LETTER

Mining Spatial Temporal Saliency Structure for Action Recognition

Yinan LIU[†a)], **Qingbo WU**[†], **Linfeng XU**[†], ***Nonmembers***, ***and*** **Bo WU**[†], ***Member***

SUMMARY Traditional action recognition approaches use pre-defined rigid areas to process the space-time information, e.g. spatial pyramids, cuboids. However, most action categories happen in an unconstrained manner, that is, the same action in different videos can happen at different places. Thus we need a better video representation to deal with the space-time variations. In this paper, we introduce the idea of mining spatial temporal saliency. To better handle the uniqueness of each video, we use a space-time over-segmentation approach, e.g. supervoxel. We choose three different saliency measures that take not only the appearance cues, but also the motion cues into consideration. Furthermore, we design a category-specific mining process to find the discriminative power in each action category. Experiments on action recognition datasets such as UCF11 and HMDB51 show that the proposed spatial temporal saliency video representation can match or surpass some of the state-of-the-art alternatives in the task of action recognition.

key words: *action recognition, video saliency, feature pooling*

1. Introduction

With the expansion of online video collections, action recognition has become an important problem in computer vision. Most researchers focus on the real-world action recognition problem; dataset such as HMDB51 [1] are quite challenging due to the variations in video size, viewpoints, scale, camera motion, and the position of the action. One of the efficient approaches in action recognition is bag-of-visual-words (bovw) [2], [3], which first extracts local features, then encodes them into a codebook, and uses video-wise pooling to build the final representation. The original bag-of-visual-words discards the useful space-time information of the video data [4]. To tackle this drawback, [3], [5], [6] pool the local features over space-time pyramids or pre-defined cuboids. However, since the real world video data is unconstrained, even the same action may happen at different spatial area and different temporal extent. Thus we should find a better way to represent the action videos. E.H.Taralova *et al.* [7] have shown the efficiency of supervoxel, however, they treat each and every supervoxel with equal weights. [8], [9] have shown that not all areas will make the same contributions to the final results.

In this paper, our ultimate goal is to capture the spatial temporal structure of the action, thus we over-segment the videos into supervoxels instead of using pre-defined rigid areas. We first extract the low-level local features, and build the bag-of-visual-word (bovw) codebook using *k*-means. Then we pool these features over the extracted supervoxels, and each supervoxel can be represented as a fixed-size feature vector. We build a second bovw codebook based on the supervoxel feature vectors, and we will use this as the pre-process for our work. We propose 3 different saliency measures to capture the appearance and motion structure, namely frame-wise image saliency, lighting saliency and motion saliency. To further capture the region-based supervoxel features, we design a simple yet efficient technique to mine the discriminative power of each visual word in the supervoxel codebook, we will give more details in the later section.

In the remainder of this paper, we first overview the proposed approach, and then describe how we adapt the saliency measures to it. We then present our discriminative mining process. Our proposal is evaluated by two challenging datasets UCF11 [10] and HMDB51 [1].

2. Spatial Temporal Saliency Representation

Our work can be sumarized in Fig. 1 and Fig. 2, we first build a bag-of-visual-word (bovw) from the low-level features. For each video, a hierarchical supervoxel extraction is adopted, our approach will automatically choose the appropriate segmentation layer for each action category. We then pool the encoded low-level features in each supervoxel to build a supervoxel-based feature vector. To obtain the appearance and motion structure from each video, we pro-

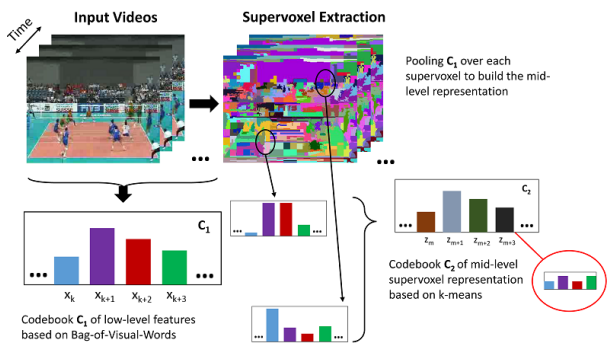

Fig. 1 The building of mid-level supervoxel representation. We first build low-level bag-of-visual-word (bovw) codebook $\mathbf{c_1}$. Then we pool $\mathbf{c_1}$ over the supervoxels and build bovw codebook $\mathbf{c_2}$. This is used as the pre-process step in our work.

Manuscript received April 26, 2016.
Manuscript revised June 14, 2016.
Manuscript publicized July 6, 2016.

[†]The authors are with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China.
a) E-mail: yates2012codec@126.com

DOI: 10.1587/transinf.2016EDL8093

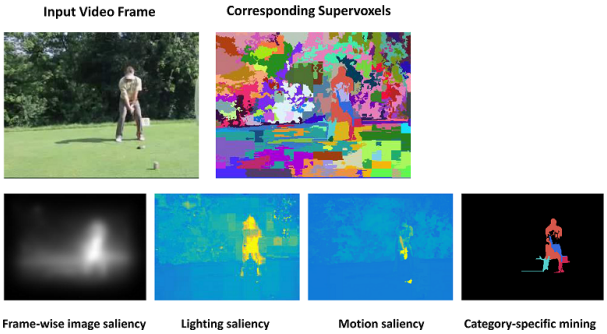

Fig. 2 Illustration of the spatial Temporal saliency structure, best viewed in colors. The first row indicates one certain video frame and the corresponding supervoxels. The first 3 elements in second row are image saliency, lighting saliency and motion saliency. The last element in second row is the proposed category-specific mining, we show the best scored supervoxels in the frame. We will give more detail in Sect. 2.

pose 3 different saliency measurements: frame-wise image saliency and lighting saliency will capture the appearance structure, while the optical flow based motion saliency can capture the motion structure. In each action category, some supervoxels are distinct and discriminative towards it, while other supervoxels might be discriminative for other categories. Based on this thought, we design a mining process to find the discriminative power of each visual word in the codebook, this can be used to further discover the S-T saliency of the video.

2.1 Saliency Measures

Saliency is an efficient way of finding area of interest [11]. In the video domain, besides the appearance cue in each frame, the inherent motion information has the same importance. Based on this consideration, we choose 3 saliency measures. For the static cues, we use a well-known image saliency algorithm [12]. Besides the RGB appearance cue, we also consider the lighting saliency which provides coarse object segmentation. To compute the lighting saliency map, a RGB frame is first converted to the LAB color space. We equally divide the Lighting component of LAB color space into 60 bins and use a center-surround sliding window saliency approach [13] to compute the lighting saliency maps. The motion saliency map mainly captures the sudden changes in temporal domain. We first extract the optical flow using [14] for consecutive frames in each video. Then we quantize the flow magnitude into 16 uniform bins and compute the motion saliency maps with the same approach as the lighting saliency.

Given a video V with T frames, for the $t^{th}(T \in [1,T])$ frame, the image saliency and lighting saliency maps are defined as S_I^t and S_L^t. The motion saliency map is defined as S_M^t where $T \in [1, T-1]$, the motion saliency is one frame less than the video sequence, thus we let $S_M^T = S_M^{T-1}$. Since all three of the saliency maps are pixel based, we take average saliency value within a supervoxel as the saliency value of the supervoxel. The combined saliency C_s for a given supervoxel $\mathbf{v}^i$ is shown in Eq. (1).

$$C_s(\mathbf{v}^i) = \omega_I S_I(\mathbf{v}^i) + \omega_L S_L(\mathbf{v}^i) + \omega_M S_M(\mathbf{v}^i) \tag{1}$$

where S_I, S_L, S_M, and ω_I, ω_L and ω_M are the image saliency, lighting saliency, motion saliency ane their weights respectively.

2.2 Category-Specific Discrimination Mining

As we mentioned before, after the pre-processing step, we can describe each supervoxel as a fixed-size feature vector. In our work, we design a simple yet efficient technique to calculate the discriminative power of each bin of the supervoxel codebook. First, for each action category, we cluster all the supervoxels from training videos into $c = 1, \ldots, C$ clusters, this is different from the step in building the supervoxel codebook, since the codebook is built only using one action category. The goal is to give each supervoxel an action specificity score which will be cumulated as the action specificity score of each visual word. We define $\phi(c) = N_f(c)/N(c)$, where $N_f(c)$ is the number of supervoxels from foreground (overlapping with ground truth bounding box larger than 50%) in cluster c, $N(c)$ is the total number of supervoxels in cluster c. Given the feature vector $\mathbf{d}_c^i$ of supervoxel $\mathbf{v}^i$, where $\mathbf{d}_c^i$ belongs to cluster c, the action specificity is calculated as Eq. (2).

$$S_a^v(\mathbf{v}^i) = \phi(c) \cdot exp(\frac{\|\mathbf{d}_c^i - \mathbf{d}_c\|}{r_c}) \tag{2}$$

where $\mathbf{d}_c$ is the cluster center of c and r_c is the radius. Since we can get each supervoxel's action specificity score, we compute the cluster's action specificity score as $S_a^c(c) = \frac{1}{K}\sum_{i=1}^{K} \mathbf{v}_c^i$, which means we compute the average value of top K supervoxel action specificity scores in cluster c as the score of the cluster. After getting the score of each cluster, we would like to find the most discriminative visual words for each action category. We find M visual words with the highest score for each action category. For a certain action class a, we treat each of these visual word as positive sample, we randomly sample the least scored visual words as negative samples. Then we train one-vs-one SVM [15] for each visual word, this will give each action category M classifiers, e.g. $\{R_1^a, \ldots, R_M^a\}$, where R_p^a is the p^{th} classifier for action a. We use these classifiers to evaluate whether a supervoxel is discriminative to certain action class.

We train one-vs-the-rest SVM classifiers for each action category. Given a test video V with T frames, using the previous steps, we can extract a set of supervoxels $\{\mathbf{v}^1, \ldots, \mathbf{v}^N\}$, each one is represented by a fixed-size feature vector. We use Eq. (1) to compute each supervoxel's saliency, and by using the category-specific classifiers, each supervoxel $\mathbf{v}^i$ will get M scores, which indicates whether this supervoxel is related to certain action. To make it clear, the codebook used in category-specific classifiers are built from the same action class. We can use either max-pooling or average pooling to decide the category-specific score of

$\mathbf{v}^i$, here in our work, we mix these two by dividing the M scores into L splits. We find the max score in each spilt, and average the sum of L scores. Specifically, we define the scores for $\mathbf{v}^i$ as $S_a(\mathbf{v}^i) = \{R_1^a(\mathbf{v}^i), R_2^a(\mathbf{v}^i), \ldots, R_M^a(\mathbf{v}^i)\}$, then the mix-pooling $S_a^x(\mathbf{v}^i)$ can be calculated as follows.

$$S_a^x(\mathbf{v}^i) = \frac{1}{L}\sum_{j=1}^{L} \max_{l\in[\frac{(j-1)}{L}M+1, \frac{j}{L}M]} (R_l^a(\mathbf{v}^i)) \tag{3}$$

Combining Eq. (1) and Eq. (3), the final saliency of each supervoxel is:

$$S_a^f(\mathbf{v}^i) = \omega_C C_s(\mathbf{v}^i) + \omega_S S_a^x(\mathbf{v}^i) \tag{4}$$

To make the definition clear, we rewrite Eq. (3) as:

$$S_a^f(\mathbf{v}^i) = \omega_I' S_I(\mathbf{v}^i) + \omega_L' S_L(\mathbf{v}^i) + \omega_M' S_M(\mathbf{v}^i) + \omega_S S_a^x(\mathbf{v}^i) \tag{5}$$

Using Eq. (5), we can compute the spatial temporal saliency of each supervoxel $\mathbf{v}^i$ in video V, and this will finally lead to the S-T saliency structure of the whole video.

3. Experiments

3.1 Datasets

We evaluate our proposed approach on two challengine datesets: UCF11 [10] and HMDB51 [1]. UCF11 contains 1168 clips, and are collected in 11 action categories. HMDB51 contains 6849 clips, in 51 action categories. Both of the datasets have frame-wise annotations, which will be used as the ground-truth. We use the train/test splits provided by the author for both datasets.

3.2 Experimental Setup

The proposed approach relies on the extracting of supervoxel and low-level feature. We use the well-known dense trajectories (DT) as our low-level feature [3]. The idea of DT is to track the dense sampled points, each for 15 frames (discard when less than 15 frames). Then, they align the points in the same trajectory across each frame, to build a space-time cuboid. HoG,HoF [16] and MBH [16] features are then extracted from this cuboid.

For the choice of supervoxel, we use a hierarchical manner, namely GBH [17]. Their approach can treat the video as stream of temporal segments and reduce the memory consumption. [17] extracts multiple layers of supervoxel, the number of supervoxels in each layer ranging from 10 to 10k. It's infeasible to use all the layers, it's also time consuming if we choose it manually. Here we use a quick selection to choose the best layer for each action category. We choose the layer which have the maximum intersection-over-union with the ground truth bounding box. This is calculated as the average of each video in the same action category. After this step, each video has 200 to 300 supervoxels.

Table 1 Average accuracies of different pooling approach on two datasets.

	UCF11 [10]	HMDB51 [1]
Baseline	86.6	55.9
CS with MaxPooling	87.5	56.3
CS with AveragePooling	87.6	56.6
CS with MixPooling	87.9	57.1

Table 2 Average accuracies of different saliency measures on two datasets.

	UCF11 [10]	HMDB51 [1]
Baseline	86.6	55.9
IS	86.9	56.3
LS	87.6	58.0
MS	87.9	58.4
LS+MS	88.5	58.6
LS+MS+IS	88.9	58.9
Saliency + CS	90.6	60.1

For computing the low-level feature codebook, we follow the setup of [3]. We randomly sample 100k data points and cluster with k-means to build the codebook of size 5k. For building the supervoxel codebook, we count the trajectories belonging to one supervoxel if over 50% of the trajectory is contained in the supervoxel. The size of supervoxel codebook is 5k as well.

We set K as 50% of the number of supervoxels in the cluster. And for each action category, we set $M = 100, L = 10$. In Eq. (1), $\omega_I = \omega_L = \omega_M = \frac{1}{3}$, in Eq. (4), $\omega_C = \omega_S = \frac{1}{2}$. For each video, we only choose the supervoxels with a score in the highest 50% in the video. We learn a one-vs-all SVM [15] classifier with χ^2 kernel, we find the parameters via a 5-fold cross validation on the training set.

3.3 Evaluation of the Pooling Methods

In this section, we evaluate the 3 pooling approach we mentioned in Sect. 2.2. This is shown in Table 1. We use the raw supervoxel bovw without saliency and category-specific mining as baseline. And compare it with category-specific (CS) mining under 3 pooling approaches.

3.4 Evaluation of the Saliency Maps

In this section, we discover the effectiveness of 3 saliency maps, this is shown in Table 2. IS, LS and MS stand for image saliency, lighting saliency and motion saliency respectively. CS is the short phrase for category-specific mining. The baseline is the same as last section. We also combine the category-specific mining to see the final performance, we only use mix-pooling as it achieves the best results than the other 2 pooling approaches in our work.

3.5 Comparison with State-of-the-Art

We finally compare our proposed approach on both datasets with the state-of-the-art action recognition approaches. We show the classification accuracy in Table 3. [3] is the

Table 3 Comparison of our proposed S-T Saliency to the state-of-the-art on both datasets

Algorithms	UCF11	HMDB51
HoG [3]	74.5	40.2
HoF [3]	72.8	48.9
MBH [3]	83.9	52.1
DT [3]	84.2	54.7
iDT [18], [19]	**90.7**	57.2
Mid-level parts [20]	84.5	37.2
CompactFV [5]	89	54.8
Our best	90.6	**60.1**

original dense trajectory, [18] is the improved dense trajectories which compensated the camera motion by using RANSAC [21] to compute homography and remove the camera motion in each frame, they also use fisher vector [22] for feature encoding, Oneata et.al. [5] extract spatial fisher vectors based on MBH and SIFT descriptors. [20] uses a deformable model to learn space-time parts.

4. Conclusion

In this paper, we propose a novel action recognition approach, namely spatial temporal saliency structure. Compared to the traditional rigid space-time area pooling, the use of supervoxel can better grab the space-time shape in each action video. We use 3 different saliency measurements to cover both the appearance and motion information of the videos. We also design a category-specific mining approach, which can help us find the discriminative supervoxels in each action category. The proposed approach achieves comparative results on 2 well-known action recognition datasets UCF11 and HMDB51.

Acknowledgments

This work was supported by the program for Science and Technology Innovative Research Team for Young Scholars in Sichuan Province, China (No. 2014TD0006).

References

[1] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre, "HMDB: A large video database for human motion recognition," Proc. International Conference on Computer Vision (ICCV), pp.2556–2563, 2011.
[2] I. Laptev, "On space-time interest points," International Journal of Computer Vision, 2005.
[3] H. Wang, A. Kläser, C. Schmid, and C.L. Liu, "Action recognition by dense trajectories," IEEE Conference on Computer Vision & Pattern Recognition, pp.3169–3176, June 2011.
[4] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," 2008 IEEE Conference on Computer Vision and Pattern Recognition, pp.1–8, 2008.
[5] D. Oneata, J. Verbeek, and C. Schmid, "Action and Event Recognition with Fisher Vectors on a Compact Feature Set," 2013 IEEE International Conference on Computer Vision, pp.1817–1824, 2013.
[6] I. Everts, J.C. van Gemert, and T. Gevers, "Evaluation of Color STIPs for Human Action Recognition," 2013 IEEE Conference on Computer Vision and Pattern Recognition, pp.2850–2857, 2013.
[7] E.H. Taralova, F.D. Torre, and M. Hebert, "Motion Words for Videos," Computer Vision – ECCV 2014, Lecture Notes in Computer Science, vol.8689, pp.725–740, Springer International Publishing, Cham, 2014.
[8] H. Jhuang, J. Gall, S. Zuffi, C. Schmid, and M.J. Black, "Towards understanding action recognition," International Conf. on Computer Vision (ICCV), pp.3192–3199, Dec. 2013.
[9] N. Ballas, Y. Yang, Z.-Z. Lan, B. Delezoide, F. Preteux, and A. Hauptmann, "Space-Time Robust Representation for Action Recognition," 2013 IEEE International Conference on Computer Vision, pp.2704–2711, 2013.
[10] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," 2009 IEEE Conference on Computer Vision and Pattern Recognition, pp.1996–2003, 2009.
[11] H. Li and K.N. Ngan, "A Co-Saliency Model of Image Pairs," IEEE Trans. Image Process., vol.20, no.12, pp.3365–3375, 2011.
[12] C.K.J. Harel and P. Perona, "Graph-based visual saliency," Neural Information Processing Systems, 2006.
[13] E. Rahtu, J. Kannala, M. Salo, and J. Heikkilä, "Segmenting Salient Objects from Images and Videos," Computer Vision – ECCV 2010, Lecture Notes in Computer Science, vol.6315, pp.366–379, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.
[14] C. Liu, Beyond pixels: Exploring new representations and applications for motion analysis, Doctoral Thesis, 2009.
[15] C.W. Hsu, C.C. Chang, and C.J. Lin, "A pratical guide to support vector classification," Tech rep, 2005.
[16] N. Dalal and B. Triggs, "Histograms of Oriented Gradients for Human Detection," 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), pp.886–893, 2005.
[17] C. Xu, C. Xiong, and J.J. Corso, "Streaming Hierarchical Video Segmentation," Computer Vision – ECCV 2012, Lecture Notes in Computer Science, vol.7577, pp.626–639, Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.
[18] H. Wang and C. Schmid, "Action Recognition with Improved Trajectories," IEEE International Conference on Computer Vision, pp.3551–3558, Dec. 2013.
[19] X. Peng, C. Zou, Y. Qiao, and Q. Peng, "Action Recognition with Stacked Fisher Vectors," Computer Vision – ECCV 2014, Lecture Notes in Computer Science, vol.8693, pp.581–595, Springer International Publishing, Cham, 2014.
[20] M. Sapienza, F. Cuzzolin, and P.H.S. Torr, "Learning Discriminative Space–Time Action Parts from Weakly Labelled Videos," Int. J. Comput. Vision., vol.110, no.1, pp.30–47, 2014.
[21] M.A. Fischler and R.C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," Commun. ACM, vol.24, no.6, pp.381–395, 1981.
[22] F. Perronnin, J. Sánchez, and T. Mensink, "Improving the Fisher Kernel for Large-Scale Image Classification," Computer Vision – ECCV 2010, Lecture Notes in Computer Science, vol.6314, pp.143–156, Springer Berlin Heidelberg, Berlin, Heidelberg, 2010.