Visual Code-Sentences: A New Video Representation Based on Image Descriptor Sequences

Yusuke Mitarai and Masakazu Matsugu

Canon Inc. Digital System Technology Development Headquarters, Tokyo, Japan

Abstract. We present a new descriptor-sequence model for action recognition that enhances discriminative power in the spatio-temporal context, while maintaining robustness against background clutter as well as variability in inter-/intra-person behavior. We extend the framework of Dense Trajectories based activity recognition (Wang *et al.*, 2011) and introduce a pool of dynamic Bayesian networks (e.g., multiple HMMs) with histogram descriptors as codebooks of composite action categories represented at respective key points. The entire codebooks bound with spatio-temporal interest points constitute intermediate feature representation as basis for generic action categories. This representation scheme is intended to serve as *visual code-sentences* which subsume a rich vocabulary of basis action categories. Through extensive experiments using KTH, UCF Sports, and Hollywood2 datasets, we demonstrate some improvements over the state-of-the-art methods.

1 Introduction

We have seen great improvements in the domain of action recognition in videos over the past few decades, especially in modeling of as well as feature representation for action categories [1]. In regard to local features, the most notable advancement is the proposal of spatio-temporal interest points by Laptev and Lindeberg (2003) [9], which provides a substrate of stable representations of actions, and the original local feature and its variants are now widely used by researchers in the field of action recognition.

Stable representation of action categories in cluttered scenes is still a challenging problem that needs to be solved with a representation framework rich discriminative power. For example, we need the ability to distinguish similar categories like running and jogging, while we also need to neglect individualities observed as personal differences that are typically measured by speed and appearance (e.g., body shape, clothing, and personal belongings). Background clutter and view-point diversity further challenge action recognition.

Recently, methods using trajectories, extracted based on spatio-temporal interest points and tracking scheme, have been very successful in recognizing actions [5, 14, 15, 23, 25]. Despite their success, they have difficulty in discriminating spatio-temporal contexts, albeit maintaining stability and robustness in recognition.

A. Fusiello et al. (Eds.): ECCV 2012 Ws/Demos, Part I, LNCS 7583, pp. 321–331, 2012.

In this paper, we address these problems with a trajectory-based approach. The main contributions of this paper are twofold: 1) Introduction of a pool of dynamic Bayesian networks or DBNs (e.g., multiple HMMs) bound with positional information in respective trajectories. Each HMM is organized to provide an intermediate representation of basis action primitives as a *code-sentence*, a set of *code words* with temporal dependency. This repository of DBN enhances discriminative power in the spatio-temporal context since each DBN can capture the spatio-temporal ordering of composite action primitives. 2) Introduction of histogram-based description of trajectory-bound intermediate features that inherit robustness and stability of BoW-like representation.

Thus, in the proposed framework of Dense Trajectories based action recognition, we seek balance between discriminability in spatio-temporal dependencies and stability against intra and inter-person behavioral variations as well as background clutter.

2 Related Work

Local spatio-temporal words/features have been exploited to recognize actions ([6, 8, 9, 11, 18, 22]). Models of human actions based on key point descriptors have been shown to perform well in action recognition from videos.

Modeling efforts in action recognition have a long history. Bag-of-words models devoid of spatio-temporal ordering information have also been exploited in action recognition [4, 10, 13]. Because of independence on spatio-temporal relationships, BoW-based approaches are limited in their ability to represent and differentiate such dependencies. Several models attempt to alleviate this limitation; new types of features capture spatio-temporal correlation [21], modeling spatio-temporal relationships by coarse spatio-temporal grid regions.

One of the standard approaches for recognizing human actions uses dynamic Bayesian networks [19, 28]. The simplest form of this approach is HMM. For modeling complex behaviors, several extensions of HMM have been advocated: coupled hidden semi Markov models [16], hierarchical HMM [12, 17], and hidden CRF [29]. Hierarchical approaches have been taken in modeling complex activities: probabilistic topic models [26], hierarchical spatio-temporal context in trajectories [23], and hierarchical HMM ([12, 17]).

Trajectory-based approaches in human action recognition have recently attracted attention in research communities and demonstrate the state-of-the-art method [25] for the challenging datasets UCF Sports, Hollywood2, and YouTube.

For modeling the dynamic structure of trajectory-aligned features, a few approaches using like a dynamic Bayesian networks as the models of trajectory-aligned features were proposed: modeling velocity histories of tracked key points [15], and trajectory transition descriptor based on a Markov stationary distribution of quantized displacement vectors [23].

3 Visual Code-Sentences

We model arbitrary actions by a set of hypothetical action primitives as *visual code-sentences* in the sense that *sentences* correspond to actions, while the *visual* *code-sentences* constitute visual code-words necessary to represent meaningful action categories. Here, *sentences* are defined as descriptor-sequences along the trajectories extracted from video, and each *sentence* is hypothetically generated from a certain component model, *code-sentence,* capturing the temporal order of state transitions. It is well known that such a model is generally given by dynamic Bayesian networks (DBN). We note that each *sentence* is position-bound (e.g., bound with key points along specific trajectory) and a generative model of a *sentence* is represented as a mixture of *code-sentences* that capture a dynamic structure of *sentences*. Entire *code-sentences* are pooled in a repository so we can generate arbitrary *sentences*.

We will show that, in the Dense Trajectories based approach, a histogram based description of the trajectory bound component models can also be used for the stability and robustness of action recognition. Changes in the dynamic structure of spatio-temporal ordering as categorical changes in actions are assumed to be distinguishable based on histogram representation, while suppressing inter- or intra-person variations.

3.1 Summary of Our Representation and Classification System

To extract the *code-sentence* representation, we need to extract a number of *sentences* from video by aligning descriptors at each key point along trajectories [25]. The component model which presumably generated the *sentence* is determined based on the likelihood of the *sentence* corresponding to each component model. Each component is modeled by a generative model of DBN that represents dynamical properties of *code-sentence*. We use HMM a simple DBN. A pool of HMMs is learned with a video dataset (see 3.3). Finally, a BoW-like representation is constructed based on the histogram of each component model which possibly generated the *sentence*.

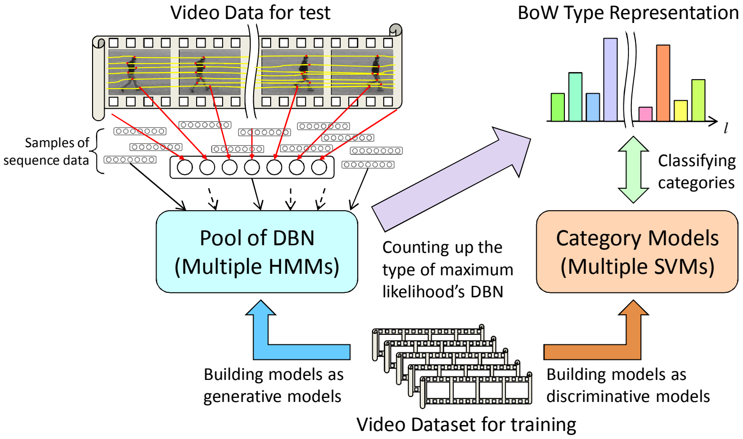
Video Data for test
BoW Type Representation
l
Samples of sequence data
Classifying categories
Pool of DBN (Multiple HMMs)
Counting up the type of maximum likelihood's DBN
Category Models (Multiple SVMs)
Building models as generative models
Building models as discriminative models
Video Dataset for training

Fig. 1. Illustration of our representation and classification system summary

In the classification stage, we similarly extract *visual code-sentences* and the associated histogram description from the Dense Trajectories of the input data. The resulting BoW-like representation from the input video is finally classified using multiple SVMs.

3.2 Model Definition of Code-Sentences

The *visual code-sentences* (VCS) as generative models of spatio-temporal sequence shall be defined so as to serve as basis features of actions in the sense that any action categories can be represented by such composite, intermediate-level description of actions. For a given L length trajectory extracted from video, the continuous-valued descriptor (e.g. HOG, MBH, etc. as shown in subsection 4.1) sequence along the trajectory, $\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L$, and $X = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_L\}$ is assumed to be generated from latent states $\{z_t\}$, so that the sequence can be described by continuous HMMs. Using a parameter vector, $\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\varphi}\}$, the probability of observing the sequence $\boldsymbol{X}=\{x_t\}$, is given by:

$$p(\boldsymbol{X}|\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\varphi}) = \sum_{z_1} p(z_1|\boldsymbol{\pi}) p(\boldsymbol{x}_1|z_1, \boldsymbol{\varphi}) \prod_{t=2}^{L} \sum_{z_t, z_{t-1}} p(z_t|z_{t-1}, \boldsymbol{A}) p(\boldsymbol{x}_t|z_t, \boldsymbol{\varphi}). \tag{1}$$

$\boldsymbol{\theta} = \{\boldsymbol{\pi}, \boldsymbol{A}, \boldsymbol{\varphi}\}$ is a set of parameters of probability functions, $p(z_1|\boldsymbol{\pi})$, $p(z_t|z_{t-1}, \boldsymbol{A})$ and $p(\boldsymbol{x}_t|z_t, \varphi)$. Let $\boldsymbol{\Theta} = \{\boldsymbol{\theta}_1, \boldsymbol{\theta}_2, \ldots, \boldsymbol{\theta}_M\}$ be a set of M parameter vectors of each HMM. Then, using these parameters, we consider a *sentence*, indexed by $l(\boldsymbol{X})$, for a given video sequence is represented by a VCS that gives the highest probability:

$$l(\boldsymbol{X}) = \underset{m}{\operatorname{argmax}}\ p(\boldsymbol{X}|\boldsymbol{\theta}_{\boldsymbol{m}}). \tag{2}$$

In the proposed framework, a VCS is given by a set of indices, $l(\boldsymbol{X})$, that represents a primitive action category in a vector quantized state space. Thus we quantize each sequence data not based on the Markov stationary distribution [23] but based on the above generative models (HMMs) which are directly modeling sequence data.

3.3 Learning Visual Code-Sentences Method

Let $\{\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_N\}$ be a set of N sequences extracted from various sets of video data. We propose to learn VCSs by generating a pool of HMMs for sequences of data. In contrast to [27], the procedure for obtaining VCS begins by random initialization of all labels, $\{l(\boldsymbol{X})\}$, and all parameter vectors, $\boldsymbol{\Theta}$. We then update the parameter of each HMM based on sequences of data to obtain the approximate estimate of the following parameter (3)

$$\boldsymbol{\theta}_{\boldsymbol{m}}^{\boldsymbol{new}} = \underset{\boldsymbol{\theta}}{\operatorname{argmax}} \prod_{l(\boldsymbol{X})=\mathrm{m}} p(\boldsymbol{X}|\boldsymbol{\theta}_{\boldsymbol{m}}) \tag{3}$$

$p(\boldsymbol{X}|\boldsymbol{\theta}_{\boldsymbol{m}})$ is defined as in Eq. (1). This step is intended to obtain approximate cluster centers of the sequences of data like a cluster center calculation step of the k-means clustering algorithm. Next, the labels of respective sequence data are updated in a manner (2). This step is assumed to be an assignment to cluster step of the k-means. After all labels are updated, the parameter of each HMM is updated similarly to approximate (3). These two steps are repeatedly performed until the update step converges. This approach is similar to modeling a set of varied sequence data by HMM mixture models. We do not exploit ordinary EM algorithms to obtain HMM mixture models. Instead, we generate a set of parameters of HMMs to explore diversity and completeness in the resulting models and avoid obtaining only similar models.

VCSs as multiple HMMs are learned with a plurality of sequence data from training video sequence, yielding 2,000 HMMs with 480,000 sequences sampled randomly from video for each descriptor type. These 2,000 HMMs include three types of HMMs: 1) Ergodic HMMs with two latent states for cyclic action primitives, 2) Left-to-Right HMMs with four latent states for action primitives corresponding to slow motions and 3) Left-to-Right HMMs with six latent states for action primitives corresponding to fast motions. The last type is permitted to skip one latent state. We obtained 400 Ergodic HMMs and 800 x 2 Left-to-Right HMMs, respectively.

The learning phase typically converges after by repeating the assignment and update step about 100 times. When too many pieces of sentence data are assigned to a particular component label in the learning phase, we divided such agglomerated data to obtain 'hard-assignments' to different class labels. The resulting VCSs as repository of multiple HMMs are used for video representations.

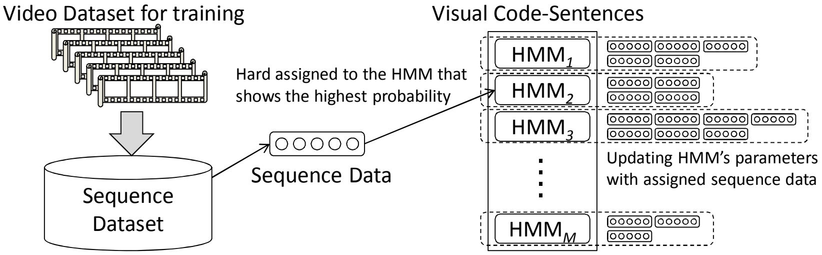
Video Dataset for training
Sequence Dataset
Sequence Data
Hard assigned to the HMM that shows the highest probability
Visual Code-Sentences
HMM$_1$
HMM$_2$
HMM$_3$
HMM$_M$
Updating HMM's parameters with assigned sequence data

Fig. 2. Learning system for VCSs. The Sequence Dataset is constructed from various sets of video data. (Assign Step): Generating probability for each HMM is calculated, and each sequence is assigned to the most probable HMM. (Update Step): Each HMM's parameter is updated independently by sequence data assigned to each HMM. The 'Assign' and 'Update' steps are performed repeatedly until the HMM parameters converge.

4 Experimental Setup

In this section, we describe the experimental setup, which uses visual code-sentences to evaluate the performance of our video representation method.

4.1 Extracting Sentence Data from Video

We use the trajectory base feature extraction method (Dense Trajectories recently proposed by Wang *et al.* [25]) to extract sentences from video. Trajectories are extracted by tracking points located on grid points till each length becomes L (we use a length $L = 15$) with a dense optical flow field in multiple spatial scales. In their original setup, the trajectories are removed if tracked points exist in the neighborhood of the start points of the trajectories. In our framework, we do not remove these trajectories to avoid heterogeneous sampling.

Descriptions on tracked points of the trajectories which constitute the sentences are the same descriptors as the ones in Dense Trajectories, i.e., motion descriptors for tracked points [25], HOG [2], HOF [10] and MBH [3]. We also use the same descriptors directly as in [25], but do not perform temporal integration and concatenation as in [25]. We treat them as sequence data so that the temporal context is retained and discriminated in the classification stage.

4.2 Video Representation with Visual Code-Sentences

We use a bag-of-features type of video representation with visual code-sentences. The representation data is a histogram with bins of each feature type's HMMs, such as motion descriptors for tracked points, HOG, HOF, X-direction MBH, and Y-direction MBH. There are 2,000 HMMs for each feature-type, so the representation data is obtained by concatenating five histograms each of which consists of 2,000 bins. Sequences of data are extracted from the video, and the generating probabilities of each sequence data are computed with each HMM of each type. We assume that the obtained sequence data corresponds to the most probable HMM.

We use a non-linear SVM with a χ^2-kernel [10] for predicting action category. A χ^2-kernel $K(\boldsymbol{h}_i, \boldsymbol{h}_j)$ is described as follows:

$$K(\boldsymbol{h}_i, \boldsymbol{h}_j) = exp\left[-\sum_{\gamma} \frac{1}{\rho A^{\gamma}} \sum_{bin} \frac{\left\{h_i^{\gamma}(bin) - h_j^{\gamma}(bin)\right\}^2}{h_i^{\gamma}(bin) + h_j^{\gamma}(bin)}\right] \tag{4}$$

$h_i^{\gamma}(bin)$ is the *bin*-th element of histogram for feature type γ. A^{γ} is the mean value of χ^2 distances of the training data about feature type γ [30], and ρ is a parameter for adjusting kernel width.

We perform the estimate of the action categories based on the following category score $S_c(\boldsymbol{h})$ corresponding to category c:

$$S_c(\boldsymbol{h}) = \sum_{\boldsymbol{h}_{i,c} \in SV_c} \alpha_{i,c} K(\boldsymbol{h}, \boldsymbol{h}_{i,c}) + \beta_{SVM_c} + b_c \tag{5}$$

where $h_{i,\,c}$ is i-th support vector corresponding to category c, and $\alpha_{i,\,c}$ is the coefficient for the i-th support vector. β_{SVMc} is a bias parameter obtained by learning SVM for category c, and b_c is a second bias parameter particular to category c to adjust unevenness of the number of training data. In classification task, the test data is classified to the category which gave the highest category score. In the retrieval task, the result lists categories in descending order of the scores.

4.3 Datasets

Video representation with visual code-sentences is evaluated on three standard human action datasets: KTH [22], UCF Sports [20] and Hollywood2 [13] shown in Figure 3. We describe each dataset in this section.

KTH Dataset. The learning process is performed with the training dataset, and hyper parameters (i.e., soft margin parameter of SVMs and parameter to adjust kernel width) are optimized with the validation dataset. Based on the original experimental setup, we evaluate average accuracy over all categories.

UCF Sports Dataset. We evaluate average accuracy over all categories by a leave-one-out setup. In training we use horizontal flipped data. Because the amount of training data per category is uneven, we optimize the second bias parameter. Thus, soft margin, kernel width, and second bias parameter are optimized.

Hollywood2 Dataset. The dataset is treated for retrieval task, and we evaluate the performance by the mean average precision. In this case, two hyper parameters are optimized, as in the case the KTH Dataset.

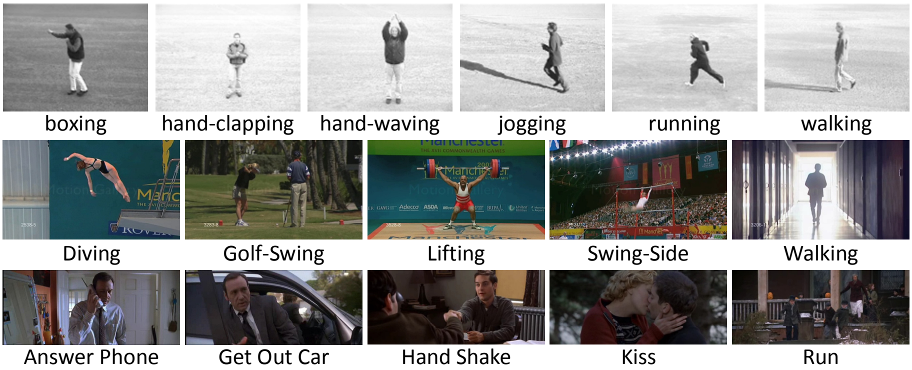

Fig. 3. Some example frames from video pulled from KTH (*first row*), UCF Sports (*second row*) and Hollywood2 (*last row*) datasets

5 Evaluation Results

In this section, we report evaluation results of the datasets, and compare our method with the state-of-the-art methods shown in Table 1.

We note that, for KTH, our method gives a slightly worse result than the state-of-the-art methods. Most failures came from the confusion between similar categories like “running” and “jogging”, probably due to robustness of VCS representation for variation in motion speed. We demonstrated improved performance over the state-of-the-art method in Gaidon *et al.* [5] on UCF Sports with parameter tuning to each category.

In the case of Hollywood2, our method yielded the best result of 58.3%. We obtained average accuracy of 93.75%, 93.28% and 93.17% for the size of VCS 1000, 500, and 250 respectively on KTH dataset. Thus, for the larger size of VCS, recognition

Table 1. Recognition performance

KTH		UCF Sports		Hollywood2	
Laptev *et al.* [10]	91.8%	Kläser *et al.* [7]	86.7%	Gilbert *et al.* [6]	50.9%
Kovashka *et al.* [8]	94.53%	Wang *et al.* [25]	88.2%	Ullah *et al.* [24]	55.3%
Wang *et al.* [25]	**95.0%**	Gaidon *et al.* [5]	90.3%	Wang *et al.* [25]	**58.3%**
Our method	93.98%	Our method	**91.1%**	Our method	**58.3%**

performance tended to be slightly higher. We also compare average precision pre action categories for Hollywood2 dataset shown in Table 2. Our method achieved the best results for 5 out of 12 action categories.

Table 2. Average Precision pre action categories for Hollywood2 dataset

	Our Method	Wang *et al.* [25]	Ullah *et al.* [24]
AnswerPhone	27.9%	**32.6%**	24.8%
DriveCar	**92.7%**	88.0%	88.1%
Eat	**66.2%**	65.2%	61.4%
FightPerson	80.9%	**81.4%**	76.5%
GetOutCar	44.9%	**52.7%**	47.4%
HandShake	33.5%	29.6%	**38.4%**
HugPerson	49.9%	**54.2%**	44.6%
Kiss	63.7%	**65.8%**	61.5%
Run	**84.9%**	82.1%	74.3%
SitDown	**66.0%**	62.5%	61.3%
SitUp	20.1%	20.0%	**25.5%**
StandUp	**69.0%**	65.2%	60.4%
mAP	**58.3%**	**58.3%**	55.3%

To gain a more concrete view on the functionality of the proposed VCS based representation, we investigate the contribution of specific VCS in the KTH dataset. Some of the VCSs act as specific components to represent a class of particular action categories. The specific VCSs are shared to represent similar categories. Some VCSs are specific to a particular category, while other VCSs are specific to discriminate small differences among similar categories. For example, "hand-waving" includes a composite action of "lifting-up right hand", and we actually found a corresponding type of motion descriptor sequence as VCS, shown in Figure 4 (a). Figure 4 (b) represents a sequence for upper body motion that does not include information on swinging arm motion (this can be used to discriminate "running" from "walking"), which is given by MBH based descriptor sequence modeled as a VCS. We also found some shared basis of action categories for foot movement commonly used in representing "running" and "walking" in MBH-based VCSs (Figure 4 (c)).

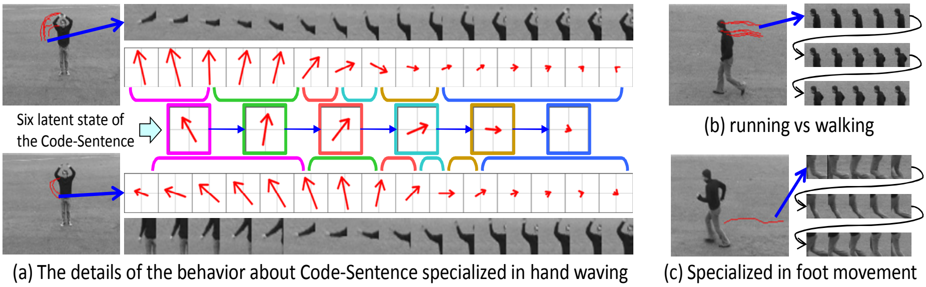
Six latent state of the Code-Sentence
(a) The details of the behavior about Code-Sentence specialized in hand waving
(b) running vs walking
(c) Specialized in foot movement

Fig. 4. Example of some specific VCSs and trajectories corresponding to each VCS

Table 3 gives 'Assignment Rate' that represents the probability of VCS found in a sentence extracted from a video of a certain category.

Table 3. 'Assignment Rate' that show exemplary VCSs found in specific action datasets

category VCS type	boxing	hand clapping	hand waving	jogging	running	walking
(a) hand-waving	7.7E-5	2.2E-5	**6.7E-3**	1.3E-6	5.0E-6	1.4E-5
(b) running vs walking	7.9E-4	5.6E-4	2.3E-4	1.2E-3	3.9E-4	**4.4E-3**
(c) foot movement	3.6E-4	1.6E-4	1.3E-4	**2.7E-3**	**2.8E-3**	**1.7E-3**

6 Conclusion

We proposed a method of video representation with *visual code-sentences* and demonstrated its validity through extensive experiments with several challenging datasets. We achieved competitive performance using a code book half the size of the method by proposed Wang *et al.* [25] with approximately 3% improvement in performance for a challenging dataset (e.g., UCF Sports). We also validated the proposed framework with supporting evidences to show that VCS can be shared and used as a basis for representing a variety of action categories.

The proposed method can be poor in distinguishing categories that differ only by motion speed (e.g., distinguishing "jogging" from "running"), however it is effective in identifying categories with inter-/intra- person variations in motion speed. In the experiments, we did not optimize descriptor parameters suitable for VCSs, and further performance improvement through parameter optimization is left for future work.

References

1. Aggarwal, J.K., Ryoo, M.S.: Human Activity Analysis: A Review. ACM Computing Surveys 43(16) (2011)
2. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. In: CVPR (2005)

3. 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)
4. Dollár, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior Recognition via Sparse Spatio-Temporal Features. In: VS-PETS (2005)
5. Gaidon, A., Harchaoui, Z., Schmid, C.: A time series kernel for action recognition. In: BMVC (2011)
6. Gilbert, A., Illingworth, J., Bowden, R.: Action Recognition using Mined Hierarchical Compound Features. TPAMI 33(5) (2009)
7. Kläser, A., Marszałek, M., Laptev, I., Schmid, C.: Will person detection help bag-of-features action recognition. Technical Report, INRIA Grenoble - Rhone-Alpes (2010)
8. Kovashshka, A., Grauman, K.: Learning a Hierarchical of Discriminative Space-Time Neighborhood Features for Human Action Recognition. In: CVPR (2010)
9. Laptev, I., Lindeberg, T.: Space-time Interest Points. In: ICCV (2003)
10. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR (2008)
11. Liu, J., Yang, Y., Shah, M.: Learning Semantic Visual Vocabularies Using Diffusion Distance. In: CVPR (2009)
12. Loy, C.C., Xiang, T., Gong, S.: Detecting and Discriminating Behavioural Anomalies. Pattern Recognition 44 (2011)
13. Marszałek, M., Laptev, I., Schmid, C.: Actions in Context. In: CVPR (2009)
14. Matikainen, P., Hebert, M., Sukthankar, R.: Trajectons: Action Recognition Through the Motion Analysis of Tracked Features. In: ICCV Workshop on Video-Oriented Object and Event Classification (2009)
15. Messing, R., Pal, C., Kautz, H.: Activity recognition using the velocity histories of tracked keypoints. In: ICCV (2009)
16. Natarajan, P., Nevatia, R.: Coupled Hidden Semi Markov Models for Activity Recognition. In: WMVC (2007)
17. Nguyen, N.T., Phung, D.Q., Venkatesch, S., Bui, H.H.: Learning and Detecting Activities from Movements Trajectories Using Hierarchical Hidden Markov Model. In: CVPR (2005)
18. Niebles, J.C., Wang, H., Fei-Fei, L.: Unsupervised Learning of Human Action Categories Using Spatial-temporal Words. In: BMVC (2006)
19. Park, S., Aggarwal, J.K.: A hierarchical Bayesian network for event recognition of human actions and interactions. Multimedia Systems 10(2) (2004)
20. Rodriguez, M., Ahmed, J., Shah, M.: Action MACH: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition. In: CVPR (2008)
21. Savarese, A., Pozo, A.D., Niebles, J.C., Fei-Fei, L.: Spatial-temporal correlations for unsupervised action classification. In: Motion and Video Computing (2008)
22. Schüldt, C., Laptev, I., Caputo, B.: Recognizing Human Actions: A Local SVM Approach. In: ICPR (2004)
23. 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)
24. Ullah, M.M., Parizi, S.N., Laptev, I.: Improving Bag-of-Features Action Recognition with Non-local Cues. In: BMVC (2010)
25. Wang, H., Kläser, A., Schmid, C., Liu, C.: Action Recognition by Dense Trajectories. In: CVPR (2011)
26. Wang, X., Ma, X., Grimson, W.E.L.: Unsupervised Activity Perception in Crowded and Complicated Scenes Using Hierarchical Bayesian Models. TPAMI 31(3) (2009)
27. Xiang, T., Gong, S.: Video Behaviour Profiling for Anomaly Detection. TPAMI 30(5) (2008)

28. Zeng, Z., Ji, Q.: Knowledge Based Activity Recognition with Dynamic Bayesian Network. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part VI. LNCS, vol. 6316, pp. 532–546. Springer, Heidelberg (2010)
29. Zhang, J., Gong, S.: Action categorization with modified hidden conditional random field. Pattern Recognition 42(1) (2010)
30. Zhang, J., Marszałek, M., Lazebnik, S., Schmid, C.: Local Features and Kernels for Classification of Texture and Object Categories: A Comprehensive Study. IJCV 73(2) (2007)