Antonini, M.: Embedded lattices tree: An efficient indexing scheme for content based retrieval on image databases. JVCI **20**, 145–156 (2009)
4. Ben Aoun, N., Elghazel, H., Hacid, M.-S., Ben Amar, C.: Graph aggregation based image modeling and indexing for video annotation. In: Real, P., Diaz-Pernil, D., Molina-Abril, H., Berciano, A., Kropatsch, W. (eds.) CAIP 2011, Part II. LNCS, vol. 6855, pp. 324–331. Springer, Heidelberg (2011)
5. Laptev, I., lek, M.M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
6. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC (2010)
7. Klaser, A., Marsza lek, M., Schmid, C.: A spatio-temporal descriptor based on 3Dgradients. In: BMVC (2008)
8. Wang, H., Klaser, A., Schmid, C., Liu, C.L.: Action recognition by dense trajectories. In: CVPR (2011)
9. Uemura, H., Ishikawa, S., Mikolajczyk, K.: Feature tracking and motion compensation for action recognition. In: BMVC (2008)
10. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
11. Wang, F., Jiang, Y.G., Ngo, C.W.: Video event detection using motion relativity and visual relatedness. In: ACM MM (2008)
12. 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)
13. 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)
14. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatiotemporal context modeling for action recognition. In: CVPR (2009)
15. Jiang, Y.-G., Dai, Q., Xue, X., Liu, W., Ngo, C.-W.: Trajectory-based modeling of human actions with motion reference points. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part V. LNCS, vol. 7576, pp. 425–438. Springer, Heidelberg (2012)
16. Jain, M., Jou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR (2013)
17. Dammak, M., Mejdoub, M., Ben Amar, C.: Feature Vector Approximation Based on Wavelet Network. ICAART, 394–399 (2012)
18. Mejdoub, M., Ben Amar, C.: Classification improvement of local feature vectors over the KNN algorithm. Multimedia Tools and Applications, 197–218 (2013)
19. Sekma, M., Mejdoub, M., Ben Amar, C.: Human action recognition using temporal segmentation and accordion representation. In: Wilson, R., Hancock, E., Bors, A., Smith, W. (eds.) CAIP 2013, Part II. LNCS, vol. 8048, pp. 563–570. Springer, Heidelberg (2013)
20. Sivic, J., Zisserman, A.: Video google: a text retrieval approach to object matching in videos. In: ICCV, vol. 2, pp. 1470–1477 (2003)
21. Sekma, M., Mejdoub, M., Ben Amar, C. Spatio-temporal pyramidal accordion representation for human action recognition. In: ICASSP, pp. 1270–1274 (2014)
22. Mejdoub, M., Fonteles, L., Ben Amar, C., Antonini, M.: Fast indexing method for image retrieval using tree-structured lattices. In: CBMI, pp. 365–372 (2008)

23. Mejdoub, M., Fonteles, L., Ben Amar, C., Antonini, M.: Fast algorithm for image database indexing based on lattice. In: EUSIPCO, pp. 1799–1803 (2007)
24. Hashimoto, M., Cesar Jr., R.M.: Object Detection by Keygraph Classification. In: Torsello, A., Escolano, F., Brun, L. (eds.) GbRPR 2009. LNCS, vol. 5534, pp. 223–232. Springer, Heidelberg (2009)
25. Jouili, S., Mili, I., Tabbone, S.: Attributed graph matching using local descriptions. In: Blanc-Talon, J., Philips, W., Popescu, D., Scheunders, P. (eds.) ACIVS 2009. LNCS, vol. 5807, pp. 89–99. Springer, Heidelberg (2009)
26. Wang, H. Schmid, C.: Action recognition with improved trajectories. In: ICCV (2013)
27. Dalal, N., Triggs, B., Schmid, C.: 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)
28. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR, June 2005
29. Vig, E., Dorr, M., Cox, D.: Space-variant descriptor sampling for action recognition based on saliency and eye movements. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VII. LNCS, vol. 7578, pp. 84–97. Springer, Heidelberg (2012)
30. Mathe, S., Sminchisescu, C.: Dynamic eye movement datasets and learnt saliency models for visual action recognition. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part II. LNCS, vol. 7573, pp. 842–856. Springer, Heidelberg (2012)
31. Jgou, H., Douze, M., Schmid, C., Prez, P.: Aggregating local descriptors into a compact image representation. In: CVPR (2010)
32. Sanchez, J., Perronnin, F., Mensink, T., Verbeek, J.: Image classification with the Fisher vector: Theory and practice. IJCV, 222–245 (2013)
33. Mahboubi, A., Benois-P, J., Barba, D.: Joint tracking of polygonal and triangulated meshes of objects in moving sequences with time varying content. In: ICIP, vol. 2, pp. 403–406 (2001)
34. Marszalek, M., Laptev, I., Schmid, C.: Actions in context. In: CVPR (2009)
35. Reddy, K., Shah, M.: Recognizing 50 human action categories of web videos. In: MVA, pp 1–11 (2012)
36. B. Chakraborty, M.B. Holte, T.B. Moeslund, J. Gonzàlez, "Selective Spatio-Temporal interest points", In CVIU, pp 396–410, 2012
37. Cho, J., Lee, M., Chang, H., Oh, S.: Robust action recognition using local motion and group sparsity. Pattern Recognition, pp 1813–1825 (2014)
38. Wang, H., Klaser, A., Schmid, C., Liu, C.-L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV, 60–79 (2013)
39. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012, Part VI. LNCS, vol. 7577, pp. 256–269. Springer, Heidelberg (2012)
40. Solmaz, B., Assari, S.M., Shah, M.: Classifying web videos using a global video descriptor. MVA, 1–13 (2012)
41. Oliva, A., Torralba, A.: Modeling the shape of the scene: A holistic representation of the spatial envelope. IJCV **42**(3), 144–175 (2001)
42. Shi, F., Petriu, E., Laganiere, R.: Sampling strategies for real-time action recognition. In: CVPR (2013)