# Motion of oriented magnitudes patterns for Human Action Recognition

Hai-Hong Phan, Ngoc-Son Vu, Vu-Lam Nguyen, Mathias Quoy

# Motion of oriented magnitudes patterns for Human Action Recognition

Hai-Hong Phan  Ngoc-Son Vu  Vu-Lam Nguyen  Mathias Quoy

ETIS - ENSEA/Universite de Cergy-Pontoise
CNRS UMR 8051, 95000-Cergy, France
{thi-hai-hong.phan,son.vu,lam.nguyen,mathias.quoy}@ensea.fr

**Abstract.** In this paper, we present a novel descriptor for human action recognition, called Motion of Oriented Magnitudes Patterns (MOMP), which considers the relationships between the local gradient distributions of neighboring patches coming from successive frames in video. The proposed descriptor also characterizes the information changing across different orientations, is therefore very discriminative and robust. The major advantages of MOMP are its very fast computation time and simple implementation. Subsequently, our features are combined with an effective coding scheme VLAD (Vector of locally aggregated descriptors) in the feature representation step, and a SVM (Support Vector Machine) classifier in order to better represent and classify the actions. By experimenting on several common benchmarks, we obtain the state-of-the-art results on the KTH dataset as well as the performance comparable to the literature on the UCF Sport dataset.

## 1 Introduction

In the recent years, human action recognition (HAR) has become one of the most popular topics in the computer vision domain due to its variety of applications, such as human-computer interaction, human activities analysis, surveillance systems, and so on. The goal of HAR is to identify the actions in a video sequence with different challenges such as cluttering, occlusion and change of lighting conditions.

More recently, a new approach based on deep learning model, especially Convolutional Neural Networks (ConvNets) architecture [1][2][3][4] has achieved great success. The architecture can learn a hierarchy of features by building high-level features from low-level ones, thereby automating the process of feature construction. Very recently, Tran *et al* [4] proposed spatiotemporal Convolutional 3D (C3D) learning features using deep 3-dimensional convolutional networks (3D ConvNets). The features achieved outstanding performance on benchmarks such as Sport1M, UCF101 and ASLAN. However, those ConvNets-based systems require a large data set and high costly computation. Therefore, until now, the approach based on hand-crafted features [5],[6],[7],[8],[9],[10] still occupy its important position in computer vision due to its comprehensibility and efficiency.

Histogram of Oriented Gradients (HOG) and Histogram of Optical Flow (HOF) were successfully used for action recognition [6]. To characterize local motion and appearance, the authors compute histograms of spatial gradient and optical flow accumulated in space-time neighborhoods of detected interest points. Klaser *et al* [11] proposed the HOG3D descriptor as an extension of the popular SIFT descriptor [12] to video sequences. These descriptors represented both shape and motion information of actions in videos. Space Time Interest Points (STIPs) [5] extracted HOG and HOF at each interesting point calculated by 3D-Harris detector. The best performance in trajectory-based pipeline was held by Motion Boundary Histogram (MBH) [13], which horizontal and vertical components of optical flow were separately computed.

Recently, Wolf *et al* in [14] proposed the Local Trinary Patterns (LTP) for action recognition. This descriptor combined the effective description properties of Local Binary Patterns (LBP) with the appearance invariance and adaptability of patch matching based methods. Also, Kliper-Gross *et al* in [15] analyze the relationship of consecutive frames by considering at each pixel over the video the changing between different frames. In those methods, for each pixel, the gray value is used directly to determine the relationships between the frames.

Different from above approaches, in this paper, we propose a novel descriptor called Motion of Oriented Magnitudes Patterns (MOMP). This descriptor considers the relationship between the local gradient distributions in neighboring patches coming from successive frames in video and characterizes the information changing across different orientations. The major advantages of MOMP are its fast computation time and simple implementation. We also associate the extracted features to VLAD (Vector of Locally Aggregated Descriptors) and SVM classifier in order to better represent and classify the actions. The VLAD coding scheme [16] has tremendous successes in large scale image retrieval due to its efficiency of compact representation. This encoding perspective increases the amount of information without increasing the visual vocabulary size, therefore does not accelerate the clustering speed or reduces memory. Experiment results on two datasets prove the efficiency of our system.

The remainder of this paper is organized as follows. Section 2 concerns to the proposed method. Section 3 presents experimental results, and conclusions are given in Section 4.

## 2 Proposed Method

This section presents in detail our Motion of Oriented Magnitudes Patterns (MOMP) descriptor. Its construction is inspired by gradient-based features and self-similarity technique. In experiments, our descriptor shows fast computation time and simple implementation.

### 2.1 MOMP - Motion of oriented magnitudes patterns descriptor

The key idea of our descriptor is to characterize actions by the relationship between the local gradient distributions of neighboring patches coming from

consecutive frames in a video. The proposed algorithm can be considered as an extension of our previous work, Patterns of Oriented Edge Magnitudes (POEM), which is very successfully used for face recognition [17],[18]. In this work, we encode the motion changing across different orientations of different frames. To extract the features we carry out three steps: (1) the gradient of each frame is computed and quantized; (2) for each pixel, the magnitudes of its neighbors are accumulated and assigned to it; (3) the features are encoded based on the sum of squared differences (SSD) of gradient magnitudes of the triplet of frames.

**(1) Gradient computation and orientation quantization:**

In this step, we compute gradient and orientation quantization of each frame in the video using Haar features. As result of this step, each pixel in the video is represented by two elements: (1) gradient magnitude determining how quickly the image changes over the considered pixel, and (2) gradient orientation determining the direction of this changing. Consider a frame $F$, let $\varphi(p)$ and $m(p)$ be the orientation and magnitude of the image gradient at pixel $p$ within $F$. The gradient orientation of each pixel is evenly discretized over $0 - \pi$ (unsigned) or $0 - 2 \times \pi$ (signed). To reduce the loss in quantization stage, we apply soft assignment technique. Therefore, a pixel feature is encoded as a $d$-dimensional vector with only at most two non-null elements (each pixel falls into at most two nearest bins regarding its gradient orientation):

$$m(p) = [m_1(p), m_2(p), ..., m_d(p)] \tag{1}$$

where $d$ is the number of discretized orientations.

**(2) Magnitude accumulation over local patches:**

The second step is to incorporate gradient information from neighboring pixels by computing a local histogram of gradient orientations over all cell pixels. Vote weights can either be the gradient magnitude itself, or some function of the magnitude. More precisely, we individually compute the convolution of the magnitude map $m$ (result of step 1) and a Gaussian mask $G$ on each orientation:

$$G(x, y) = \frac{1}{(2\pi\sigma^2)e^{-(x^2+y^2)/2\sigma^2}} \tag{2}$$

where $\sigma$ is standard deviation. At pixel $p$, the feature is now represented by a $d$-dimension vector $v(p)$:

$$v(p) = [v_1(p), v_2(p), ..., v_d(p)] \tag{3}$$

where

$$v_i(p) = \sum_{p_j \in C} g_j * m_i(p_j) \tag{4}$$

with $C$ is a cell centered on $p$, $g_j$ is the $j$-th element of Gaussian filters. It is clearly seen that $v(p)$ conveys the oriented and magnitude information of not only the center pixel $p$ but also its neighbors. In this way, we incorporate the

richer information to a pixel.

**(3) Encoding:**

At the final step of feature extraction, the features obtained at the second step are encoded using the LTP-based self-similarity within more extended image regions, called blocks, coming from previous, current and next frames, as illustrated in Figure 1.

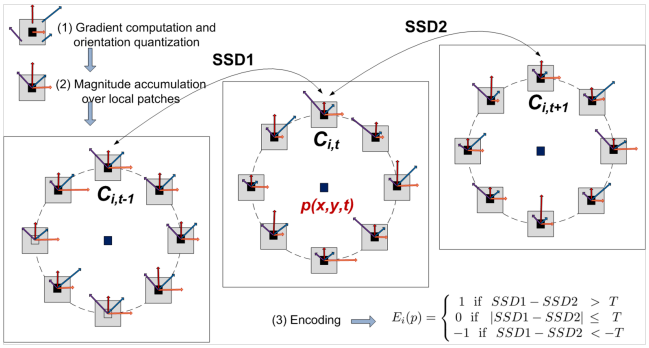


**Fig. 1.** The illustration of MOMP feature extraction

$E_i(p)$ is calculated for all pixels of the frame. We use a threshold of $T$,

$$E_i(p) = \begin{cases} 1 & \text{if } SSD1 - SSD2 > T \\ 0 & \text{if } |SSD1 - SSD2| \leq T \\ -1 & \text{if } SSD1 - SSD2 < -T \end{cases} \tag{5}$$

where $SSD1$ and $SSD2$ are calculated as following, with $d$ is the number of discretized orientations:

$$SSD1 = \sum_{j=1}^{d} \left[ v_j(p)_{p \in C_{i,t-1}} - v_j(p)_{p \in C_{i,t}} \right]^2 \tag{6}$$

$$SSD2 = \sum_{j=1}^{d} \left[ v_j(p)_{p \in C_{i,t+1}} - v_j(p)_{p \in C_{i,t}} \right]^2 \tag{7}$$

Considering $n$ neighbor cells surrounding the pixel $p(x, y, t)$ within the block in the current frame $t$, we obtain a $n$-trit string $E_i(p)$ ($i$ varies from 1 to $n$) denoted by $E(p)$. We divided the entire frame into $w \times h$ patches of equal size where the histograms of the $n$-digit trinary strings are computed with respect to the positive part (equal 1) and the negative part (equal -1). The histograms of these two parts are then concatenated to generate a $2^{n+1}$-bin vector. Therefore,

the length of MOMP descriptor, namely $D$, is $2^{n+1}$.

We analyze here the differences between the MOMP and LTP [14]:

- Both LTP and MOMP use three bits. However, LTP compares SSD of a cell at the current frame with its neighboring ones (at other positions) in the past or the future frames; while MOMP computes SSD between three cells at the same position in three successive frames (Figure 1).
- LTP calculates SSD based on gray intensity, wheras gradient-based values are used in MOMP. In this way, the information characterized is more robust to illumination change.
- MOMP encodes the information over different orientations, therefore conveys richer information about the video sequence.

### 2.2 Feature representation - Vector of locally aggregated descriptors (VLAD)

VLAD [16] shows great popularity in action recognition from video data due to its simplicity and good performance. It models video as collections of local spatio-temporal patches. A spatio-temporal patch is represented by a feature vector. VLAD employs only the nearest neighbor visual word in dictionary to aggregate each descriptor feature. In this paper, we also apply the model for feature representation thanks to its advantages:

- When applying VLAD, we can use a the small number of visual words $k$. For instance, for human action recognition, many work obtained the good results with the value of $k$ ranging from k=16 to k=256 ([16][19][20]). This means that, for big datasets, the computation cost is much lower than standard bag of words technique with $k$ increasing up to thousands.
- VLAD can be considered a simplified version of the Fisher Vector [21] and it is computationally more efficient. For our method, we utilize $L2$-normalize over feature vectors represented by VLAD.

### 2.3 Classification - Support Vector Machines (SVM) classifier

After the feature representation step, each video is described by a $k \times D$-dimension feature where $k$, $D$ respectively are number of visual words in VLAD and the length of each descriptor vector. These vectors are then used as input of a SVM classifier which is widely used for action recognition. We use the SVM classifier with $RBF$-kernel and the publicly available LIBSVM library [22] (the parameters will be detailed in Section 4.1).

## 3 Experiment results

### 3.1 Experimental Setup

Experiments were conducted on two datasets: KTH [23] and UCF Sport [24]. The KTH dataset includes 599 video sequences for six action classes: boxing,

hand clapping, hand waving, jogging, running and walking. The videos are performed in four different scenarios (indoors, outdoors, outdoors with scale change and outdoors with different clothes) with slight camera motion and a simple background. Figure 2(a) illustrates the example frames from KTH dataset. We follow to the protocol of [23]: the actions of 16 persons are used as training and the actions of 9 remaining people are used for testing. We evaluate the performance of a multi-class classifier and report degree of average accuracy over all categories. With respect to this dataset, the first 200 frames in each video are used to extract the descriptors.


**Fig. 2.** Example frames from video sequences of KTH (a) and UCF (b) datasets

UCF Sport dataset contains ten categories from 150 video sequences of different sporting action that reveals a large intra-class variability due to a wide scope scenes and viewpoints. There are different actions: diving, golf swing, kicking, lifting, riding, running, skateboarding, swing bench, swing side-angle and walking, as illustrated in Figure 2(b). Following the standard setting [11], Leave-one-out cross-validation (LOOCV) is performed on this database. LOOCV selects a video sequence for testing set and the remaining videos as the training set, and the overall accuracy is obtained by averaging the accuracy of all iterations. In our experiments, every frame are down-sampled by factor of 0.5.

**Parameter settings**: Parameters in our experiments:

- Descriptor parameters : we choose number of orientation $d = 5$ to compute gradient and orientation quantization. Gaussian filter with kernel size $5 \times 5$ and the standard deviation $\sigma = 1$. Cell size for SSD computation $r = 3$; $n = 8$ is number of neighboring cells in each block and threshold $T = r \times r \times \tau^2$ (where $r \times r$ cell size, $\tau$: the threshold per pixel, ranging from 5 to 7). We select $16 \times 16$ patches ($w$=16, $h$=16) to calculate histogram at Step 3 in Section 2.1. As a result, a descriptor is a 512-dimension vector ($D$=512).

- VLAD parameters: the extracted descriptors are clustered using $K$-means ($k = 24$ clusters). The length of the feature vector representing a video is

therefore a 12288-dimension vector. We utilize $L2$-norm over feature vectors represented by VLAD.

– SVM parameters: in this case of multi-class classification, we implement a one-vs-all non-linear SVM with radial basis function (RBF) kernel: $C = 4$, $\gamma = 0.5$.

### 3.2 Experimental Results on KTH dataset

Table 1 shows the confusion matrix containing the detailed confusion between action classes. It can be noted that the confusion happens mainly between "running" and "jogging" classes due to their similarity of local space-time events. Also, there is a slight misclassification of "hand clapping" and "hand waving". While the best performance belongs to "walking" category with 99.4% accuracy, "running" class is identified to the lowest recognition rate (86.7%). Moreover, two distinguished groups - hand actions (i.e. boxing, hand-clapping and hand-waving) and leg actions (i.e. jogging, running and walking) - are completely separated, this proves the efficiency of our proposal.

**Table 1.** Confusion matrix on KTH dataset

| | **Box** | **clap** | **Wave** | **Jog** | **Run** | **Walk** |
|---|---|---|---|---|---|---|
| **Boxing** | 97.3 | 2.7 | 0 | 0 | 0 | 0 |
| **clapping** | 2.7 | 91.9 | 5.4 | 0 | 0 | 0 |
| **Waving** | 0 | 0.8 | 99,2 | 0 | 0 | 0 |
| **Jogging** | 0 | 0 | 0 | 91.9 | 5.4 | 2.7 |
| **Running** | 0 | 0 | 0 | 10,6 | 86.7 | 2.7 |
| **Walking** | 0 | 0 | 0 | 0.6 | 0 | 99,4 |

**Comparison to state-of-the-art:** Several literatures on the KTH dataset are revealed in Table 2. The average accuracy of our method is 94,4%. It can be seen that our proposed method outperforms the almost considered algorithms even some Convolutional Neuron Networks-based ones, except the Action Bank in [25]. While our algorithm is simple to implementation and of low complexity, Action Bank [25] requires a huge dataset in the training step.

### 3.3 Experimental Results on UCF Sport dataset

UCF Sport dataset is more challenging than the KTH dataset due to a wide range of scenes and view points. Regarding to Table 3, diving-side, swing-bench and swing-side obtain the best performances, 100%, 95% and 92% respectively. We can see that the most of the errors are due to mixing up of the classes "kicking" and "riding".

**Table 2.** Comparision of accuracy (%) on the KTH dataset

| **Algorithm** | **Accuracy** (%) | **Algorithm** | **Accuracy** (%) |
|---|---|---|---|
| Wang *et al* [8] | 94.2 | Kovashka *et al* [26] | 94.53 |
| Laptev *et al* [6] | 91.8 | Gross *et al* (MIP)[15] | 93.0 |
| Klaser *et al* [11] | 91.4 | Le *et al* [3] | 93.9 |
| Action Bank *et al*[25] | 98.2 | W. Taylor *et al* (Conv) [27] | 90.0 |
| Liu *et al*[28] | 93.5 | S.Ji *et al* (Conv) [1] | 90.2 |
| Lior Wolf *et al* [14] | 90.1 | **MOMP** | **94.4** |

**Table 3.** Confusion matrix on UCF Sport dataset

| | **Dive** | **Golf** | **Kick** | **Lift** | **Ride** | **Run** | **Skate** | **SwBench** | **SwSide** | **Walk** |
|---|---|---|---|---|---|---|---|---|---|---|
| **Diving** | 1.0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| **Golf** | 0 | 0.78 | 0 | 0 | 0.05 | 0 | 0 | 0 | 0 | 0.17 |
| **Kicking** | 0 | 0 | 0.75 | 0 | 0.05 | 0.05 | 0 | 0 | 0 | 0.10 |
| **Lifting** | 0 | 0 | 0 | 0.83 | 0 | 0 | 0 | 0 | 0 | 0.17 |
| **Riding** | 0 | 0 | 0.25 | 0 | 0.67 | 0.08 | 0 | 0 | 0 | 0 |
| **Run-Side** | 0 | 0.08 | 0.23 | 0 | 0 | 0.61 | 0 | 0 | 0 | 0.08 |
| **SkateBoard** | 0 | 0.08 | 0 | 0 | 0 | 0 | 0.58 | 0.17 | 0 | 0.17 |
| **Swing-Bench** | 0 | 0 | 0.05 | 0 | 0 | 0 | 0 | 0.95 | 0 | 0 |
| **SideAngle** | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0.08 | 0.92 | 0 |
| **Walk-Front** | 0 | 0.045 | 0.045 | 0 | 0 | 0 | 0.045 | 0 | 0.045 | 0.82 |

**Comparison to state-of-the-art:** From Table 4, the overall accuracy we obtain for this dataset is 80.0%. The results in the right side show the good performance of our method on UCF Sport dataset when compared to recent methods with the same experimental settings (those methods use similarly simple classifier). Although not more effective than some methods like dense trajectories and motion boundary descriptors (MBH) [8], the proposed framework is much more simple and faster.

## 4 Conclusion

In this paper, we introduce novel features based on the relationships between the local gradient distributions of neighboring patches coming from successive frames in video. The descriptor is very efficient to compute and simple to implement , thus can be suitable for real-time applications. This descriptor is then combined with VLAD and SVM in order to better represent and classify the actions. Experimental results show that our proposed framework obtains good performance on some action benchmarks such as KTH and UCF Sport datasets. In future, the proposed algorithm will be evaluated on other datasets or other applications such as texture classification or face recognition.

**Table 4.** Comparision of accuracy (%) on the UCF Sport

| **Description method** | **Accuracy (%)** | **Description method** | **Accuracy (%)** |
|---|---|---|---|
| Harris3D+HOG [9] | 71.4 | Kovashka *et al* [26] | 87.27 |
| Harris3D+HOF [9] | 75.4 | Dense+HOG/HOF [9] | 81.6 |
| Harris3D+HOG/HOF [9] | 78.1 | MBH+Dense traj. [8] | 84.2 |
| Gabor+HOG/HOF [29] | 77.7 | ConvNet (Le *et al*) [3] | 86.5 |
| Hessian+HOG/HOF [9] | 79.3 | **MOMP** | **80.0** |

## References


1. Ji, S., Xu, W., Yang, M., Yu, K.: 3d convolutional neural networks for human action recognition. Pattern Analysis and Machine Intelligence **35** (2013) 221–231
2. Simonyan, K., Zisserman, A.: Two-stream convolutional networks for action recognition in videos. In: Advances in Neural Information Processing Systems. (2014) 568–576
3. Le, Q.V., Zou, W.Y., Yeung, S.Y., Ng, A.Y.: Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In: CVPR 2011 IEEE Conference on, IEEE (2011) 3361–3368
4. Tran, D., Bourdev, L., Fergus, R., Torresani, L., Paluri, M.: Learning spatiotemporal features with 3d convolutional networks. In: 2015 IEEE International Conference on Computer Vision (ICCV), IEEE (2015) 4489–4497
5. Laptev, I.: On space-time interest points. International Journal of Computer Vision **64** (2005) 107–123
6. Laptev, I., Marszałek, M., Schmid, C., Rozenfeld, B.: Learning realistic human actions from movies. In: CVPR 2008. IEEE Conference on, IEEE (2008) 1–8
7. Wei, Q., Zhang, X., Kong, Y., Hu, W., Ling, H.: Group action recognition using space-time interest points. In: International Symposium on Visual Computing, Springer (2009) 757–766
8. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. International Journal of Computer Vision **103** (2013) 60–79
9. Wang, H., Ullah, M.M., Klaser, A., Laptev, I., Schmid, C.: Evaluation of local spatio-temporal features for action recognition. In: BMVC 2009-British Machine Vision Conference, BMVA Press (2009) 124–1
10. Wang, H., Schmid, C.: Action recognition with improved trajectories. In: Proceedings of the IEEE International Conference on Computer Vision. (2013) 3551–3558
11. Klaser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: BMVC 2008-19th British Machine Vision Conference, British Machine Vision Association (2008) 275–1
12. Lowe, D.G.: Distinctive image features from scale-invariant keypoints. International Journal of Computer Vision **60** (2004) 91–110
13. Dalal, N., Triggs, B., Schmid, C.: Human detection using oriented histograms of flow and appearance. In: European conference on computer vision, Springer (2006) 428–441
14. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: Computer Vision, IEEE 12th International Conference on, IEEE (2009) 492–497

15. Kliper-Gross, O., Gurovich, Y., Hassner, T., Wolf, L.: Motion interchange patterns for action recognition in unconstrained videos. In: European Conference on Computer Vision, Springer (2012) 256–269
16. Jégou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: CVPR 2010. IEEE Conference on, IEEE (2010) 3304–3311
17. Vu, N.S., Caplier, A.: Face recognition with patterns of oriented edge magnitudes. In: European conference on computer vision, Springer (2010) 313–326
18. Vu, N.S.: Exploring patterns of gradient orientations and magnitudes for face recognition. Information Forensics and Security **8** (2013) 295–304
19. Jain, M., Jégou, H., Bouthemy, P.: Better exploiting motion for better action recognition. In: CVPR 2013. (2013) 2555–2562
20. Kantorov, V., Laptev, I.: Efficient feature extraction, encoding and classification for action recognition. In: Proceedings of the IEEE Conference on CVPR. (2014) 2593–2600
21. Perronnin, F., Dance, C.: Fisher kernels on visual vocabularies for image categorization. In: 2007 IEEE Conference on Computer Vision and Pattern Recognition, IEEE (2007) 1–8
22. Chang, C.C., Lin, C.J.: Libsvm: a library for support vector machines. ACM TIST **2** (2011) 27
23. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: a local svm approach. In: Pattern Recognition, 2004. ICPR 2004. Proceedings of the 17th International Conference on. Volume 3., IEEE (2004) 32–36
24. Rodriguez, M.D., Ahmed, J., Shah, M.: Action mach a spatio-temporal maximum average correlation height filter for action recognition. In: CVPR, 2008 IEEE Conference on, IEEE (2008) 1–8
25. Sadanand, S., Corso, J.J.: Action bank: A high-level representation of activity in video. In: Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on, IEEE (2012) 1234–1241
26. Kovashka, A., Grauman, K.: Learning a hierarchy of discriminative space-time neighborhood features for har. In: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on 2010, IEEE (2010) 2046–2053
27. Taylor, G.W., Fergus, R., LeCun, Y., Bregler, C.: Convolutional learning of spatio-temporal features. In: European conference on computer vision, Springer (2010) 140–153
28. Liu, L., Shao, L., Li, X., Lu, K.: Learning spatio-temporal representations for action recognition: A genetic programming approach. Cybernetics, IEEE Transactions **46** (2016) 158–170
29. Kläser, A.: Learning human actions in video. PhD thesis, PhD thesis, Université de Grenoble (2010)