

# LOCALIZING AND RECOGNIZING ACTION UNIT USING POSITION INFORMATION OF LOCAL FEATURE

Yan Song<sup>1,2</sup>, Shouxun Lin<sup>1</sup>, Yongdong Zhang<sup>1</sup>, Lin Pang<sup>1,2</sup>, Juan Cao<sup>1</sup>

<sup>1</sup>Laboratory of Advanced Computing Research, Institute of Computing Technology,  
Chinese Academy of Sciences, Beijing, China

<sup>2</sup>Graduate University of Chinese Academy of Sciences, Beijing, China  
{songyan, sxlin, zhyd, panglin, caojuan}@ict.ac.cn

## ABSTRACT

Action recognition has attracted much attention for human behavior analysis in recent years. Local spatial-temporal (ST) features are widely adopted in many works. However, most existing works which represent action video by histogram of ST words fail to have a deep insight into a fine structure of actions because of the local nature of these features. In this paper, we propose a novel method to simultaneously localize and recognize action units (AU) by regarding them as 3D  $(x, y, t)$  objects. Firstly, we record all of the local ST features in a codebook with the information of action class labels and relative positions to the respective AU centers. This simulates the probability distribution of class label and relative position in a non-parameter manner. When a novel video comes, we match its ST features to the codebook entries and cast votes for positions of its AU centers. And we utilize the localization result to recognize these AUs. The presented experiments on a public dataset demonstrate that our method performs well.

*Index Terms*— human action, action unit, recognition

## 1. INTRODUCTION

Human behavior analysis has received much concern in the field of computer vision and multimedia analysis in recent years. Applications as intelligent visual surveillance, human-computer interaction and multimedia content retrieval demand for efficient and robust human action recognition techniques. Specifically, video retrieval based on action detection becomes a new way instead of traditional key frame-based indexing [1] [2]. To reach the target of understanding complex actions in real-world videos, we need robust methods to recognize and localize AUs in it.

There are many approaches for human action recognition [3]. Lots of methods based on human shape or silhouette [4] and human body model [5] require good segmentation of human body which needs human detection and tracking for general videos. In recent years, because of the success in object recognition field, local descriptors of interest points are extended to 3D for action recognition,

and video is regarded as 3D volume instead of image sequences. Local spatial-temporal features [6] are widely used. [7] presented a learning method for human action categories based on ST words and Latent Dirichlet Allocation. [8] proposed a scheme integrating ST features with SVM classification. And [9] combined local ST feature with spin-image feature by graph embedding. Compared with the previous works, these local ST feature based methods do not need preprocess such as segmentation which still remains a difficult task. Moreover, they perform better than global feature based methods do in the presence of occlusions. However, the weakness of local ST descriptor is that it lacks global spatial and long temporal information [7]. Most of these methods represented action video clips as ST word histograms so that they ignored spatial position relations and temporal orders of these local features. Most above methods were based on the assumption that a video clip only contains one person executing one action. Therefore they could not extract AUs and analyze actions consisting of different AUs.

We propose a novel method to localize and recognize human action units. The algorithm is designed to tell “When”, “Where” and “What” in the test video. Localization result  $(x, y, t)$  is a point in 3D space that  $t$  represents “When” and  $(x, y)$  represents “Where”. Recognized action type is “What”. To achieve this goal, we focus on AU rather than the whole action. An action unit is defined as the temporally indecomposable period in actions. Figure 1(b) gives a demonstration that three colored bars represent three “wave” AUs. Inspired by the work for object recognition in [10], we consider AU as a kind of 3D  $(x, y, t)$  object. And the center of AU is defined as the position which is in the middle of every dimension. Firstly local ST features are clustered into  $K$  prototypes represented by cluster centers. Then we construct a codebook recording all of the information of action class labels and relative positions of these local appearances, that is, where the local features appear on the AU “objects”. The codebook is built up in a non-parametric manner that it can describe the distribution in detail rather than to assume that it fits an oversimplifying Gaussian [10]. When a novel action comes,

we look up its ST features in the codebook and the corresponding entries cast votes for its AU center positions. Thus we get the probability distribution of its AU positions. Then localization result is utilized to recognize the AUs. In this way, we utilize the advantages of local descriptor while overcoming its intrinsic drawbacks.

The rest of the paper is organized as follows. Section 2 presents the probabilistic formulation for the algorithm. Section 3 presents the concrete implementation method. Experiments are showed in section 4. And we conclude this paper and discuss our future work in section 5.

## 2. PROBABILISTIC FORMULATION

In this section, we give the probabilistic formulation of our method (extending the work in [10]). We assume that there are  $K$  local movement prototypes in all kinds of actions. Each prototype is represented by  $I_i$  ( $1 < i < K$ ). If we observe a local movement  $a$  at location  $l$  in a novel video (we assume each observed local movement only belongs to a single AU), we can utilize the following marginalization to infer its AU center position  $x$  and its action class label  $c$ :

$$p(x, c | a, l) = \sum_i p(x, c | I_i, a, l) p(I_i | a, l) \quad (1)$$

The first term on the right means the probability distribution of AU position and action class label of a given local movement prototype. It is independent on  $a$ . This distribution is learned in the training step. The second term means the probability that  $a$  matches a local movement prototype. It is independent on  $l$ . Then (1) can be written as:

$$p(x, c | a, l) = \sum_i p(c | I_i, x, l) p(x | I_i, l) p(I_i | a) \quad (2)$$

Then we get the localization result as follows:

$$\begin{aligned} X &= \arg \max_x \sum_j p(x | a_j, l_j) \\ &= \arg \max_x \sum_j \sum_i p(x | I_i, l_j) p(I_i | a_j) \end{aligned} \quad (3)$$

$j$  means the  $j$ th local movement in the AU., we can utilize the localization result to get a reasonable recognition result :

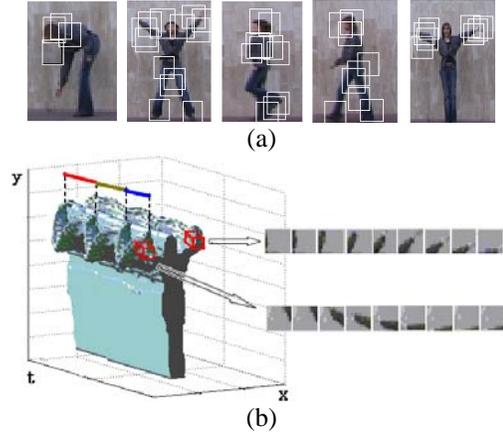
$$C = \arg \max_c \sum_j \sum_i p(c | X, I_i, l_j) p(X | I_i, l_j) p(I_i | a_j) \quad (4)$$

## 3. IMPLEMENTATION

In this section we present the concrete implementation of our method which can be divided into four steps. The first one is feature extraction which includes interest point detection and cuboid description. The second one is codebook construction which is also the training process of the method. Voting is the third step and at last we search for the densest locations in the voting space to localize and recognize AUs.

### 3.1. Feature Extraction

We choose to use the method in [6] to compute ST features. It is proved to perform well in human action recognition [7] [8] [9]. They apply two linear filters to spatial and temporal dimensions. Response function is represented as  $R = (I * g_\sigma * h_{ev})^2 + (I * g_\sigma * h_{od})^2$ , where  $g$  is the Gaussian filter applied on spatial dimensions ( $x, y$ ) and  $h_{ev}$  and  $h_{od}$  are a quadrature pair of 1D Gabor filters applied on temporal dimension  $t$ , which are defined as  $h_{ev}(t; \tau \omega) = -\cos(2 \pi t \omega) \exp(-t^2 / \tau^2)$  and  $h_{od}(t; \tau \omega) = -\sin(2 \pi t \omega) \exp(-t^2 / \tau^2)$ . Parameters  $\sigma$  and  $\tau$  are spatial and temporal detector scales respectively. The pixels which change intensely correspond to local maximums in  $R$ . The local cuboids of these points are extracted and gradient-based detector is applied. PCA is applied to reduce dimensionality. Figure 1(a) is some detected cuboids shown in single frames of action “bend”, “jack”, “skip”, “walk” and “wave”. Left of figure 1(b) is an action volume of “wave” in 3D space and the right unfolds two cuboids along temporal dimension.



**Fig.1 (a) Examples of detected cuboids shown in single frames (b) Two temporally unfolded cuboids and three AUs represented by three colored bars.**

### 3.2. Codebook Construction

Firstly, we introduce our codebook structure. Codebook has  $K$  entries. Each entry includes two parts: an ST feature prototype and a set of instances which can simulate the probability distribution of its AU center position and action class label.

According to the structure mentioned above, codebook construction consists of two steps. Firstly, we group all the ST features in the training set to  $K$  clusters by k-means clustering and Euclidean distance as metric. Cluster centers represent ST feature prototypes. Secondly, we perform the second iteration over the training set to assign each ST feature instance to the most similar ST feature prototype. Meanwhile, each instance is recorded with its action class label  $c$  and relative position ( $\Delta x, \Delta y, \Delta t$ ) to the AU center it belongs to. The codebook saves all the local movement

prototypes and the probability distribution of their relative positions to AU centers and their action class labels.

### 3.3. Probabilistic Voting

For a novel action video, probabilistic voting is the step in which we get the probability distribution of its AU center positions and action class labels. We extract ST feature by the way described in section 3.1. Then we match each of them to the codebook entries. Here, it is possible to let  $p(I_i/a)$  be the relative matching score, but for simplicity we only choose the most similar one. Once an entry in the codebook is matched, all its instances cast votes to the voting space  $(x, y, t)$  for positions of AU centers. And each vote has an action class label. If an ST feature occurs at location  $(x_{test}, y_{test}, t_{test})$  and the relative position of an instance of the matched entry is  $(\Delta x, \Delta y, \Delta t)$ , then the voting coordinates are calculated as  $(x_{test} + \Delta x, y_{test} + \Delta y, t_{test} + \Delta t)$ .

Figure 2(a) simulates the voting process that each cuboid descriptor matches a codebook entry and all of its instances cast votes. A cuboid casts several votes scattered in the voting space, but numbers of cuboids give a simulated distribution of AU center position. Figure 2(b) shows a real voting result which contains three AUs.

We notice that there is a problem when the numbers of AUs belonging to different action types in the training set have great differences. There is a bias in the recognition process because the action with more AUs predominates. To solve this problem, each vote is weighted by  $1/N_C$ , where  $N_C$  is the AU number of class C in the training set.

### 3.4. Localization and Recognition

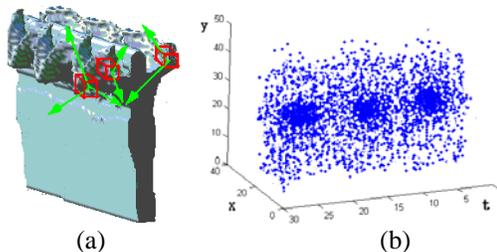
Since we get the voting result, which means the probability distribution of AU center position, we search for the densest positions in the voting space by mean-shift algorithm [11]. This step corresponds to formula (3). The positions where mean-shift terminates are the most probable locations of AU centers in the test action.

Given the center position of an AU, we can utilize it to obtain the action class label. We collect instances whose votes contribute to the densest position and only consider their votes for class label. Then recognition result is the action type which gets the most votes. This process corresponds to formula (4).

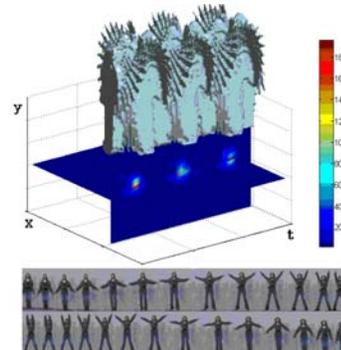
Figure 3 shows localization results of the action “jack” consisting of three AUs. The bottom of the figure shows successive frames of one AU. Visualization of voting space is explored with two orthogonal slice planes below the action volume and three densest positions are obvious. Each high light point corresponds to an AU center.

## 4. EXPERIMENT

We apply the proposed method on a publicly available data set: Weizmann action data set [12]. It consists of 10 actions



**Fig.2 (a) Simulation of voting process (b) 3D plots of a real voting result containing three AUs.**



**Fig.3 Localization result of action “jack”**

performed by 9 persons. There are 92 videos in all.

We extract ST feature as described in section 3.1. Parameters are set to  $\sigma = 2$  and  $\tau = 1.2$ . And ST feature is represented by concatenated vector of gradient which is reduced by PCA to 100 dimensions. ST features are clustered by k-means algorithm with  $k=500$ . We adopt leave-one-out cross-validation so we have 10 runs and the final result is the average. There are some actions with different directions so we firstly mirror the frames to obtain consistent action directions.

Firstly, we give an evaluation method for localization of AU. The work in [7] can localize actions in videos but they didn’t give a performance report on localization result. It is considered correctly localized if the output location  $L_o$  and the real location  $L_r$  satisfy the following:

$$\begin{aligned} |L_o(x) - L_r(x)| &< W * 0.2 \\ |L_o(y) - L_r(y)| &< H * 0.2 \\ |L_o(t) - L_r(t)| &< T * 0.3 \end{aligned}$$

where  $W, H, T$  mean the maximum width, height and the temporal duration of the AU. Localization recall is defined as the rate of number of correctly localized AUs and the ground-truth. And false alarm is defined as the rate of number of falsely localized AUs and all of localized ones. We ignore those AUs which are incomplete in time domain. Table 1 shows the result of localization. We notice that recall of “Pjump” is low because AU period is short and ambiguous that the clusters of votes tend to overlap and densest positions tend to merge. False alarms of “Run” and “Walk” are high. This is because we define that AUs of these two kinds of actions contain two footsteps with

different feet but local ST features can not tell the difference between them so that false centers appear. This problem may be solved by defining AUs of these two kinds of actions as only one footstep.

Then we recognize these correctly localized AUs and figure 4 shows the confusion matrix. All of the accuracies exceed 80% and average accuracy achieves 93%. We test our method on AUs so it is inappropriate to compare it with others which test on video clips. It is noticed that some visually similar actions are confused. The action of “jack” is recognized as “pjump” because their holistic directions of human body are similar. It is also the explanation for “skip” and “run”. It is interesting to notice that “wave1” and “wave2” tend to be confused. “wave1” is waving with one hand and “wave2” is waving with two hands. So their local ST features are very similar.

**Table 1.** Results of localization for AUs

Action type	Recall (%)	False Alarm (%)
Bend	100	0
Jack	100	0
Jump	80	4
Pjump	52.5	0
Run	100	33.4
Side	95.8	4.3
Skip	91.2	0
Walk	100	25.7
Wave1	83.3	0
Wave2	90.5	0

## 5. CONCLUSION AND FUTURE WORK

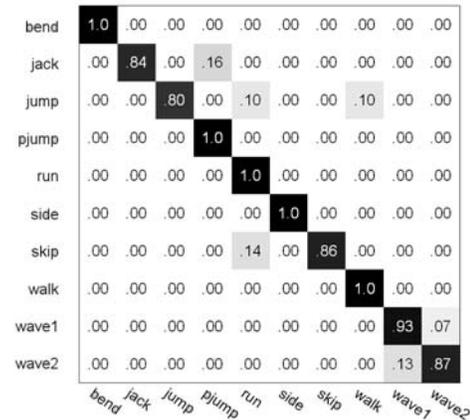
We present a method for localizing and recognizing AUs in videos based on local ST features by considering their position information. We give a method to evaluate the localization of AU. And the experimental result on a public dataset is promising. For future work, we plan to test our algorithm on videos containing compound actions and two or more actions executed simultaneously. And we will modify the algorithm to be scale-adaptive that it can be applied to real-world videos.

## 6. ACKNOWLEDGEMENT

This work was supported by the National Basic Research Program of China (973 Program, 2007CB311100), the National High Technology and Research Development Program of China (863 Program, 2007AA01Z416), the National Nature Science Foundation of China (60873165, 60802028), Co-building Program of Beijing Municipal Education Commission.

## 7. REFERENCES

[1] S.H. Jung, Y.L. Guo, H. Sawhney and R. Kumar, “Action Video Retrieval based on Atomic Action Vocabulary,” in *proceeding of the 1<sup>st</sup> ACM international conference on MIR*, pp. 245-252, 2008.



**Fig.4** Confusion matrix

[2] R.R. Ji, X.S. Sun, H.X. Yao, P.F. Xu and T.Q. Liu, “Attention-Driven Action Retrieval with DTW-based 3D Descriptor Matching,” in *proceeding of the 16<sup>th</sup> ACM international conference on MM*, pp.619-622, 2008.

[3] P. Turaga, R. Chellappa, V.S. Subrahmanian and O. Udrea, “Machine Recognition of Human Activities: A Survey,” *IEEE transaction on Circuits and Systems for Video Technology*, vol. 18, no. 11, pp. 1473-1488, Nov. 2008.

[4] A.F. Bobick and J.W. Davis, “The Recognition of Human Movement using Temporal Templates,” *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol.23, no.3, pp.257-267, Mar. 2001.

[5] G. Mori, X. Ren, A.A. Efros and J. Malik, “Recovering Human Body Configurations: Combining Segmentation and Recognition,” in *proceeding of IEEE International Conference on Computer Vision and Pattern Recognition*, pp.326-333, 2004.

[6] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie, “Behavior Recognition via Sparse Spatio-Temporal Features,” in *proceeding of IEEE international workshop on Visual Surveillance Performance Evaluation and Tracking Surveillance*, pp.65-72, 2005.

[7] J.C. Niebles, H.C. Wang and F.F. Li, “Unsupervised Learning of Human Action Categories using Spatial-Temporal Words,” *International Journal of Computer Vision*, vol.79, pp.299-318, 2008.

[8] C. Schudt, I. Laptev and B. Caputo, “Recognizing Human Actions: A Local SVM Approach,” in *proceeding of the 17<sup>th</sup> International Conference on Pattern Recognition*, vol.3, pp.32-36, 2004.

[9] J. Liu, S. Ali and M. Shah, “Recognizing Human Actions using Multiple Features,” in *proceeding of IEEE International Conference on Computer Vision and Pattern Recognition*, pp.1-8, 2008.

[10] B. Leibe, A. Leonardis and B. Shiele, “Robust Object Detection with Interleaved Categorization and Segmentation,” *International Journal of Computer Vision*, vol.77, pp.259-289, 2008.

[11] D. Comaniciu and P. Meer, “Mean Shift: A Robust Approach Toward Feature Space Analysis,” *IEEE transaction on Pattern Analysis and Machine Intelligence*, vol.24, pp.603-619, May 2002.

[12] M. Blank, L. Gorelick, E. Shechtman, M. Irani and R. Basri, “Action as Space-Time Shapes,” in *proceeding of IEEE International Conference on Computer Vision*, vol.2, pp.1395-1402, Oct. 2005.