COMBINING SPARSE APPEARANCE FEATURES AND DENSE MOTION FEATURES VIA RANDOM FOREST FOR ACTION DETECTION

Shuang Yang[*], *Chunfeng Yuan*[*], *Haoran Wang*[†], *Weiming Hu*[*]

*National Laboratory of Patten Recognition, Institute of Automation, CAS, Beijing, China
†School of Automation, Southeast University, Nanjing, China
Email: {syang, cfyuan, wmhu}@nlpr.ia.ac.cn; zidonghuabs@sina.com

ABSTRACT

This paper presents a new method to detect human actions in video by combining sparse appearance features and dense motion features in the unified random forest framework. We compute sparse appearance features to capture the main appearance changes and dense motion features to capture the tiny motion changes in the video. We take advantage of the randomization of channel selection in random trees to combine these two complementary types of features. In addition, linear classification is applied to grow each tree with high efficiency. Each leaf in these trees stores the class distribution and location information of the training samples and action detection for the test video is accomplished by Hough voting of the leaves in each tree. Experimental results demonstrate that our method achieves the state-of-the-art performance on two datasets.

***Index Terms*—**Action detection, Multiple features, Random forest, Hough voting

1. INTRODUCTION

Human action recognition from videos has been widely used in many computer vision applications. In many of these settings, it is not only essential to correctly identify the action class, but also desirable to partition out the temporal or spatial-temporal range within which the activity occurs in the video. However, most of the existing methods only focus on the classification problem and assume that the range has been known exactly. In this paper, we focus on the joint classification and localization problem.

1.1. Related Work

Recently, many approaches apply appearance-based features for action recognition and detection and they are proved to be effective [1-5]. Oshin *et al.* [6] perform action recognition based on the distribution of spatial-temporal interest points. Gorelick *et al.* [7] use 3-D shapes induced by the silhouettes of human for action recognition. However, the appearance features based recognition is not always valid, especially for the videos with moving background. To solve this problem, some other approaches tend to combine multiple features for recognition and detection [8-12]. Among the various work for action detection with multiple features, trees based methods have been proved to be efficient [13-17]. Lin *et al.* [15] use shape and motion features for action recognition. Yao *et al.* [17] apply dense sampled low-level features integrated with random forest [18] to detect actions in video. However, most of the current trees based methods have two main limitations. First, most of these methods just combine many features together, but do not consider the complementary property and redundancy property between features. Second, they grow each tree by a number of repetitious comparisons between the elements of each feature. So it is time-consuming to handle multiple high dimensional features by these methods.

1.2. Our Work

To solve the above limitations, this paper combines two complementary types of features for action detection: sparse appearance features and dense motion features. They are appropriate to jointly represent the actions in video for their complementary property, which is few considered in earlier studies. Besides, we apply the linear classification method to improve the efficiency and the accuracy of each random tree in the forest. The main contributions of this work are summarized as follows: First, we present a new method to represent the actions with the two complementary types of features for action detection. Second, we take advantage of the randomization property to combine these two types of features in each random tree for action detection, and furthermore, this framework is also suitable for combining multiple high dimensional features. Third, linear classification method is introduced to grow each tree in the forest. This makes our approach more robust compared with the traditional random forests. Finally, we compare the experimental results with other related methods and demonstrate the effectiveness of our method.

2. THE PROPOSED APPROACH

As the general framework shown in Fig.1, we extract the sparse appearance features and dense motion features in multiple spatial scales at first, and then divide each video into temporal or spatio-temporal volumes. Each volume is represented by two complementary types of features. In the training stage, a subset of the samples is randomly

978-1-4799-0356-6/13/$31.00 ©2013 IEEE

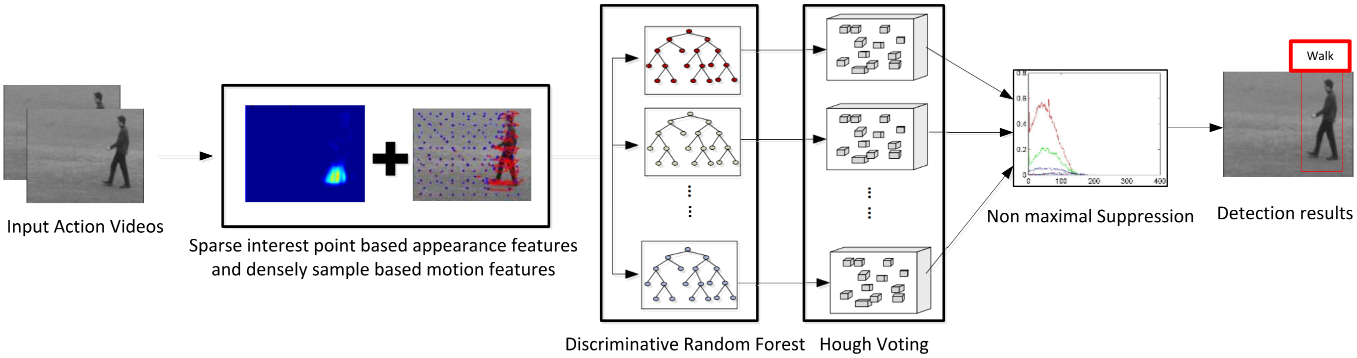
Walk
Input Action Videos
Sparse interest point based appearance features
and densely sample based motion features
Non maximal Suppression
Detection results
Discriminative Random Forest
Hough Voting

Fig.1. The proposed framework for action detection.

selected as the root node to grow each random tree, and the linear classification method is introduced to split the branches. In the testing stage, Hough voting is performed for each action's class and location by the leaves of the trees in the forest. Finally, non-maxima suppression is applied to get the final detection results.

2.1. Representation of Human Actions

The sparse appearance features are popular for its efficiency to capture the main appearance changes of the actions. However, this type of feature is too sparse to capture actions with tiny appearance changes, such as the Golf-Swing action which only involves the movement of the arms and the golf club in a small range. So the dense motion features are applied to complete the representation with the sparse appearance features for action detection.

For the sparse appearance features, we apply the extended Harris interest point detection and get the response values of each region in the video. We keep the sparse salient regions with a response value above some threshold. Then 3-D SIFT features [19] are computed to describe these regions. Compared with the traditional 2D SIFT descriptor, the 3D SIFT features capture the appearance changes both in the spatial and temporal domain, so they are more robust. In the end, all these features are clustered to generate the appearance codebook Γ_A, which represents the salient appearance changes of the action in the video.

In order to complement the sparse appearance features for action representation, we adopt the dense sampling based motion features. Feature points are densely sampled in multiple spatial scales, and then tracking is performed using the optical flow in the corresponding spatial scale over a fixed number of frames. Then we separately compute the derivatives for horizontal and vertical components of the optical flow to get the MBH (Motion Boundary Histogram) features [12, 20]. Finally, all these features are clustered to generate the motion codebook Γ_M.

The optical flow of each feature point represents the absolute changes between frames and the MBH features compute the relative changes between optical flows, which means the MBH features only capture the motions between the foreground and the background. So we can remove the continuous movement of the background in complex videos and keep the action regions only.

For each video, we divide it into temporal or spatial-temporal volumes, and then each volume is represented by four types of information: sparse appearance features, dense motion features, location information, and the action class, defined as X_1, X_2, O and c respectively.

Specifically, the features in each volume are quantized according to the appearance codebook Γ_A and the motion codebook Γ_M respectively .Then the two types of quantized features are used to generate the histogram representation X_1 and X_2 of the volume individually, as shown in Fig.2.

The location of each volume is represented by the average of all the offset of the feature points in the volume. For each volume in a test video, the location O is defined as the average of all the feature point coordinates (x_i, y_i, t_i) in the volume. For each volume in a training video, the location O is defined as the average of all the offset O_i of the feature point (x_i, y_i, t_i) relative to the action center (c_x, c_y, c_t), which is defined as

$$O_i = (c_x - x_i,\ c_y - y_i,\ c_t - t_i) \qquad (1)$$

2.2. Random Forest Framework

We take advantage of the random channel selection in random trees to combine the four types of information for action detection. In addition, linear classification is applied to grow each random tree in the forest which makes the framework more efficient and robust.

For each random tree, a number of training samples are randomly selected as the root node to avoid overfitting. The dimensionalities of the two features X_1 and X_2 are assumed to be d_1 and d_2 respectively. For each node S of each tree, one type of feature X with dimension d is selected at random to split the node ($X \in \{X_1, X_2\}$ and $d \in \{d_1, d_2\}$). Each type of feature is selected with equal probability, so the framework can fully utilize the two types of features and is also suitable for multiple other features.

The split method in most of the current work is to compare the absolute values at two random positions of the samples, which is time-consuming and inaccuracy for high dimensional data. This paper proposes to randomly select a sub-feature and splits the node by a linear classifier with the

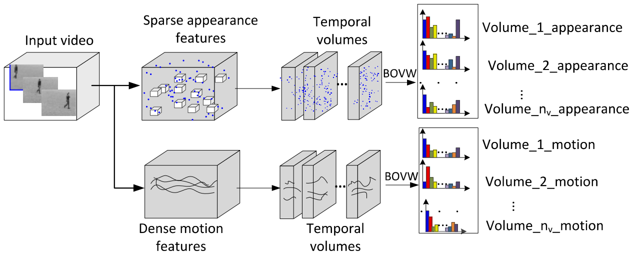

Fig.2. Illustration of the representation.

sub-feature.

At first, a sequence $\boldsymbol{p} = [p_1, p_2 \dots p_m]$ is generated at random to represent the different indexes of the elements in feature *X*. Then we get the selected sub-feature *f* as

$$f = [\, X_{p_1}, X_{p_2}, \dots, X_{p_m}] \tag{2}$$

where $p_i \leq d(i = 1,2 \dots m)$ and $X_{p_i}$is the p_i-th element of *X*.

The sub-feature *f* is then used to perform the linear classification to split the node *S*. The binary test $t_{S,f,m,k,b}$ at node *S* with linear classification parameters *k* and *b* is defined as

$$t_{S,f,m,k,b} = \begin{cases} 0, & if\ k * f + b \geq 0 \\ 1, & otherwise \end{cases} \tag{3}$$

The best binary test should separate the samples in the node S with minimal class uncertainty or location uncertainty so that the samples belonging to the same class or in the same neighborhood are separated in the same child node. We define $|S_c|$ as the number of the training samples with class label *c* in node *S*, $|S|$ as the number of all the samples in node *S*, *C* as the number of all the action classes, and c_s be the number of training samples with class label *c* in node *S*. Then the class uncertainty U_C and the location uncertainty U_o of node S is defined as

$$U_C = -|S| \times \sum_{c=1}^{C} \frac{|S_c|}{|S|} \log(\frac{|S_c|}{|S|}) \tag{4}$$

$$U_o = \sum_{c=1}^{C} \sum_{i=1}^{c_s} \left| |O_i - \overline{O_c}| \right|_2^2 \tag{5}$$

$$\overline{O_c} = \frac{1}{c_s} \sum_{i=1}^{c_s} O_i \tag{6}$$

where O_i and $\overline{O_c}$ are the offset of the *i*-th training sample and the mean offset of class *c* in node *S* respectively.

In each split of node *S*, the class uncertainty or the location uncertainty is selected at random to make sure that the samples either belonging to the same class or in the same spatial-temporal neighborhood will arrive at the same leaf. We iterate the process with different values and the final parameters are defined as

$$\{k, b\} = argmin_{k,b}\ \{U, where\ U = U_c\ or\ U_o\} \tag{7}$$

When we get *k*=1 and *m*=1, the node is split by a surface in the direction parallel to the axes, which is equivalent to the traditional method. In our method, the separate surface can change in different directions and locations with different parameters *k* and *b*. So our approach is more robust and more efficient to complex data.

Each random tree continues growing with the above split method until the tree reaches some maximum depth or the number of samples in the leaf below some value. The parameters *p, k* and *b* are recorded for each split in the growing process and the following information is stored in each leaf *l*: (1) the class distribution $q_{c,l}$ which is defined as the ratio between the number of samples $n_{c,l}$ within class *c* in the leaf and the number of the samples n_l in the leaf; (2) the offset $O_{c,l,i}$ of each sample *i* with class label *c* in the leaf.

2.3. Action Detection

Each volume represented by (X_1, X_2, *O*, *c*) in the test video goes through each tree according to the split parameters *p, k, b* and reaches the leaf node *l*. Hough voting is performed by the leaf and each vote is weighted to avoid the problem induced by the difference in the quantity of the training samples. The weight of each vote by leaf *l* for class *c* is defined as

$$w_{c,l} = \frac{q_{c,l}}{n_{c,l}} \tag{8}$$

Let O_j=(x_j, y_j, t_j) be the offset of the *j*-th test volume which arrives at leaf *l* and O_{cli} =(Δx_i, Δy_i, Δt_i)$_{cli}$ be the offset of the *i*-th training sample with class label *c* in leaf *l*, then the vote from the *i*-th training sample in leaf *l* is considered available only when the following conditions are satisfied:

$$\begin{gathered} 0 \leq x_j - \Delta x_i \leq n\_width \\ 0 \leq y_j - \Delta y_i \leq n\_height \\ 0 \leq t_j - \Delta t_i \leq n\_frame \end{gathered} \tag{9}$$

where *n_width*, *n_height* and *n_frame* are the width, height and number of frames of the test video respectively.

When the conditions are satisfied, the vote is weighted by $w_{c,l}$ to vote for class *c* and location G_l defined as

$$G_l = (x_j - \Delta x_i,\ y_j - \Delta y_i,\ t_j - \Delta t_i,) \tag{10}$$

Let $N_{c,l}$ be the number of volumes satisfied the condition, then the final vote for class *c* defined as

$$vote_c = \sum_l w_{c,l} * N_{c,l} \tag{11}$$

Then non-maximal suppression is performed to get the entire local maximum of the votes, and we decide the final class label and location by the max number of local maximum above a threshold.

3. EXPERIMENTAL RESULTS

We tested our approach on two public datasets: the KTH dataset [21] and the UCF-Sports action dataset [22] which is much more challenging. Our experiments were done with a five-fold cross-validation [17, 23]. We compared our results with other related methods which combined many types of features [12, 17]. In [12], action recognition is performed by combining MBH features with three other types of features. In [17], Hough voting is used with random forest to combine six types of features for action detection. The results show that the two types of features in our framework outperform the others.

3.1. Experimental Results on the KTH Dataset

Experiments were done in two groups on the KTH dataset:

(1) only utilizing sparse appearance features in the proposed framework; (2) combining sparse appearance features with the dense motion features in the way presented.

We compare our results with the state-of-the-art results in Table 1. As Table.1 shown, with single sparse appearance features, the classification accuracy achieves 93.8%, which is comparable to [17]. So this proves the effectiveness of our framework. When combining the two types of features in our unified framework, we report an average accuracy of 96% which get the highest accuracy compared with others. The results show that the only two types of features in our proposed framework outperform [12] in which four types of features are combined. In addition, we present the confusion matrix in Fig.6, which shows that the accuracy of five classes among the total six classes is above 95%.

In addition, we present the temporal localization results by dividing the video into temporal volumes. This can be extended easily to the spatial domain by dividing the video into spatio-temporal volumes. The result is considered correct only if the classification is right and the intersection–union ratio between the predicted and ground truth boxes is greater than 0.5. The localization results are presented in Table 2 and show the effectiveness of our approach.

Method	Accuracy(%)
Feature-tree[24]	72.9
Voc. forest[14]	93.2
Hough forest[17]	93.5
Dense Trajectories[12]	94.2
Space-time pyramids [25]	91.8
Ours(only using sparse features)	**93.8**
Ours (combining sparse and dense features)	**96.0**

Table 1. Comparison of the recognition results on the KTH dataset.

Method	Fold1	Fold2	Fold3	Fold4	Fold5	Mean
sparse features	85.8	83.2	81.7	79.8	90.8	**84.3**
Combined	93.3	90.8	85.8	79.8	95.8	**89.1**

Table 2. The localization results on the KTH dataset. (%)

Box	**98**	2				
Hand-clap		**98**	2			
Hand-wave		3	**97**			
Jog				**79**	15	6
Run				3	**96**	1
Walk				4		**96**

Fig.6. Confusion matrix on the KTH dataset. (%)

3.2. Experimental Results on the UCF-Sports Datasets

The UCF-Sports action dataset contains 150 broadcast sports action videos from 10 action classes with a wide range of scenes and viewpoints in unconstrained scenes.

On the UCF Sports action dataset, we also performed two groups of experiments similar to those on the KTH dataset. We present the five folds recognition results in Table 3 and compare the results with other related methods in Table 4. We achieve an average accuracy of 87.3% when only the sparse appearance features are used. It outperforms [17] in which recognition is also performed by Hough voting. When the dense motion features are combined in the proposed framework, the highest accuracy is 98.1% in the 4-th fold and we achieve an average accuracy of 90.3%. The result outperforms [12] in which MBH descriptor is combined with three other types of features. As the confusion matrix in Fig.7 shown, there are 6 classes in the total 10 classes for which the accuracy is above 95%.

Method	Fold1	Fold2	Fold3	Fold4	Fold5	Mean
sparse features	82.4	89.4	86.7	94.4	84.6	**87.3**
Combined	86.8	90.9	90	98.1	86.5	**90.3**

Table 3. The recognition results of the five folds on the UCF dataset. (%)

Method	Accuracy (%)
subspace forest[27]	91.3
Spatio-temporal features[26]	85.6
Dense Trajectories[12](MBH)	84.8
DenseTrajectories[12](combined)	88.2
Ours(only using sparse features)	**87.3**
Ours (combining sparse and dense features)	**90.3**

Table 4. Comparison of the recognition results on the UCF dataset.

Dive	**100**									
Golf		**83**		8	3				6	
Kick			**100**							
Lift				**100**						
Ride			4		**63**	21		13		
Run				4	12	**69**	11	04		
Skate							**100**			
Sw-1		2						**98**		
Sw-2			12						**88**	
Walk							2	2		**96**

Fig.7. Confusion matrix on the UCF dataset. (%)

4. CONCLUSIONS

We have presented a new method to detect actions in video by combining sparse appearance features and dense motion features. Randomization of the channel selection in each random tree is utilized to combine the two complementary types of features, and linear classification is introduced to grow each tree with high efficiency. Moreover, this framework is suitable for combining high dimensional data and the experiments show the efficiency.

5. ACKNOWLEDGEMENTS

This work is partly supported by NSFC (Grant No. 60935002, 61100099), the National 863 High-Tech R&D Program of China (Grant No. 2012AA012504), the Natural Science Foundation of Beijing (Grant No. 4121003), and the Project Supported by Guangdong Natural Science Foundation (Grant No. S2012020011081).

6. REFERENCES

[1] J. Liu, J. Luo, and M. Shah, "Recognizing Realistic Actions from Videos 'in the Wild'," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.1996-2003, 2009.

[2] J. C. Niebles and L. Fei-Fei, "A Hierarchical Model of Shape and Appearance for Human Action Classificat- -ion," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.1-8, 2007.

[3] D. Parikh, C. Zitnick, and T. Chen, "Unsupervised Learning of Hierarchical Spatial Structures in Images," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.2743-2750, 2009.

[4] A. Kovashka and K. Grauman, "Learning A Hierarchy of Discriminative Space-time Neighborhood Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.2046 -2053, 2010.

[5] I. Laptev and T. Lindeberg, "Space-time Interest Points," *Proc.Int. Conference on Computer Vision,* pp.432-439, 2003.

[6] O. Oshin, A. Gilbert, I. Illingworth and R Bowden, "Action Recognition Using Randomized Ferns," *In Proc. Int. Conference on Computer Vision,* pp.530-537, 2009.

[7] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as Space-time Shapes," *In Proc. Int. Conference on Computer Vision*, pp.1395-1402, 2007.

[8] A. Oikonomopoulos, I. Patras, and M. Pantic, "An Implicit Spatio-temporal Shape Model for Human Activity Localization and Recognition," *Proc.IEEE Conf. Computer Vision and Pattern Recognition,* pp.27-33, 2009.

[9] B. Packer, K. Saenko, and D. Koller, "A Combined Pose, Object and Feature Model for Action Understanding," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.1378- 1385, 2012.

[10] C.Y. Chen and K. Grauman, "Efficient Activity Detection with Max-subgraph Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,*pp.1274 -1281, 2012.

[11] B. Yao, A. Khosla, and L. Fei-Fei, "Combining Randomi- -zation and Discrimination for Fine-Grained Image Categorization," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.1577-1584, 2011.

[12] H.Wang, A.Kläser, C.Schmid, and C.L.Liu. "Action Recognition by Dense Trajectories," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.3169- 3176, 2011.

[13] J. Gall and V. Lempitsky, "Class-Specific Hough Forests for Object Detection," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.1022- 1029, 2009.

[14] K. Mikolajczyk and H. Uemura, "Action Recognition with Motion-appearance Vocabulary Forest," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp1-8,2008.

[15] Z. Lin, Z. Jian, and L.S.Davis, "Recognizing Actions by Shape-motion Prototype trees," *Proc. Int. Conference on Computer Vision*, pp.444-451, 2009.

[16] Marcenaro, L., Marchesotti, L., Regazzoni, C.S., "Self- -organizing shape description for tracking and classifying multiple interacting objects, " *Image and Vision Computing*, 24 (11), pp.1179-1191,2006.

[17] A.Yao, J.Gall, and L. Van Gool, "A Hough Transform-Based Voting Framework for Action Recognition," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.2061-2068, 2010.

[18] L. Breiman, "Random Forests," Machine Learning, pp.5-32, 2001.

[19] P. Scovanner, S. Ali and M. Shah, "A 3-dimensional SIFT Descriptor and Its Application to Action Recognition," *Proceedings of the International Conference on Multimedia*, pp. 357–360, 2007.

[20] N. Dalal, B. Triggs, and C. Schmid, "Human Detection Using Oriented Histograms of Flow and Appearance," *Proc. European Conf. Computer Vision*, pp. 428-441, 2006.

[21] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing Human Actions: A Local SVM Approach, " *Proc. Int. Conference on Pattern Recognition*, pp. 32-36, 2004.

[22] M. D. Rodriguez, J. Ahmed, and M. Shah, "Action Mach: A Spatio-temporal Maximum Average Correlation Height Filter for Action Recognition, "*Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[23] T Lan, Y Wang, and G Mori, "Discriminative Figure-Centric Models for Joint Action Localization and Recognition," *In Proc. Int. Conference on Computer Vision*, pp.2003-2010, 2011.

[24] K. K. Reddy, J. Liu, and M. Shah, "Incremental Action Recognition Using Feature-tree," *In Proc. Int. Conference on Computer Vision*, pp.1010-1017, 2009.

[25] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld, "Learning Realistic Human Actions from Movies," *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[26] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of Local Spatio-temporal Features for Action Recognition," *Proc. British Machine Vision Conference*, 2009.

[27] S. O'Hara and B.A. Draper, "Scalable Action Recognition with A Subspace Forest," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* pp.1210-1217, 2012.