A SURF-based Spatio-Temporal Feature for Feature-fusion-based Action Recognition

Akitsugu Noguchi and Keiji Yanai

Department of Computer Science,
The University of Electro-Communications,
1-5-1 Chofugaoka, Chofu-shi, Tokyo, 182-8585, Japan
noguchi-a@mm.cs.uec.ac.jp, yanai@cs.uec.ac.jp

Abstract. In this paper, we propose a novel spatio-temporal feature which is useful for feature-fusion-based action recognition with Multiple Kernel Learning (MKL). The proposed spatio-temporal feature is based on moving SURF interest points grouped by Delaunay triangulation and on their motion over time. Since this local spatio-temporal feature has different characteristics from holistic appearance features and motion features, it can boost action recognition performance for both controlled videos such as the KTH dataset and uncontrolled videos such as Youtube datasets, by combining it with visual and motion features with MKL. In the experiments, we evaluate our method using KTH dataset, and Youtube dataset. As a result, we obtain 94.5% as a classification rate for in KTH dataset which is almost equivalent to state-of-art, and 80.4% for Youtube dataset which outperforms state-of-the-art greatly.

1 Introduction

Recently, the number of videos on the Web is increasing rapidly. To organize them, content-based video analysis has become important. For example, video summarization and content-based video retrieval help users to find videos which they want to watch efficiently. However, most of the existing works on action recognition have focused on controlled videos such as ones taken by a fixed camera so far, only a few works on action recognition have focused on uncontrolled videos such as ones on video sharing Web sites.

In this paper, we propose a novel spatio-temporal feature which is useful for feature-fusion-based action recognition with Multiple Kernel Learning (MKL). The proposed spatio-temporal (ST) feature is based on moving SURF interest points grouped by Delaunay triangulation and on their motion over time. Since this local spatio-temporal feature has different characteristics from holistic appearance features and motion features, it can boost action recognition performance for both controlled videos such as the KTH dataset and uncontrolled videos such as Youtube datasets, by combining it with visual and motion features with MKL. For feature fusion with MKL, we use Gabor texture features for appearance features, and global optical-flow histograms as motion features. As representation of features, we employ Bag-of-Frame (BoFr) which is dense temporal sampling within a shot. For both Gabor features and optical-flow features, we extracted them from all the frames within a shot, vector-quantized them after aggregating all of them, and built a BoFr vector for each shot.

In the experiments, we made two kinds of experiments. One is action recognition with the KTH dataset [1] for evaluation on the proposed method, and the other is Web video shot classification which is one of possible applications of the proposed method with the Youtube data. The KTH action video dataset is a standard controlled-video dataset which of videos are taken by a fixed camera with uniform backgrounds, while the Youtube datasets we used in this paper includes the Youtube dataset used in [2] and in-house dataset collected from Youtube by ourselves. As a result, we obtain 94.5% as a classification rate for in the KTH dataset which is almost equivalent to state-of-art, and 80.4% for the Youtube dataset in [2] which outperforms state-of-the-art greatly.

In this paper, units to be classified are shots which are generated from videos by dividing them at the shot boundaries. This shot division is carried out in advance as a pre-processing.

In the rest of this paper, we describe related work in Section 2. Then in Section 3, we explain a novel spatio-temporal feature which is the main contribution of this paper. In Section 4, we propose feature fusion of holistic appearance and motion features as well as proposed spatio-temporal features by Multiple Kernel Learning. Section 5 describes the experimental results. Finally we conclude this paper in Section 6.

2 Related Work

Recently, a spatio-temporal feature has drawn attention for human action recognition and content-based video analysis. As a method to extract spatio-temporal features, several methods to extend two dimensional features to the temporal dimension were proposed. The most representative method is "cuboids" where many local cubic spatio-temporal regions are extracted. Dollár et al. proposed a method to detect local cuboids by applying 2-D Gaussian kernels in the spatial space and 1-D Gabor filters in the temporal direction [3], and they generated video visual words by vector-quantizing local cuboids in the same way as bag-of-visual-words for object recognition. Laptev et al. proposed [4] an extended Harris detector to extract cuboids. Laptev et al. and Dollar et al. extracted Histogram of Gradient (HoG) and Histogram of Flow (HoF) from detected cuboids to represent them, while Kläser et al. proposed three-dimensional HoG to represent cuboids [5]. As another method than cuboids, Kobayashi et al. proposed Cubic Higher order Local Auto-Correlation (CHLAC) which performs well in surveillance field [6].

Although "cuboid" representation enables us to handle action recognition for videos in the same way as object recognition for still images, computational costs to extract cuboid features by the methods described above are relatively high. Moreover, it is difficult to decide the appropriate size of cuboids. To overcome these problems, in this paper, we propose a novel spatio-temporal feature employing SURF features [7] and Lucas-Kanade optical flow detection methods [8] both of which are very fast detectors. Since we do not use cuboids, the proposed method is more simple, fast and efficient to extract spatio-temporal features than the existing ones. Therefore, the proposed feature is suitable for a large amount of Web video data.

Many researchers have studied about action recognition. Gilbert et al. proposed the method based on very dense corner features, and can classify in real time [9]. Uemura et al. used a motion model based in optical flow combined with SIFT feature correlation [10]. Kim et al proposed an extended Canonical Correlation Analysis (CCA) for action recognition [11].

Recently, multiple feature fusion by Multiple Kernel Learning (MKL) was proved to be effective for object recognition [12]. MKL-based fusion is also applied to action recognition. Sun et al. proposed using MKL for action recognition to select several useful features from 68 kinds of trajectory-based features [13]. Han et al. also employed MKL to combine more than 30 object-part-based features [14]. Both of papers focused on selecting features from many ones with MKL, while we use MKL to fuse only three features for weighting.

There are a few researches on action recognition for unconstrained videos such as Web videos. Cinbis et al. proposed a method to learn action automatically from Web, and recognize action [15]. In this work, only static features are used as an action descriptor. On the other hand, in this paper, we consider not only appearance features but also motion features. Liu et al. proposed the action recognition method on Web by combining spatio-temporal features and appearance features based on AdaBoost [2]. Liu et al. utilized cuboid features proposed in [3] as a spatio-temporal feature, and SIFT as an appearance feature.

3 Spatio-Temporal Feature Extraction

In this paper, we propose a novel spatio-temporal (ST) feature which is based on the SURF (Speeded-Up Robust Feature) features [7] and on optical flows detected by the Lucas-Kanade method [8].

For designing a new ST feature, we set the premise that we combine it with holistic appearance features and motion features by Multiple Kernel Learning (MKL). Therefore, the important thing is that it has different characteristics from two kinds of holistic features.

Following this premise, we extend the method proposed in [16]. In the original method, we detect interest points and extract feature vectors employing the SURF method [7], and then we select moving interest points employing the Lucas-Kanade method [8]. In the original and proposed method, we use only moving interest points where ST features are extracted and discard static interest points, because we expect that it is a local feature which represents how objects in a video are moving. In addition to the original method, we newly introduce Delaunay triangulation to form triples of interest points where both local appearance and motion features are extracted. This extension enables us to extract ST features not from one point but from a triangle surface patch, which makes the feature more robust and informative. The characteristic taken over from the original method [16] is that it is much faster than the other ST features such as cuboid-based features, since it employs SURF [7] and the Lucas-Kanade method [8], both of which are known as very fast detectors.

Table 3 shows the algorithm to extract the proposed spatio-temporal feature. We explain the detail in this section.

[Step 1] Extract SURF points and descriptors We apply the SURF method [7] to detect interest points and extract SURF descriptors for the de-

Table 1. Flow of extracting the proposed spatio-temporal feature.

step 1	: Extract SURF points and descriptors
step 2	: Select moving points
step 3	: Apply Delaunay triangulation
step 4	: Extract local motion features with normalization of the directions of motions
step 5	: Concatenate SURF descriptors and motion vectors regarding each triangle

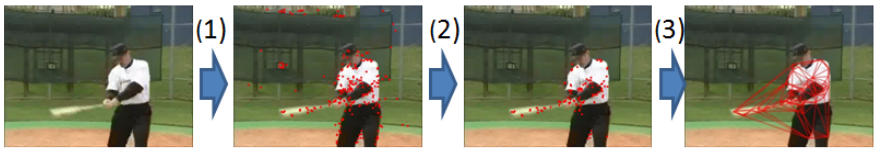

Fig. 1. The transition from Step 1 to Step 3. (1) detected SURF points, (2) detected SURF points with motion, and (3) obtained Delaunay triangles.

tected interest points from the frame images which are extracted from a given video shot at every N frames. Extracted SURF descriptors represent local appearances around interest points. The second image from the left in Figure 1 shows an example of detected SURF points. In the experiments, we set N as 5.

[Step 2] Select moving points We calculate motion vectors at each interest point by the Lukas-Kanade optical flow detector [8], and select moving points, because we like to extract ST features only from moving objects. We calculate optical flows between the first frame and the $\lfloor N/2 \rfloor$-th frame in the N-frame unit as shown in the left side of Figure 2 for selecting interest points.

[Step 3] Apply Delaunay triangulation We apply the Delaunay triangulation to make triplets of moving interest points. This enables us to extract ST features from not independent points but groups of neighboring points. A triplet of three points which form a Delaunay triangle is a spatial unit where the proposed ST features are extracted. Figure 1 shows the transition of a sample image from Step 1 to Step 3.

[Step 4] Extract local motion features with normalization of the directions of motion vectors We extract optical flows with the Lucas-Kanade method from $(M-1)$ intervals among the N frames which is a temporal unit from which a ST feature are extracted, after picking up M frames out of N frames (M should be a factor of N) as shown in the right side of Figure 2. We calculate optical flows from $(M-1)$ consecutive intervals at each moving point in order to consider consecutiveness of motions. To track each interest point, we use optical flows detected by the Lucas-Kanade method. In case that M is 2, we extract rough motions. On the other hand, in case that M equals to N, motion information becomes condensed. In the experiment, we set both N and M as 5.

As representation of local motion features, we generate a 5-dim vector for each interval of each moving interest point from the motion matrix estimated by the Lucas-Kanade method. The 5-dim vector consists of x^+, x^-, y^+, y^- and no optical flow x^0, where x^+ means the degree of the positive elements along x-axis and x^- means the degree of the negative elements along x-axis. The

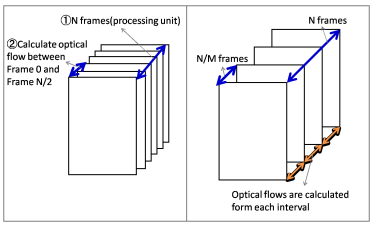

Fig. 2. Reference frames of optical flow calculation for selecting moving points (left), and reference frames for calculating local motion features (right)

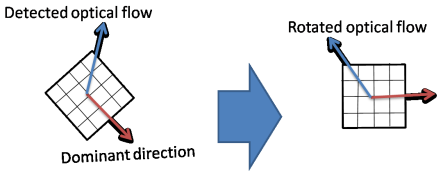

Fig. 3. Normalizing the direction of an optical flow rotating it based on the dominant direction detected by the SURF detector.

motion feature for each interval is normalized so that the summation of all the elements equals to 1. We concatenate all the 5-dim vectors extracted from $(M-1)$ intervals into one motion vector for each moving point, and totally the dimension of motion feature becomes $(M-1) \times 5$.

We hope that this feature is invariant to rotation, since we combine this ST feature with a holistic motion feature which is not invariant to rotation. The same feature should be extracted from "walk to right" and "walk to left", since we intend to design ST features to categorize actions ignoring the directions of actions. Actually, Noguchi et al. [16] showed introducing rotation-invariance into ST features improved the KTH performance. To this end, we rotate optical flows along the dominant direction of visual features to normalize their direction. Figure 3 shows the rotation of an optical flow.

The rotated optical flow vector (x, y) are represented as follows:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} cos\theta & -sin\theta \\ sin\theta & cos\theta \end{bmatrix} \begin{bmatrix} x_0 \\ y_0 \end{bmatrix} \tag{1}$$

where (x_0, y_0) is the original optical flow vector for a moving point, and θ is the dominant direction of the SURF descriptor at the same point.

[Step 5] Concatenate SURF descriptors and normalized motion vectors regarding each triangle In the final step, we generate a ST feature vector by combining SURF features and normalized motion features regarding each Delaunay triangle.

As local appearance features, we utilize SURF descriptors of the three points which form a Delaunay triangle. The SURF descriptors of the three points are concatenated in the descending order of their SURF scale values. Since the SURF descriptor is 64 dimensions, the dimension of the appearance feature is $64 \times 3 = 192$.

As local motion features for a triangle, we concatenate $(M-1) \times$ 5-dim motion vectors of the three points in the descending order of their SURF scale values. In addition, we add the difference value of the sizes of Delaunay triangles between consecutive frames regarding $(M-1)$ intervals. This feature vector on size change of triangles becomes $(M-1)$ dimensions.

Fig. 4. The case that motion features are more effective (left), and the case that appearance features are more effective (right)

After weighting the motion vector with w, we concatenate both local appearance and motion vectors into in one $(192 + (M-1) \times 15 + (M-1))$-dim vector. In the experiments, we set 5 to both M and N, and totally the dimension of the final feature vector becomes 256.

4 Feature Fusion

In this paper, we employ Multiple Kernel Learning (MKL) for feature-fusion-based action recognition.

To perform action recognition, we use a Support Vector Machine (SVM) with a linear combination kernel to combine different features. To estimate weights of the kernel, we use Multiple Kernel Learning (MKL) which can estimate optimal weights of the linear combination kernel. First, we extract appearance, motion and spatio-temporal features from each shot. Next, we train a SVM and estimate kernel weights with MKL using training shots. Finally, we classify test shots with a trained SVM and estimated weights.

4.1 Feature Fusion with Multiple Kernel Learning

The most important feature for action recognition is different depending on kinds of action and videos. For example, to distinguish "running" from "jogging", motion features are more important than appearance features as shown in the left-side of Figure 4. On the other hand, to distinguish "boxing" from "hand-clapping", appearance features are more important than motion features. In this paper, we utilize Multiple Kernel Learning (MKL) and estimate weights to fuse different kinds of features using a weighted linear combination.

Since the proposed ST feature is based on local appearances and motions around moving interest points, we use two holistic features on appearance and motion as additional features to be integrated by MKL. We use Gabor histograms as a holistic appearance feature, and optical flow histograms as a holistic motion feature.

To fuse features, we utilize MKL to recognize action. MKL handles a combined kernel which is a weighted liner combination of several single kernels, while a standard SVM can handle only a single kernel. MKL can estimate optimal weights for a linear combination of kernels as well as SVM parameters simultaneously in the train step. The training method of a SVM employing MKL is sometimes called as MKL-SVM. With MKL, we can train a SVM with an

adaptively-weighted combined kernel which fuses different kinds of image features. The combined kernel is as follows:

$$K_{comb}(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{K} \beta_j k_j(\mathbf{x}, \mathbf{x}') \quad \text{with } \beta_j \geq 0, \ \sum_{j=1}^{K} \beta_j = 1. \tag{2}$$

where β_j is weights to combine sub-kernels $K_j(\mathbf{x}, \mathbf{y})$. As a kernel function, we used a chi-square RBF kernel.

By preparing one sub-kernel for each image features and estimating weights by the MKL method, we can obtain an optimal combined kernel. We can train a SVM with the estimated optimal combined kernel from different kinds of image features efficiently.

Sonnenburg et al. [17] proposed an efficient algorithm of MKL to estimate optimal weights and SVM parameters simultaneously by iterating training steps of a standard SVM. This implementation is available as the SHOGUN machine learning toolbox at the Web site of the first author of [17]. In the experiment, we use the MKL library included in the SHOGUN toolbox as the implementation of MKL.

Feature extraction We explain holistic appearance features and motion features which are used for feature fusion by MKL as other features than the proposed local ST feature. The characteristic of these three features are different from each other. All these features are not used as they are, but are vector-quantized and converted to bag-of-features (BoF) vectors regarding a shot.

Appearance feature : We use Gabor texture histograms as an appearance feature. A Gabor texture feature represents texture patterns of local regions with several scales and orientations. In this paper, we use 24 Gabor filters with four kinds of scales and six kinds of orientations. Before applying the Gabor filters, we divide a frame image extracted from video shots into 20×20 blocks. We apply the 24 Gabor filters to each block, then average filter responses within the block, and obtain a 24-dim Gabor feature vector for each block. Totally, we extract 400 24-dim Gabor vectors from each frame image.

Motion feature : Although the proposed ST feature contains motion information, this motion information represents only local motion. As a holistic motion feature, we built motion histograms over a frame image. This feature is expected to have different discriminative power from the proposed ST feature. We extract motion features at grid points with every 8 pixels using the Lucas-Kanade methods [8]. Extracted motion features from each grid are voted to histogram of 7 direction and 8 motion magnitude.

Vector Quantization of Features: Bag-of-Frames In most of the existing work on video shot classification, features are extracted only from key frames. However, extracted features depend on selected frames, and it is difficult to select the most informative key frame. Then, we extract features from all frames

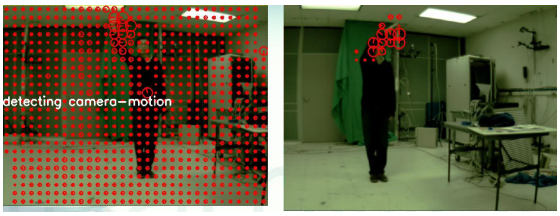

Fig. 5. Example of camera motion (left), example of no camera motion (right)

within each video shot, we vector-quantize all of them and convert them into the bag-of-features (BoF) representation within each shot. While the standard BoF represents the distribution of local features within one image, the BoF employed in this paper represents the distribution of features within one shot which consists of several frame images. We call this BoF regarding one video shot as bag-of-frames (BoFr). ST features are obtained from every N frame images, while motion and appearance features are obtained from one frame image. In both cases, we aggregate all the features within all the frame images extracted from one video shot, and convert them into the BoFr histograms.

In the experiment, we set the size of the codebook of the 256-dim ST features, the 96-dim appearance features and the 56-dim motion features as 5000, 3000 and 3000, respectively.

Camera motion detection Most researches on spatio-temporal feature do not consider camera motions, since most of them assume a fixed camera. However, it is important to treat with camera motion in case of classifying Web video shots. Although compensation of camera motion is possible, accurate compensation is difficult for Web videos. Web videos contain various kinds of intentional and unintentional camera motions, and their resolution is usually low. In this paper, we adopt a simple strategy that we do not extract ST features and motion features from the frames where camera motion is detected, which is the same as [2]. In the actual implementation, we detect camera motion before extracting features as pre-processing.

To detect camera motion, we calculate motion features based on the Lucas-Kanade method at every 8-pixel grid as shown in Figure 5. If the region where motion is detected is larger than a predefined threshold, we consider camera motion is detected.

Although ST and motion features are not extracted from the frames where camera motion is detected, we can construct BoFr vectors if other frames in the shot have no camera motion. If all the frames of a shot are judged with camera motion, we set ST and motion vectors as zero vectors. However, we can still extract appearance features.

Table 2. Comparison of the processing time of ST feature extraction

ours	Kläser	CHLAC
1.38 sec.	18.62 sec.	125 sec.

5 Experiments

In the experiments, we evaluate the proposed method in terms of action recognition with KTH and Youtube datasets, and classify Web video shots.

5.1 Evaluation on Extraction Speed

Before showing results on action recognition, we explain about processing speed briefly. Table 2 shows comparison on the extraction time when extracting ST features from a 312-frame video shot the size of which is 80×60. We compared the proposed ST features with CHLAC (Cubic Higher order Local Auto-Correlation) [6] and a gradient-based method by Kläser et al. [5] on AMD Phenom II X4 3.0GHz with 8GB memory. CHLAC and the gradient-based method took 125 seconds and 18.62 seconds, respectively, while the proposed ST feature took only 1.38 seconds. This shows that our ST feature is very light-weight.

5.2 Datasets

Datasets for action recognition We utilize three kinds of datasets: KTH dataset, Youtube dataset which is build by Liu et al. [2], and another Youtube dataset which we build by ourselves. In this paper, we call the dataset built by Liu et al. as "Wild Youtube dataset" and one built by us as "Our Youtube dataset".

KTH dataset contains only videos taken in a controlled environment such as "no camera motion", "only one person action in shot". On the other hand, Web videos such Youtube videos are take in unrestricted environments. Therefore, it is much more difficult to recognize action in the Youtube datasets than KTH.

Next, we recognize action for the Wild Youtube dataset (Figure 6) to compare our proposed method and the method proposed by Liu et al. [2]. This dataset 11 actions ("basketball shooting", "volleyball spiking", "trampoline jumping", "soccer juggling", "horse riding", "cycling", "diving", "swinging", "golf swinging", "tennis swinging", "walking_with_dog"). Because this dataset includes "camera motion" or "changing view location", "back ground clutter", it is a challenging dataset.

Datasets for Large-scale Web video shot ranking We collected videos from Youtube, and built our original Youtube dataset (Figure 7). This dataset contains six motion ("batting", "running", "walking", "shoot", "jumping", "eating"), which collect from Youtube by ourselves. Table 3 shows the statistics of this data. We utilize total 37,179 shots which extracted from 974 videos in this experiment, which is much more large-scale than the Wild Youtube dataset.

Fig. 6. Wild Youtube dataset

Fig. 7. Our Youtube dataset

Table 3. Data for large scale Web video shot ranking

Action	Number of videos	Number of shots	Average time [second]	Total time [hour]	Training data	
					positive	negative
batting	174	8,980	5.9	14.6	31	75
running	170	7,342	6.6	13.4	28	66
walking	174	6,567	7.4	13.4	23	63
shoot	164	7,718	5.3	11.3	14	75
eating	142	3,442	7.7	7.3	22	64
jumping	160	3,130	6.6	5.8	27	40
Total	984	37,179	6.6	65.8	145	383

We select shots at random for training data and classify them into positive or negative samples by hand. As pre-processing, we divided collected videos into shots with a simple color-histogram-based shot boundary detection. We use this dataset for experiments on large-scale Web video shot ranking which is one of possible practical applications of ST features and action recognition. This will help people search a large-scale video collection for the shots including the given actions.

5.3 Experimental results

Action recognition

KTH dataset : KTH dataset is one of the most widely used dataset. We train leave-one-out and in the test phase we apply 1-vs-all multi-class classification, following the experiment setup of [3]. Table 7(left) shows the confusion matrix in case of using the MKL-based feature fusion method with all kinds of the features. From this table, it is difficult to distinguish "running" and "jogging". Table 4 shows the comparison of our method and state-of-the-art regarding the classification rate. In this table, "visual", "motion", "ST [16]", "VMR", "MKL(M+V)", "MKL(V+M+VMR)" means the result by only a visual appearance feature, by a holistic motion feature, by our previous ST features proposed in [16], by the proposed ST features, by combining appearance and visual features with MKL, and by combining all of the three features with MKL, respectively.

The result, 91.7%, by "VMR" which is the proposed ST feature, outperformed the result, 86.3%, by the original ST features which is equivalent to the ST feature without Delaunay triangulation. The gain by using Delaunay triangulation was 5.4%. By using the motion feature, we obtained 92.7% which

Table 4. Comparison classification rate for KTH dataset

	visual	motion	ST[16]	VMR	MKL V+M	MKL V+M+VMR
walking	0.47	0.93	0.94	1.00	0.94	0.99
jogging	0.09	0.93	0.76	0.86	0.87	0.92
running	0.41	0.87	0.81	0.83	0.92	0.87
boxing	0.76	0.96	0.91	0.90	0.96	0.96
waving	0.98	0.92	0.90	0.98	0.98	0.98
clapping	0.65	0.95	0.86	0.93	0.94	0.96
average	0.487	0.927	0.863	0.917	0.935	0.945

Table 5. Compare to state-of-art for KTH dataset

Dállor et al. [3]	81.2%
Liu et al. [2]	91.8%
Gilbert et al. [9]	96.7%
Kim et al. [11]	95.33%
Uemura et al. [10]	93.7%
ours	94.5%

Table 6. Comparison classification rate for Wild Youtube dataset

	visual	motion	VMR	MKL V+M	MKL V+M+VMR	Liu[2]
b_shooting	0.46	0.32	0.44	0.53	0.61	0.53
cycling	0.82	0.73	0.59	0.85	0.88	0.73
diving	0.88	0.72	0.82	0.92	0.93	0.81
g_swinging	0.78	0.57	0.72	0.82	0.87	0.86
h_riding	0.85	0.82	0.74	0.89	0.87	0.72
s_juggling	0.14	0.56	0.54	0.52	0.61	0.54
swinging	0.68	0.55	0.63	0.60	0.68	0.57
t_swinging	0.79	0.44	0.53	0.89	0.88	0.80
t_jumping	0.66	0.62	0.67	0.55	0.73	0.79
v_spiking	0.89	0.47	0.77	0.88	0.91	0.73
walking	0.66	0.43	0.53	0.83	0.86	0.75
average	0.691	0.565	0.634	0.735	0.804	0.712

is the best result among single features. By combining appearance and motion features with MKL, we obtain 93.5%, while by combining all features the result was improved to 94.5%. This means the proposed ST feature boosted the classification performance. We obtained the observation that motion feature is more important than appearance features for KTH dataset.

Figure 8(right) shows the feature weights estimated by MKL. The weight of appearance features was low because variation in holistic appearances of frame images is very small in KTH dataset. On the other hand, weights of motion feature is relatively high, especially for "running" and "jogging". It means that motion feature is important for classifying these actions. Compared to the state-of-the-art methods as shown in Table 5, our method is mostly equivalent, which means our method is effective for action recognition for controlled video datasets such as KTH.

Wild Youtube dataset : Videos on the Web are not as simple as KTH dataset, since they are uncontrolled and probably include camera motion. We made an experiment using wild Youtube dataset to compare Liu's results [2]. For evaluation, we utilized 5 fold cross validation, which is the same as [2]. Table 7(right) shows confusion matrix by MKL with appearance, motion and ST features. Table 6 shows comparison between our method and Liu et al. [2]. "Basket_shooting" was not classified well in all the case, because Basket_shooting shots contain lots of camera motion.

The classification rate by ST features is 63.4% and one by appearance features is 69.2%. By combining appearance and motion features with MKL, we obtained 73.5%, and by combining all the features we obtained 80.4%. This shows

Table 7. Confusion matrix classifying by MKL (left)KTH dataset and (right)Wild Youtube dataset

	walking	jogging	running	boxing	waving	clapping
walking	0.99	0.01	0	0	0	0
jogging	0.04	0.92	0.04	0	0	0
running	0	0.13	0.87	0	0	0
boxing	0.01	0	0	0.96	0	0.03
waving	0	0	0	0	0.98	0.02
clapping	0	0	0	0.04	0	0.96

KTH dataset (94.5%)

	b_shooting	cycling	diving	g_swinging	h_riding	s_juggling	swinging	t_swinging	t_jumping	v_spiking	walking
b_shooting	61.4	1.4	9.3	14.3	2.1	5.7	0.7	1.4	0	3.6	0
cycling	0	88.2	1.4	0	9.7	0	0.7	0	0	0	0
diving	2.6	0	92.9	0.6	2.6	0	0.6	0	0	0.6	0
g_swinging	2.8	0	0.7	87.3	1.4	5.6	2.1	0	0	0	0
h_riding	0	6.6	2	1.5	87.4	0.5	1	0.5	0	0.5	0
s_juggling	2.6	0	3.9	14.8	2.6	61.3	9	3.2	0	2.6	0
swinging	0	9.6	0.7	5.1	0	2.2	68.4	0	9.6	0	4.4
t_swinging	0	0	4.2	0	0	0	0	88	1.8	3.6	2.4
t_jumping	0	0.8	0	1.7	0	3.4	1.8	0	73.1	0.8	8.4
v_spiking	4.3	0	0	0.9	0.9	0	0	0.9	1.7	90.5	0.9
walking	0	0.8	1.6	0.8	3.3	0	0.8	2.4	0.8	3.3	86.2

Wild Youtube dataset (80.4%)

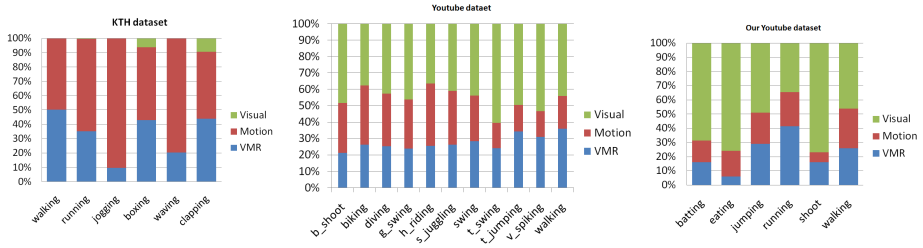

Fig. 8. Estimated weights by MKL (left)KTH dataset, (middle)Wild Youtube dataset and (right)Our Youtube dataset

the proposed ST feature improved the classification rate by 9.2%. Both results outperformed Liu's result greatly which was 71.2%, which indicated that the proposed MKL-based fusion method is effective for unrestricted video datasets such as the Wild Youtube dataset. Figure 8 shows the weights of combining features. The weights of appearance features tend to become large compared to the KTH. Because Youtube data contain lots of camera motion, reliability of motion feature is low. This shows compensation of camera motion is needed for Web video shots.

Large scale web video shot ranking We made experiments on Web video shots ranking with the same supervised method as the method for KTH and Wild Youtube. All test shots have been ranked based on the output value of SVM. Figure 9 shows the result of Web shot ranking. Since this dataset contains about 37,000 shots, we cannot compute the recall rate. Instead, we show the precision at the N-th ranking. The X-axis of Figure 9 shows N-th ranking, and the Y-axis represents the precision within the N-th ranking. For example, the precision of 20-th ranking on results on "batting" by VMR is 1.0, and 40-th ranking is 0.975. We evaluate 5 different method, VMR(ST feature), appearance, motion, random(baseline) and MKL with all kinds of the features. The rightside of Figure 8 shows the weights estimated by the MKL. The weight of appearance

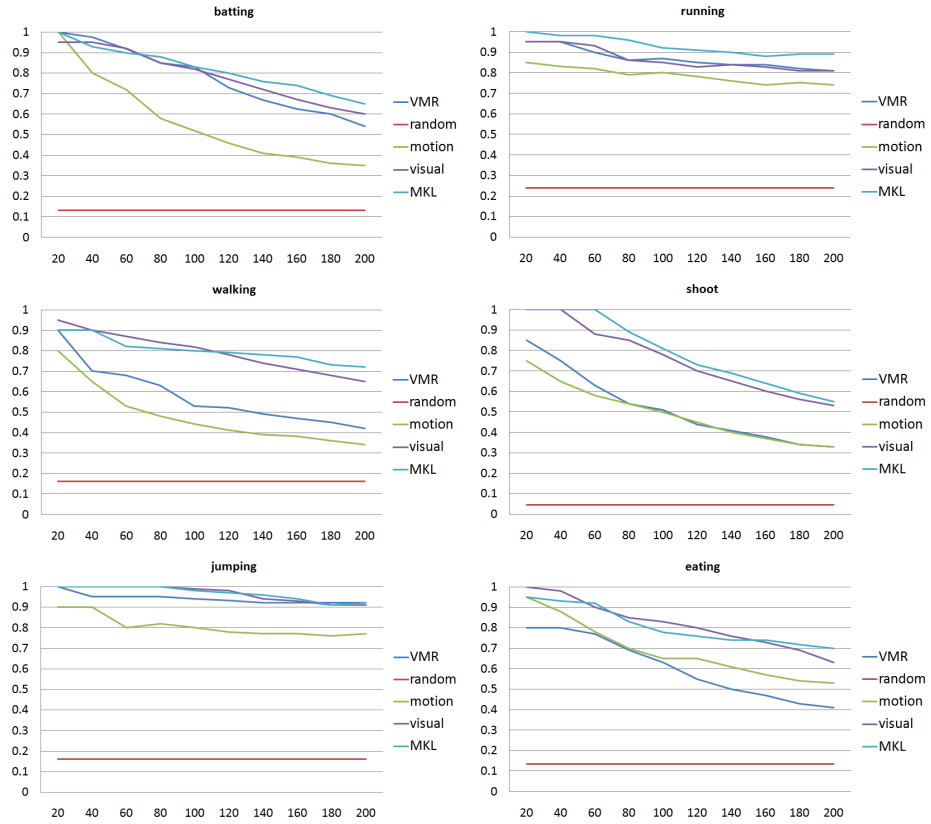

Fig. 9. Results of web video ranking

features tends to be relatively high, especially for the actions the places of which occur are limited such as "batting", "eating" and "shoot".

Regarding the average result over six kinds of actions, appearance feature produced the best result among single features. This tendency is similar to Wild Youtube dataset. The average of the precision of appearance features over six actions within 20-th and within 200-th are 0.98 and 0.69, respectively. This means appearance features is more effective than motion to classify web video shots. In terms of the average results over six actions, MKL was the best. However, as shown in Figure 9, for "walking", "jumping" and "eating", the result by only appearance features are almost equivalent to one by MKL-based fusion of all the features. This shows that some actions have typical scenes where the actions happen frequently, which is sometimes called "context".

6 Conclusions

We proposed a SURF-based light-weight spatio-temporal feature which is suitable for large-scale video shot datasets, and we proposed an action recognition method by combining features based on Multiple Kernel Learning (MKL), which

enables robust action recognition for both controlled and uncontrolled videos. In the experiments, we evaluated our method using KTH dataset, and Youtube dataset. As a result, we obtained 94.5% as a classification rate for in KTH dataset which was almost equivalent to state-of-the-art, and 80.4% for Youtube dataset which outperformed state-of-the-art greatly. In addition, we made experiments on large-scale Web video shots ranking with the proposed methods.

As future work, we plan to introduce camera motion compensation and treat with video shots including multiple actions.

References

1. Schuldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local SVM approach. In: Proc. of International Conference on Pattern Recognition. (2004) 32–36
2. Liu, J., Luo, J., Shah, M.: Recognizing realistic action from videos. In: Proc. of IEEE Computer Vision and Pattern Recognition. (2009)
3. Dollar, P., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. of Surveillance and Performance Evaluation of Tracking and Surveillance. (2005) 65–72
4. Laptev, I., Lindeberg, T.: Local descriptors for spatio-temporal recognition. In: Proc. of IEEE International Conference on Computer Vision. (2003)
5. Kläser, A., Marszałek, M., Schmid, C.: A spatio-temporal descriptor based on 3d-gradients. In: Proc. of BMVA British Machine Vision Conference. (2008) 995–1004
6. Kobayashi, T., Otsu, N.: A three-way auto-correlation based approach to human identification by gait. In: Proc. of IEEE Workshop on Visual Surveillance. (2006) 185–192
7. Herbert, B., Andreas, E., Tinne, T., Luc, G.: Surf: Speeded up robust features. Computer Vision and Image Understanding (2008) 346–359
8. Lucas, B., Kanade, T.: An iterative image registration technique with an application to stereo vision. In: Proc. of International Joint Conference on Artificial Intelligence. (1981) 674–679
9. Gilbert, A., Illingworth, J., Bowden, R.: Fast realistic multi-action recognition using mined dense spatio-temporal features. In: Proc. of IEEE International Conference on Computer Vision. (2009) 925–931
10. Uemura, H., Ishikawa, S., Mikolajczyk, K.: Feature tracking and motion compensation for action recognition. In: Proc. of BMVA British Machine Vision Conference. (2008)
11. Kim, T., Wong, S., Cipolla, R.: Tensor canonical correlation analysis for action classification. In: Proc. of IEEE Computer Vision and Pattern Recognition. (2009)
12. Varma, M., Ray, D.: Learning the discriminative power-invariance trade-off. In: Proc. of IEEE International Conference on Computer Vision. (2007) 1150–1157
13. Sun, J., Wu, X., Yan, S., Cheong, L.F., Chua, T.S., Li, J.: Hierarchical spatio-temporal context modeling for action recognition. In: Proc. of IEEE Computer Vision and Pattern Recognition. (2009)
14. Han, D., Bo, L., Sminchisescu, C.: Selection and context for action recognition. In: Proc. of IEEE International Conference on Computer Vision. (2009)
15. Cinbins, N.I.., Cinbins, R.G., Sclaroff, S.: Learning action from the web. In: Proc. of IEEE International Conference on Computer Vision. (2009) 995–1002
16. Noguchi, A., Yanai, K.: Extracting spatio-temporal local features considering consecutuveness of motions. In: Proc. of Asian Conference on Computer Vision(ACCV). (2009)
17. Sonnenburg, S., Rätsch, G., Schäfer, C., Schölkopf, B.: Large scale multiple kernel learning. Journal of Machine Learning Research **7** (2006) 1531–1565