A Dense SURF and Triangulation based Spatio-Temporal Feature for Action Recognition

Do Hang Nga and Keiji Yanai

Department of Informatics, The University of Electro-Communications
Chofu, Tokyo 182-8585 JAPAN
{dohang,yanai}@mm.cs.uec.ac.jp

Abstract. In this paper, we propose a novel method of extracting spatio-temporal features from videos. Given a video, we extract its features according to every set of N frames. The value of N is small enough to guarantee the temporal denseness of our features. For each frame set, we first extract dense SURF keypoints from its first frame. We then select points with the most likely dominant and reliable movements, and consider them as interest points. In the next step, we form triangles of interest points using Delaunay triangulation and track points within each triple through the frame set. We extract one spatio-temporal feature from each triangle based on its shape feature along with the visual features and optical flows of its points. This enables us to extract spatio-temporal features based on groups of related points and their trajectories. Hence the features can be expected to be robust and informative. We apply Fisher Vector encoding to represent videos using the proposed spatio-temporal features. We conduct experiments on several challenging benchmarks, and show the effectiveness of our proposed method.

1 Introduction

For action recognition in a video, both spatio and temporal features are needed to represent actions, while only spatio features such as SIFT and SURF are needed for object or scene recognition on a still image. There exist several well-known features to represent the movements of actors performing the actions such as Histogram of Oriented Optical Flow (HOOF) [1] or trajectory of interest points [2]. In addition to exploiting spatial or temporal features separately, using spatio-temporal (ST) features which integrate both visual and motion characteristics of actions is also preferred among approaches of action recognition in videos. Some ST features have been proposed so far as spatio-temporal extensions of local image descriptors, such as Cuboids [3,4], 3D-SIFT [5], HOG3D [6], and Local Trinary Patterns [7].

To extract local spatio-temporal features from videos, one of the most popular methods is based on Cuboids [3,4]. However, to decide the cuboid size is a tough task. Instead of detecting local Cuboids before extracting features from them, some recent methods tracked interest points in video sequence then leveraged the motion information from their trajectories [2,8]. These approaches obtained good results for action recognition. To track interest points, either tracker based technique or point matching based technique has been employed. Our method proposed in this paper is also based on trajectories of interest points. We apply LDOF [9] to estimate optical flows of all video frames.

In this paper, we propose a dense spatio-temporal feature that is comparable to state-of-the-arts. Empirical results have shown that dense features perform better for complex videos [2,10,11,12]. Our idea is inspired by the work of Noguchi et al. [13]. This method is the baseline we refer in this paper. They proposed to extract spatio-temporal features based on moving SURF keypoints. We address some problems of their method such as their failure in feature extraction of videos containing camera motion or holistic decision of motion threshold in the selection of interest points. We propose simple yet effective solutions to solve these problems. That means, similar to their method, our method is also based on SURF interest points with robust movements, nevertheless how we determine those points is different. Moreover, we propose to improve their feature by exploring more aspects of selected points and introducing several novel spatio-temporal descriptors. The experimental results show significant improvements of our method over the baseline.

According to the success of the BoV (Bag of Visual words) on the research field of image classification, it has became the most popular model for video representation. Noguchi et al also applied BoV model to encode videos. In this paper, we use Fisher Vector to represent videos instead. Fisher vector encoding technique was first applied to image classification task several years ago, shown to extend the BOV representation [14]. The advantage of this technique has been demonstrated that it is not limited to the number of occurrences of each visual word but it also encodes additional information about the distribution of the descriptors. The methodology is that, (1) extracting local features from images, (2) modelling the distribution of those features as mixtures of Gaussian (GMM), training a soft codebook, (3) applying Fisher kernels on obtained codebook to encode each image as a Fisher Vector. Recently, some works on action recognition have also employed this approach to encode videos and showed the effectiveness of Fisher Vector encoding over the traditional BoV [15]. In this paper, we apply Fisher encoding technique as described in [14].

We aim to recognize human actions in realistic videos where the background is complex. We conduct our experiments on large-scale benchmark datasets for action recognition such as UCF-50, HMDB51 adn UCF-101 to validate our proposed method. These datasets are not only large-scale but also challenging benchmarks for human action recognition in realistic and challenging settings such as large variations in camera motion, object appearance and pose, viewpoint and complicated background.

Our contribution is three-fold: first, a method of extracting spatio-temporal features which is comparable to the state of the arts, second, a simple yet efficient selection of interest points, and finally, novel descriptorization of spatio-temporal features. The reminder of this paper is organized as follows: Section 2 discusses more about some related works. In Section 3, we overview the proposed method, and In Section 4 the method to extract the ST proposed feature is described in detail. Section 5 explains about conducted experiments on three large-scale action recognition datasets, and presents the results. Conclusions are presented in Section 6.

2 Related work

So far, in the research field of action recognition, local spatio-temporal features has been explored to represent videos. Many methods of extraction and descriptorization of spatio-temporal features [3,16,12,4] have been proposed over the past few years. To determine space-time regions (called as Cuboids) where features are extracted, Dollar et al. [3] proposed to apply 2-D Gaussian kernels to the spatial space and 1-D Gabor filters to the temporal direction. Laptev et al. [16] proposed an extended Harris detector to extract Cuboids. As another method other than using Cuboids, extracting local features based on trajectories of interest points also showed good results for action recognition [2,8,15]. Matikainen et al. [17] proposed to extract trajectories using a standard KLT tracker, cluster the trajectories, and compute an affine transformation matrix for each cluster center.

In this paper, we propose to improve method of extracting ST feature proposed by Noguchi et al. [13]. Following [13], we also extract features based on moving SURF points and use Delaunay triangulation to model the spatial relationships between interest points. We address some problems of their method such as the inability to handle camera motion or holistic decision of motion thresholds for selecting points which cause failure in extracting features of some videos. We propose to solve these problems by simple yet efficient methods of motion compensation and point selection.

As treatment for camera motion, Cinbis et al. [18] applied video stabilization using homography-based motion compensation approach. They estimated camera flow by calculating the homography between consecutive frames and compensate optical flow of points by removing estimated camera flow. Similarly, Jain et al. [8] also removed camera motion from original optical flow, nevertheless they consider affine motion as camera motion. Wu et al. [19] decomposed Lagrangian particle trajectories into camera-induced and object-induced components for videos acquired by a moving camera. In. [2], Heng Wang et al. did not compensate camera motion in advance but employed motion boundary histograms which already have constant motion removed. We also reduce the influence of any existed camera motion by canceling the constant motion. Our proposed method of motion compensation improves significantly performance of feature extraction over the baseline since it helps not only extract features in case that camera motion exists but also detect more robust interest points.

3 Overview of the spatio-temporal proposed feature

Our spatio-temporal feature is based on dense SURF keypoints with dominant and reliable movements. Here dominant and reliable points are supposed to belong to the human who is performing the action. Our feature investigates the shape of triangles produced by applying Delaunay triangulation on those informative points along with the movements (flow vectors) of points within the same Delaunay triple. We show that concatenating these features with SURF features of interest points can form a powerful spatio-temporal feature.

Our proposed feature is dense in both space and time. We extract features with temporal step size of N frames. We set the value of N to be small so that the extracted features are temporally dense. We operate tracking of interest points through N frames. As trajectories may drift from their precise locations during the tracking process, limiting the tracking process within short duration like this is supposed to be able to overcome this problem. In our experiments, we fix N as 5. The process of extracting our proposed spatio-temporal feature from each frame set is summarized as follows:

1. Extract dense SURF keypoints of the first frame using Dense SURF [20].
2. Compute optical flows from k^{th} frame ($k = 1, 2, ..., N - 1$) to the next frame ($k + 1^{th}$ frame) using LDOF [9].

3. Estimate camera motion in each frame and compensate motion if camera motion detected (Section 3.2).
4. Select points which are expected being more informative than the others (Section 3.3) and from triangles of selected points using Delaunay triangulation.
5. Extract ST features from each triangle based on its shape along with motion features of its points through the frame set (Section 3.4).

The main improvements of our method over our baseline can be summarized as follows: (1) treatment of camera motion, (2) selection of interest points and, (3) enhancement on descriptorization of ST features. We explain in details these improvements in following subsections.

4 The detail of the proposed method

4.1 Detection and compensation of camera motion

In Noguchi et al.'s work [13], once camera motion has been detected in a frame set, obtained information would be considered as noise, thus no points would be selected. Consequently, no features are extracted if the whole video contains camera motion. We propose to solve this problem by following a simple 2-step technique:

1. Step 1: Confirm the existence of camera motion based on optical flows of SURF keypoints. If detecting camera motion, determine the direction and magnitude of camera motion before going to the next step.
2. Step 2: Compensate motion by canceling camera motion from original flows of SURF keypoints.

Detection of camera motion: At the first step, we aim to find out at each frame how the camera move in both horizontal direction (forward or backward) and vertical direction (up or down). This step is based on our assumption that if most points move toward the same direction, camera motion exists. Let denote P^{x^+} and P^{x^-} as number of points with positive and negative optical flows respectively, $P_m^{x^+}$ and $P_m^{x^-}$ as number of moving points which shift forward and backward respectively, so that we suppose that camera is moving forward if Eq.1 and Eq.2 are satisfied or backward if Eq.3 and Eq.4 are satisfied:

$$P_m^{x^+} \geq kP^{x^+} \tag{1}$$

$$P_m^{x^+} > P_m^{x^-} \tag{2}$$

$$P_m^{x^-} \geq kP^{x^-} \tag{3}$$

$$P_m^{x^-} > P_m^{x^+} \tag{4}$$

Here, k is a fraction threshold representing minimal required proportion of moving points over all points with the same direction. In our experiments, we set k as $\frac{2}{3}$. A point is considered as moving points if its absolute optical flow is larger than or equal to 1.The camera is supposed as horizontally stable if none of above condition is satisfied. If the camera is detected as being moved, camera motion is calculated as average of

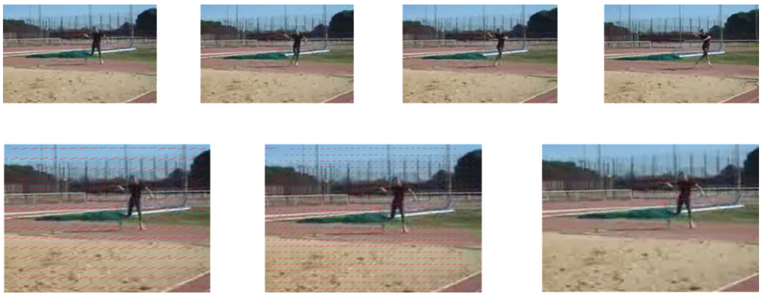

Fig. 1. An example that shows efficiency of proposed method of reducing camera motion and selecting interest points. The first row presents a frame set of consecutive frames which contains camera motion. In this case, camera is moving (to the right), thus interest points are not detected according to the baseline. The most left figure of the second row shows optical flows of extracted dense SURF keypoints before the camera motion compensated. The middle figure and the right figure of the second row respectively present points determined as moving points by the baseline (with fixed motion threshold) and our method (with flexible threshold). Point selection of the both are performed after compensating camera motion. This example shows that our method is not only able to reduce the effect of camera motion but also to select more representative interest points than the baseline.

absolute optical flows of points which moved to the same direction as camera. Camera motion for vertical direction is estimated in the similar manner.

Compensation of camera motion: Flow of each SURF keypoint is compensated simply as follows:

$$f_i = f_i - df_{camera} \tag{5}$$

Here, f_i refers to flow of point i, f_{camera} refers to camera flow. d equals 1 if camera moved to positive direction or -1 if camera moved to negative direction. f_{camera} is measured separately for all considered directions (forward, backward, up and down) and compensation is operated in each of those directions. By our manner, camera motion can be compensated in most cases except for zooming. Handling this case of camera motion is one of our future works. See Fig.1 for an example result of our motion compensation method.

4.2 Selection of interest points

According to the baseline, selection of interest points is based on their optical flows between the first frame and the middle frame of the frame set. A point is believed as an interest point if its flow is larger than the pre-defined motion threshold. As a result, in case of no significant movement than the threshold from the first to the middle frame, no feature can be extracted. Moreover, their motion threshold is determined in holistic manner and fixed for every video of every type of actions. However, due to camera motion, video resolution, movements of background objects, and especially the diversity of actions as well as actors, points selected based on a constant motion threshold may not always be representative. For example, even though that significant movements are expected to be caused mainly by the actor, in the background there may be objects which move dominantly at several frames. Hence, the points belong to these objects may be imprecisely taken as interest points. In addition, magnitude of movement may vary largely from action to action. For instance, sport activities such as jumping trampoline or swimming are supposed to cause large displacements. On the other hand, daily activities such as drinking or talking in general generate smaller optical flows. We demonstrate that in order to overcome these problems, motion threshold should be flexible.

We propose to determine motion threshold flexibly and select as reliable moving points as possible. The idea is that robustness of a point should be compared to its fellows at the same time rather than to a fixed threshold. In our method, motion threshold is estimated for every frame in all directions based on flows of its SURF points. The following equation represents how we calculate motion threshold for a frame in forward direction (x^+). Thresholds for the remaining directions are similarly calculated.

$$thresh_{f_{x^+}} = aver_{f_{x^+}} + \alpha(max_{f_{x^+}} - aver_{f_{x^+}}) \tag{6}$$

Here, $thresh_{f_{x^+}}$ means the motion threshold for frame f in x^+ direction. $aver_{f_{x^+}}$ and $max_{f_{x^+}}$ respectively refer to the average and the maximal flow magnitude at frame f in x^+ direction. The qualification that a point should satisfy to be considered as a moving point is that in at least one of four considered directions, its flow magnitude is somewhat greater than the average flow of that direction. The constant α controls that qualification. In our experiments, we set α as 0.5. Thus, the motion threshold is near to the median of the average and the max flows. However, in some case, at some frames, all objects including actor stay still, thus it is not necessary that there always must be moving points. We suppose that nothing in a frame moved if all of its thresholds are smaller than 1.

After determining which points are moving points through the frame set, instead of simply taking all points which ever moved like Noguchi et al, we aim to select as many representative points as possible. We postulate a hypothesis that points with more movements are more reliable and informative. For example, through the whole frame set, points moved 2 times are expected to be more reliable as well as representative than points moved only once. Based on this hypothesis, we propose to select points greedily based on number of times they moved through the frame set. Our algorithm of point selection is described as follows:

```
**(Algorithm for selecting interest points.)**
$M$ = maximal number of movements $(M \leq N - 1)$
$T$ = total number of moving points
$GS$ = group of selected points (initialized as empty)
for $i = M$ to 1 do
    $GS$ = $|GS$, points moved i times $|$
    if $|GS| \geq \beta T$ then
        break;
    end if
end for
```

Following the above algorithm, the group of selected points is only a proportion of moving points but expected to consist of most representative points. In our experiments, we set β as $\frac{1}{2}$. Fig.1 shows the effectiveness of our method of selecting interest points over the baseline.

4.3 Descriptorization of ST features

After selecting interest points, following the baseline, we apply Delaunay triangulation to form triples of them. One ST feature can be obtained from each triple. Our proposed feature extracted from a triple is constructed based on following descriptors. We classify them to *spatial descriptor* which represents static visual features of points, *temporal descriptor* which presents movements of points through the frame set and *spatio-temporal descriptor* which characterizes trajectory-based visual features of points or group of points. Below we describe in detail each descriptor.

Spatial Descriptors. To form spatial descriptor, we combine SURF descriptors of three points of the triple at the first frame. Each SURF is extracted with subregions of 3 by 3 pixels, Haar filters of 4 by 4 pixels and 4 subregions for SURF. Thus we obtain 64-dimension SURF descriptor for each point [20]. However, concatenating three SURF descriptors of three points forms a descriptor which is high-dimensional $(3 \times 64 = 192)$but may consist of repeated information. Thus, we apply PCA on this descriptor to acquire a lower dimensional but more representative one, as well as reduce computational cost. We denote this dimension reduced descriptor as PSURF. In our experiments, PSURF is a 96-dimension vector.

Temporal Descriptors. We propose to extract following 2 temporal features:

(1) **A histogram of Optical Flow (HOOF)**. $3(N-1)$ flow vectors of 3 points are binned to B_o-bin histogram. Following [1], each flow vector is binned according to its primary angle from the horizontal axis and weighted according to its magnitude. That means, a flow vector $v = [x, y]$ with its angle $\theta = tan^{-1}(y\frac{x}{})$ in the range:

$$-\frac{\pi}{2} + \pi\frac{b-1}{B_o} \leq \theta < -\frac{\pi}{2} + \pi\frac{b}{B_o} \tag{7}$$

will contribute by $\sqrt{x^2+y^2}$ to the sum in bin b. Finally, the histogram is normalized to sum up to 1.

(2) **A Histogram of Direction of Flows (HDF)**. Following Noguchi et al., we binned flow vectors of 3 points within the triple according to their direction. However, in [noguchi], one histogram is calculated for each point, thus many bins become zeros. This makes the motion descriptor have small effectiveness on action discrimination. Here we propose to bin all flow vectors ($3(N-1)$ flows) into 4 bins: $x^+(|f_{x^+}| \geq 0), x^-(f_{x^-} \geq 0), y^+(f_{y^+} \geq 0), y^-(f_{y^-} \geq 0)$. Similarly to HOOF, this histogram is also weighted by flow magnitude and normalized to sum up to 1.

Spatio-temporal Descriptors. We propose to generate the following 3 descriptors. The first two represent visual characteristics of triangles through the frame set. The last one descriptorizes the shape of trajectories. The last two are newly introduced by us. Refer to Fig.2 for illustration of these proposed two features.

(1) **Areas of Triangle (AT)**: Following Noguchi et al.' work, the areas of the triangle at all frames are calculated then concatenated and normalized to form a N-dimension descriptor.

(2) **A Histogram of Angles of Triangle (HAT)**. To better explore the shape characteristics of obtained triangles, we propose to investigate their angles by binning them based on their magnitude. Here, we consider only two angles since given the degrees of any two out of three angles, it is sufficient to characterize the shape of a triangle. Using two optional angles is not preferred here since they may be not representative for their

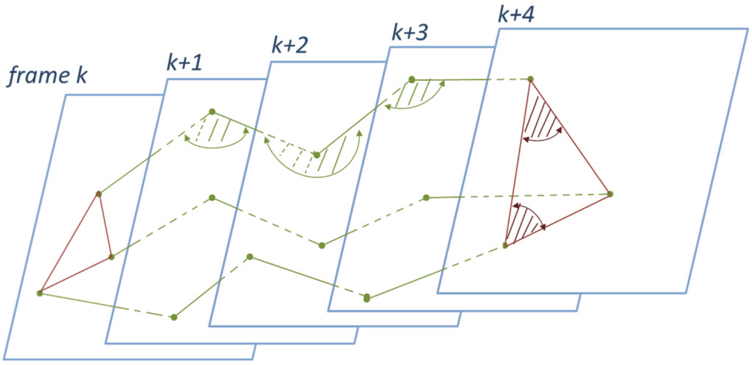

Fig. 2. Illustration of proposed spatio-temporal features. We additionally explore characteristics of interest points by exploiting angles of triangles formed by them (red ones) and angles shaped by consecutive trajectories of them (green ones). We show here an example of trajectories of grouped interest points in a frame set of 5 frames. 2×5 smallest angles of triangles are binned to obtain a HAT and 3×3 trajectory based angles are binned to obtain a HAF following proposed method described in Section 3.4.

triangle. Thus, one can consider use two largest or two smallest angles. However, two largest angles can range from 0^o to 180^o while two smallest angles range only from 0^o to 90^o. Hence, we select two smallest angles since binning them is expected to be more efficient and easier to define histogram bin. Moreover, it can not happen that both of two smallest angles are larger than 60^o. Based on this observation, we set up histogram bin as follows: for $\theta > 60^o$, the histogram bin is of size 30, otherwise, the histogram bin is of size 15. In this manner, $2 \times N$ smallest angles are binned to 5 bins: [0-15], [15-30], [30-45], [45-60], [60-90]. Each angle is weighted by sum of magnitude of its two edges and normalized at the end.

A Histogram of Angles of Flows (HAF). To exploit trajectories of interest points for modelling the action, some work straightly employ them as descriptors [2]. However, this approach suffers from the problem that trajectories may vary largely due to the velocity of the actor. To reduce the effect of variety of velocity, we propose to extract features based on angles shaped by trajectories. These angles are supposed to be more informative than trajectories themselves(See Fig3). The angles are binned by the same method as shown in Equ. 7. Number of histogram bin for HAF is denoted as B_a.

Finally all above descriptors are concatenated to form our ST feature which has 96 (PSURF) + B_o (HOOF) + 4 (HDF) + N (AT) + 5 (HAT) + B_a (HAF) = 105+B_o+N+B_a dimension. In our experiment, we set $N = 5$, $B_o = 6$ and $B_a = 4$, thus we obtain a 120-dimension ST descriptor.

5 Experiments and Results

We made experiments on large-scale benchmark datasets for three large-scale action recognition datasets including UCF-50 [1], HMDB51 [21] and UCF-101 [2] to validate

[1] http://crcv.ucf.edu/data/UCF50.php

[2] http://crcv.ucf.edu/data/UCF101.php

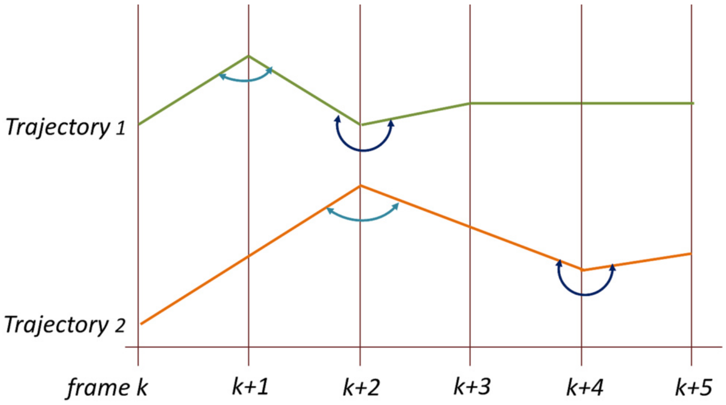

Fig. 3. An example that illustrates the effect of variety in velocity on action recognition and the efficiency of our proposed method. We show trajectories of points which belong to two actors performing the same action in 6 consecutive frames. We assume that the actors move in similar way but at different speed. As shown here, Trajectory 1 which corresponds to faster actor and Trajectory 2 which belongs to lower actor only match at first, thus trajectory based descriptors become nearly totally different. On the other hand, according to our method, exploiting angles shaped by trajectories help to find out more the similarity between these two trajectories. The similar angles (marked by same color) can be binned to the same bin, hence this angle based descriptor can be expected to reduce the effect of diversity in velocity.

our proposed method. These datasets are not only large-scale but also challenging benchmarks for human action recognition in realistic and challenging settings such as large variations in camera motion, object appearance and pose, viewpoint and complicated background. The above three datasets consists of Web video clips collected from YouTube manually. Each of them has 50, 51 and 101 action categories, respectively.

To validate the enhancement of proposed feature over the baseline, we conduct experiments with ST feature proposed by the baseline and our proposed feature. We use Fisher Vector encoding as described in [14] to represent videos. We train multi-class SVMs [22] to perform multi-class action classification. For comparison, we made experiments with our proposed features coded by the conventional bag-of-visual-words (BoV) as well as the proposed features coded by Fisher Vector (FV).

UCF50 As evaluation criteria for UCF50, we follow method as suggested by the author [23], that is “Leave One Group Out Cross Validation” which will lead to 25 cross-validations. The videos belonging to the same group are kept being separated in training and testing, since the videos in a group are obtained from single long video, sharing videos from same group in training and testing sets would give high performance. We compared our proposed methods with other state-of-the-arts [24,25,**?**,23]. The results are shown in Fig.5.

Our proposed methods was ranked in the second in this ranking. The top result was [23]. Note that [23] fused several kinds of features, while we used only one kinds of the proposed ST feature and it it possible to fuse other kinds of features to boost performance.

Table 1. Experimental results on UCF-50.

Method	Average Precision (AP)
HOG/HOF [24]	47.9 %
Our (BoV)	51.1%
Action Bank [25]	57.9 %
Motion Interchange [26]	68.51 %
Our (Fisher)	**71.6%**
Reddy and Shah [23]	76.9%

HMDB51 As for HMDB51, we use three distinct training and testing splits generated by its authors. For each action category, a set of 70 training and 30 testing clips was selected so that they fulfill the 70/30 balance for each meta tag, and clips from the same video were not used for both training and testing. HMDB is more diverse and difficult dataset than UCF-50 and UCF-101. As shown in Fig.2, the state-of-the-art are still less than 30%, while the top result exceeded 70% for UCF-50. As results, our proposed methods outperformed other state-of-the-arts.

Table 2. Experimental results on HMDB-51.

Method	Average Precision (AP)
Our (BoV)	17.3%
HOG/HOF [24]	20.44 %
Action Bank [25]	26.9 %
Motion Interchange [26]	29.17%
Our (Fisher)	**29.5%**

UCF101 We follow evaluation set up as suggested in the action workshop page [3]. We adopt the provided three standard train/test splits to evaluate our results. In each split, clips from 7 of the 25 groups are used as test samples, and the rest for training. The result of each experiment reported here is calculated as the mean of average accuracies over the three provided test splits. We compare our results to the result reported in [27].

According to [27], their result is provided by baseline approach of action recognition [27] using standard BoV. The results are shown in Table 1. Note that [27] follows the old experimental set up, that is “Leave One Group Out Cross Validation”.

Although UCF101 is the largest action dataset up to date, there are too few reported results for it. As demonstrated in [13], their method of extracting ST features is comparable to state-of-the-arts. By applying Fisher Vector encoding, their method obtains slightly better results. We propose to improve their method and significantly boost the overall precision as shown in Table 1. The results demonstrate that, our method could select more representative points as well as explore better the visual characteristics of them.

[3] http://crcv.ucf.edu/ICCV13-Action-Workshop/

Table 3. Experimental results on UCF-101.

Method	Average Precision (AP)
[13] (Fisher)	38.2%
Our (BoV)	40.1%
[27] (BoV)	44.5%
Our (Fisher)	**60.1%**

6 Conclusions

In this paper, we propose a novel method of extracting spatio-temporal features which is able to efficiently select interest points and descriptorize their features. The proposed methods consists of dense-sampling of SURF and Delaunay triangulation of selected interest points, the idea of which is based on our previous work [13].

The experimental results show significant improvement of our method over the baseline regarding the three up-to-date large-scale action recognition datasets: UCF-50, HMDB51, and UCF-101. In future work, we will introduce detailed analysis on human-object interaction for action recognition, and recent approach of compensating camera motion in order to handle more complicated cases such as zooming.

References

1. R. Chaudhry, A. Ravichandran, G. Hager, and R. Vidal. Histograms of oriented optical flow and binet-cauchy kernels on nonlinear dynamical systems for the recognition of human actions. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1932–1939, 2009.
2. H. Wang, A. Klaser, C. Schmid, and C-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *International Journal of Computer Vision*, 103(1):60–79, 2013.
3. P. Dollar, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. of Surveillance and Performance Evaluation of Tracking and Surveillance*, pages 65–72, 2005.
4. I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.
5. P. Scovanner, S. Ali, and M. Shah. A 3-dimensional sift descriptor and its application to action recognition. In *Proc. of ACM International Conference Multimedia*, pages 357–360, 2007.
6. A. Kläser, M. Marszałek, and C. Schmid. A spatio-temporal descriptor based on 3d-gradients. In *Proc. of British Machine Vision Conference*, pages 995–1004, 2008.
7. L. Yeffet and L. Wolf. Local trinary patterns for human action recognition. In *Proc. of IEEE International Conference on Computer Vision*, pages 492–497, 2009.
8. M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2013.
9. T. Brox, C. Bregler, and J. Malik. Large displacement optical flow. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 41–48, 2009.
10. F. V. Jensen, H. I. Christensen, and J. Nielsen. Bayesian methods for interpretation and control in multi-agent vision systems. In *Proc. of SPIE 1708, Applications of Artificial Intelligence X: Machine Vision and Robotics*, pages 536–548, 1994.

11. E. Nowak, F. Jurie, W. Triggs, and M. Vision. Sampling strategies for bag-of-features image classification. In *Proc. of European Conference on Computer Vision*, pages IV:490–503, 2006.
12. G. Willems, T. Tuytelaars, and L.V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proc. of European Conference on Computer Vision*, pages 650–663, 2008.
13. A. Noguchi and K. Yanai. A surf-based spatio-temporal feature for feature-fusion-based action recognition. In *ECCV WS on Human Motion: Understanding, Modeling, Capture and Animation*, 2010.
14. F. Perronnin and C. Dance. Fisher kernels on visual vocabularies for image categorization. In *Proc. of IEEE Computer Vision and Pattern Recognition*, pages 1–8, 2007.
15. I. Atmosukarto, B. Ghanem, and N. Ahuja. Trajectory-based fisher kernel representation for action recognition in videos. In *Proc. of IAPR International Conference on Pattern Recognition*, pages 3333–3336, 2012.
16. I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2003.
17. P. Matikainen, M. Hebert, and R. Sukthankar. Trajectons: Action recognition through the motion analysis of tracked features. In *ICCV Workshop on Video-Oriented Object and Event Classification*, 2009.
18. N. I. Cinbis and S. Sclaroff. Object, scene and actions: Combining multiple features for human action recognition. In *Proc. of European Conference on Computer Vision*, pages 494–507, 2010.
19. W. Shandong, O. Omar, and S. Mubarak. Action recognition in videos acquired by a moving camera using motion decomposition of lagrangian particle trajectories. In *Proc. of IEEE International Conference on Computer Vision*, pages 1419–1426, 2011.
20. J.R.R. Uijlings, A.W.M. Smeulders, and R.J.H. Scha. Real-time visual concept classification. *IEEE Transactions on Multimedia*, 2010.
21. H. A. Jhuang, H. A. Garrote, E. A. Poggio, T. A. Serre, and T. . Hmdb: A large video database for human motion recognition. In *Proc. of IEEE International Conference on Computer Vision*, 2011.
22. I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *The Journal of Machine Learning Research*, 6:1453–1484, 2005.
23. K. K. Reddy and M. Shar. Recognizing 50 human action categories of web videos. *Machine Vision and Applications*, 24:971–981.
24. I. Laptev, A. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2008.
25. S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *Proc. of IEEE Computer Vision and Pattern Recognition*, 2012.
26. O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *Proc. of European Conference on Computer Vision*. 2012.
27. S. Khurram, R. Z. Amir, and M. Shar. UCF101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, abs/1212.0402, 2012.