Robust Action Recognition Using Local Motion and Group Sparsity

Jungchan Cho[a], Minsik Lee[a], Hyung Jin Chang[b], Songhwai Oh[a,*]

[a] *Department of Electrical and Computer Engineering and ASRI,*
Seoul National University, Korea
[b] *Department of Electrical and Electronic Engineering,*
Imperial College London, UK

Abstract

Recognizing actions in a video is a critical step for making many vision-based applications possible and has attracted much attention recently. However, action recognition in a video is a challenging task due to wide variations within an action, camera motion, cluttered background, and occlusions, to name a few. While dense sampling based approaches are currently achieving the state-of-the-art performance in action recognition, they do not perform well for many realistic video sequences since, by considering every motion found in a video equally, the discriminative power of these approaches is often reduced due to clutter motions, such as background changes and camera motions. In this paper, we robustly identify local motions of interest in an unsupervised manner by taking advantage of group sparsity. In order to robustly classify action types, we emphasize local motion by combining local motion descriptors and full motion descriptors and apply group sparsity to the emphasized motion features using the multiple kernel method. In experiments, we show that different types of actions can be well recognized using a small number of selected local motion descriptors and the proposed algorithm achieves the state-of-the-art performance on popular benchmark datasets, outperforming existing methods. We also demonstrate that the group sparse representation with the multiple kernel method can dramatically improve the action recognition performance.

Keywords: Action recognition, Motion descriptor, Sparse representation, Dynamic scene understanding

1. Introduction

Action recognition is an important problem in computer vision, which can be applied to many interesting applications, such as automatic video indexing and retrieval, human-computer interaction, and intelligent surveillance. Many works have recently focused on enhancing motion information by using trajectories obtained from point trackers [1, 2, 3] since an action in a video occupies in a 3D space (2D spatial domain and 1D time domain) unlike an object in a 2D image. However, there are a number of issues which make it difficult to recognize actions from real world videos using trajectory-based methods.

Many real-world video sequences contain a large amount of camera motion, which makes the recognition performance degrade. While one approach for solving the problem is to correct the camera motion using video stabilization as a preprocessing step before action recognition [4], perfect video stabilization is not possible in many practical cases and there is a danger of losing critical information. To handle camera motion, Wang et al. [3] introduced a motion boundary based descriptor, initially developed in the context of human detection [5]. Motion boundaries are computed by a derivative operation on the optical flow field. Thus, motion due to the translational local camera movement is canceled out and relative motion is captured [3]. It is demonstrated that motion boundary based descriptors outperform other existing motion descriptors in many realistic videos.

Another issue is the large variability in actions. When different subjects are performing the same action, they do not have the same appearance and their movements can be quite different for the same action. Even for a person performing the same action multiple times, each performance can be quite different from the previous one. Therefore, robust classification is an important issue in the human action recognition problem and it is necessary to develop a more robust alternative.

Recently, the concept of sparse representation has received significant attention and demonstrated promising performance in signal processing and computer vision [6, 7, 8]. It has been discovered in neuroscience [9] that the human vision system seeks a sparse representation of an incoming image using an overcomplete dictionary. In addition, recent studies go beyond sparsity and take into account additional information about the underlying structure of solutions [7]. Namely, the solution has a natural grouping of its components and the use of this group sparsity can reduce degrees of freedom in a solution, thereby leading to a better solution [10]. In [7], a group sparsity method has been successfully applied to object recognition by kerneliz-

[*] Corresponding author. Tel.: +82 2 880 1511.
Email addresses: cjc83@snu.ac.kr (Jungchan Cho), mlee.paper@gmail.com (Minsik Lee), hj.chang@imperial.ac.uk (Hyung Jin Chang), songhwai@snu.ac.kr (Songhwai Oh)

ing the accelerated proximal gradient (APG) method [11].

In this paper, we propose a method for robust action recognition using local motion and group sparsity. An overview of the proposed method is shown in Figure 1. The first part of the proposed method is the local motion selection. Since descriptors extracted from an uncontrolled realistic video may include clutter motions due to camera motion, background changes, and occlusions, if we can isolate local motion from global motion, the performance of action recognition can be improved. However, motion clustering is not a trivial task since the number of motion clusters of trajectories is not available. We propose a new motion clustering method using the group sparsity formulation and select important local motions using spatial information of motion clusters. From an extensive set of experiments, we show that the selected local motion descriptors provide better performance than that of full motion descriptors, despite the fact that the selected local motion descriptors are only about 28.6% of full motion descriptors. It demonstrates that the proposed method can robustly select important information for action recognition.

While the proposed local motion selection method efficiently selects local motion as shown in Section 5, if we can reduce the risk of incorrect action recognition coming from an incomplete motion separation, the performance can be further improved. To reflect this point, we emphasize local motion in the second part, by appending the information obtained from local motion to the full motion information obtained from a video clip using the multiple kernel method [12]. Since full motion descriptors already include local motion descriptors, the distance between actions with similar local motion patterns is shortened and the distance between actions with different local motion patterns is lengthened in our proposed scheme. Therefore, we can say that the local motion descriptors play a role of correcting the distance between samples contaminated by global motion patterns, making action recognition more robust. We call this approach as a local motion emphasis.

Finally, in the third part, we classify action classes using the group sparse representation with the multiple kernel method, instead of a support vector machine (SVM), a popular classifier which is widely used in many action recognition algorithms. Our experimental results show that the proposed action recognition method with local motion emphasis and group sparsity significantly outperforms the baseline method using full motion descriptors and an SVM classifier [3]. For example, the proposed method outperforms the baseline method [3] by 9.5% for the Olympic Sports dataset [13].

The remainder of this paper is organized as follows. Section 2 briefly explains related work and the baseline method [3]. The proposed motion clustering and local motion selection methods are described in Section 3. In Section 4, we explain the motion emphasis and classification using the group sparse representation with the multiple kernel method. Experimental results are described in Section 5.

2. Preliminaries

2.1. Related Work

The proposed method is based on trajectories of points, subspace clustering, and sparse representation, and we briefly introduce them in this section.

Trajectory-based method: To overcome limitations of 2D based descriptors, many works have recently tried to enforce motion information using trajectories obtained from point trackers [1, 2, 3]. Messing et al. [1] proposed velocity history features based on a sophisticated latent velocity model and side information, such as appearance, position, and high level semantic information. They have demonstrated the superiority of velocity history features on high resolution video sequences of complicated activities. Sun et al. [2] proposed an approach which hierarchically models the spatio-temporal context information about trajectories obtained by matching SIFT descriptors between consecutive frames and showed impressive results on realistic action and event recognition. Wang et al. [3] proposed a dense trajectory-based approach by combining point tracking and dense interest point sampling and achieved the state-of-the-art results for action recognition compared to sparse interest point sampling techniques, such as the Kanade-Lucas-Tomasi (KLT) tracker [14]. Since it is the baseline method in our paper, we give details of the method proposed in [3] in Section 2.2.

Subspace-based clustering: Recently, a subspace based clustering method using sparse representation has been proposed [15] and shown to provide better performance than existing methods, such as k-means, especially when data points are corrupted by noises and outliers. This is based on the idea that each data point in a subspace can be represented as a linear combination of other points in the same subspace and the coefficients obtained from sparse representations have the connectivity information between data points. Therefore, the coefficients can be used to define an affinity matrix of data points and, by performing the normalized cut [16] on this graph, subspace clustering can be achieved. However, the normalized cut is computationally expensive for clustering motions in a video, hence, we proposed a new approach based on group sparsity.

Sparse representation: A work on image-based face recognition [6] has shown that the sparse representation is naturally discriminative as it selects only a small number of basis vectors that can most compactly represent the given signal. In [6], a single overcomplete dictionary is formed by concatenating vectorized training samples of all classes. Given a test image, its sparsest representation over the dictionary is found by l_1 minimization. The underlying assumption of this method is that a good number of training samples are available per class and they span the sample space well. Guha et al. [8] also explored the effectiveness of sparse representation obtained by learning a set of overcomplete dictionaries in the context of action recognition in videos. They proposed three different dictionary training frameworks: (1) one dictionary for all classes (*shared*), (2) one dictionary per class (*class-specific*), and (3) a concatenation of class-specific dictionaries (*concatenated*). When analyzing their experimental results, we find that the *shared* method shows

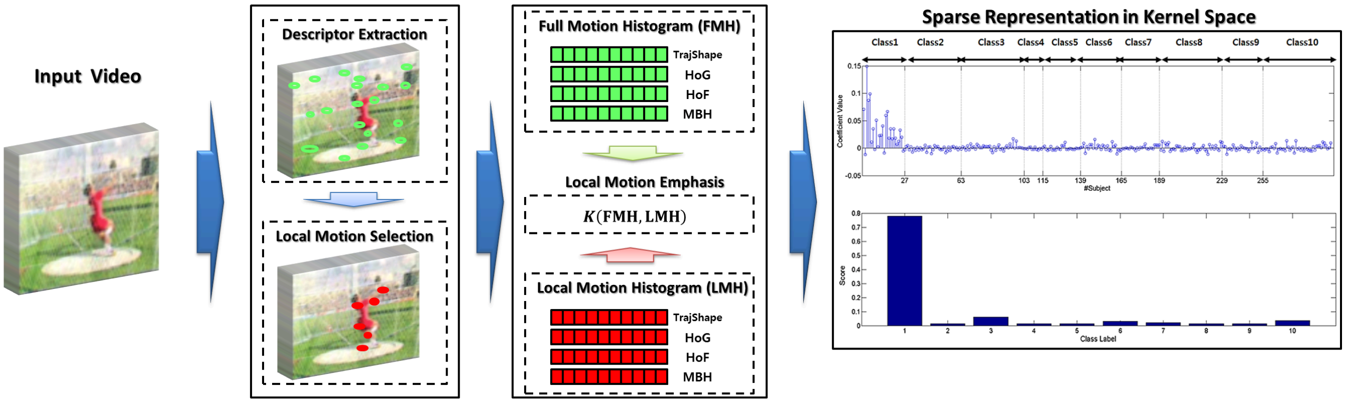


Figure 1: An overview of the proposed method. Our algorithm consists of three parts. In the first part, local motions are selected by the proposed local motion selection method (Section 3.2). In the second part, local motions are emphasized by adding local motion descriptors to full motion descriptors using the multiple kernel method. Finally, the classification is performed using the group sparse representation with the multiple kernel method. The right figure shows the group sparse representation of a test sample and decision scores based on training samples in each class.

lower performance than other two methods. It illustrates the fact that the solution has a certain group sparse structure. It is a motivation for the proposed classification method based on group sparsity.

2.2. *Baseline Method - Dense Trajectories*

We adopt the dense trajectory approach by Wang et al. [3] to generate motion descriptors and it is briefly introduced in this section. Note that our approach can be applied to any local patch trajectories. Feature points are sampled in eight spatial scales with a grid spaced by W pixels and each point $P_t = (x_t, y_t)$ at frame t is tracked to the next frame $t+1$ by median filtering of a dense optical flow field $\omega_t = (u_t, v_t)$.

$$P_{t+1} = (x_{t+1}, y_{t+1}) = (x_t, y_t) + (M * \omega_t)|_{(\bar{x}_t, \bar{y}_t)}, \qquad (1)$$

where M is the median filtering kernel whose size is $N_\omega \times N_\omega$ pixels and $(\bar{x}_t, \bar{y}_t)$ is the rounded position of P_t. Points of subsequent frames are concatenated to form a trajectory $\mathcal{T} = (P_t, P_{t+1}, P_{t+2}, \ldots)$. To extract a dense optical flow, the algorithm by Färneback [17] is adopted.

In the point tracking process, the effects of noise, light conditions, and other factors appear in the form of a drift which is an accumulation of small errors. To avoid this drifting problem, the maximum length of a trajectory is limited to L. Also, trajectories with sudden large or small displacements are removed, since trajectories with small displacements do not contain significant motion information and trajectories with sudden large displacements are most likely to be erroneous. A trajectory is considered to have a small displacement, if the diameter of the smallest region containing the trajectory is less than N_{min} pixels. A trajectory has a large displacement, if the diameter of the smallest region containing the trajectory is larger than N_{max} pixels or the displacement vector between two consecutive frames is larger than a threshold α.

After tracking feature points, the shape of a trajectory, called *TrajShape*, is described by concatenating a set of displacement vectors $\Delta P_t = (P_{t+1} - P_t) = (x_{t+1} - x_t, y_{t+1} - y_t)$. In order to make a trajectory shape descriptor invariant to scale changes, a concatenated vector is normalized by the overall magnitude of motion displacements:

$$\mathbf{s} = \frac{[\Delta P_t, \cdots, \Delta P_{t+L-1}]}{\sum_{i=t}^{t+L-1} \|\Delta P_i\|}. \qquad (2)$$

Also, the local motion and appearance in a 3D video volume around a trajectory are described by a histogram of oriented gradients (HOG) [5], a histogram of optical flow (HOF), and a motion boundary histogram (MBH). HOG encodes the local appearance information, while HOF and MBH capture local motion patterns. A 3D video volume, which has the size of $N \times N$ pixels and L frames, is subdivided into $n_\sigma \times n_\sigma \times n_\tau$ cells and each feature is computed at each cell. For HOG, gradient orientations are quantized into eight bins. HOF has nine bins in total, with one extra bin for zero angle. Both descriptors are normalized with their l_2 norm. MBH computes a histogram based on the derivatives of optical flows on both horizontal and vertical components. Like HOG, eight bins are used to quantize orientations and values are normalized using the l_2 norm. Although the descriptors have been shown to be effective for action recognition in unconstrained videos, they are still contaminated by many global motion patterns generated by camera motion and background motions. The proposed local motion emphasis method alleviates the effect of the contaminated descriptors using local motion descriptors, which improve the action recognition performance as shown in Section 5.

3. Motion Clustering and Local Motion Selection

Since the number of motion clusters of trajectories is not available, motion clustering is a challenging task. We propose a new motion clustering method using group sparsity. The main idea is that given a set of trajectories which begin at the same frame, the proposed motion clustering method selects a few key trajectories, which can represent all other trajectories and assigns the membership based on the similarity to the key trajectories.

3.1. Motion Clustering

Let $\mathbf{S} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_p] \in \mathbb{R}^{m\times p}$ be a feature matrix consisting of p TrajShape descriptors, where m is the dimensionality of a TrajShape descriptor. In general, selecting similar samples with the sparse representation can be formulated as

$$\min_{\mathbf{R}} \frac{1}{2}\|\mathbf{S} - \mathbf{S}\mathbf{R}\|_F^2 + \kappa\|\mathbf{R}\|_1, \tag{3}$$

where $\mathbf{R} = [r_{ij}] \in \mathbb{R}^{p\times p}, i = 1, \ldots, p, j = 1, \ldots, p$ and κ is a regularization parameter. The Frobenius norm $\|\mathbf{R}\|_F$ is defined as $\|\mathbf{R}\|_F = \sqrt{\sum_{i,j} r_{ij}^2}$ and the l_1 norm is defined as $\|\mathbf{R}\|_1 = \sum_{i,j} |r_{ij}|$. However, the problem (3) does not exclude a trivial solution, i.e., an identity matrix $\mathbf{I}$. ($\mathbf{I}$ makes the first term of (3) to zero and is also very sparse.) Hence, to avoid this trivial solution, we modify the l_1 norm constraint in (3) into the $l_{2,1}$ norm, defined as $\|\mathbf{R}\|_{2,1} = \sum_{i=1}^{p} \|\mathbf{r}_i\|_2$, where $\mathbf{r}_i$ denotes the i-th row of $\mathbf{R}$. The problem is now reformulated as:

$$\min_{\mathbf{R}} \frac{1}{2}\|\mathbf{S} - \mathbf{S}\mathbf{R}\|_F^2 + \kappa\|\mathbf{R}\|_{2,1}. \tag{4}$$

With this $l_{2,1}$ norm constraint, the solution contains many zero-valued rows in $\mathbf{R}$, i.e., row sparsity. Notice that $\mathbf{S}$ plays a role as a set of bases and $\mathbf{R}$ corresponds to a set of coefficients. Hence, the absolute values of elements in $\mathbf{R}$ represent the connectivity between trajectories [15].

For example, assuming that there are nine trajectories which begin at frame t as shown in the left of Figure 2, we can obtain a trajectory shape matrix $\mathbf{S}$. By solving (4), we can also obtain a coefficient matrix $\mathbf{R}$ as shown in the bottom of the motion clustering box in Figure 2. The white color in matrix $\mathbf{R}$ indicates that the value is "0" and the darker the color is, the higher the value is. In this example, the coefficient matrix $\mathbf{R}$ has zero values except for three rows. Since an entry in $\mathbf{R}$ indicates the connectivity between a pair of trajectories [15], we can say that there are three key trajectories, i.e., $\mathcal{T}_3$, $\mathcal{T}_7$ and $\mathcal{T}_8$. The remaining trajectories are related to the key trajectories and $|r_{ij}|$ represents the relationship between trajectory i and trajectory j as shown in the motion clustering box of Figure 2, where the width of a line indicates the strength of the connectivity between two trajectories. The coefficient $\mathbf{R}$ obtained from (4) has row sparsity, but all the values in the selected rows can be dense. It means that all trajectories have connections with selected key trajectories. Therefore, we can not directly use $\mathbf{R}$ for trajectory clustering.

The subspace clustering method from [15] uses coefficients to define an affinity matrix of an undirected graph and performs the normalized cut [16] on this graph. However, the normalized cut is computationally expensive. To efficiently solve this problem, we assign the membership of trajectories based on the key trajectory with the highest connection, i.e., the highest $|r_{ij}|$ value, as shown in third stage of motion clustering process in Figure 2. In our example, three clusters based on three key trajectories $\mathcal{T}_3$, $\mathcal{T}_7$, and $\mathcal{T}_8$ can be made as $\mathcal{C}^1 = \{\mathcal{T}_2, \mathcal{T}_3\}, \mathcal{C}^2 = \{\mathcal{T}_1, \mathcal{T}_4, \mathcal{T}_5, \mathcal{T}_6, \mathcal{T}_7\}$, and $\mathcal{C}^3 = \{\mathcal{T}_8\}$. Since our goal is to isolate the local motion from global motion and emphasize them, the proposed greedy motion clustering is sufficient.

From now on, we explain how to obtain a solution for (4). The $l_{2,1}$ norm is indeed a general version of the l_1 norm since if $\mathbf{R}$ is a vector, then $\|\mathbf{R}\|_{2,1} = \|\mathbf{R}\|_1$. In addition, $\|\mathbf{R}\|_{2,1}$ is equivalent to $\|\mathbf{d}\|_1$ by constructing a new vector $\mathbf{d} \in \mathbb{R}^p$ with $d_i = \|\mathbf{r}_i\|_2$. Although there exist general optimization algorithms for solving (4), such as a subgradient based algorithm, the convergence rate can be quite slow since $\|\mathbf{R}\|_{2,1}$ is nonsmooth. Recently, Beck et al. [11] proposed an efficient algorithm for solving a nonsmooth convex optimization problem with a guaranteed convergence rate of $O(1/K^2)$, where K is the number of iterations. Following the framework of [11], let us consider $f(\mathbf{R}) = \|\mathbf{S} - \mathbf{S}\mathbf{R}\|_F^2$ and $g(\mathbf{R}) = \kappa\|\mathbf{R}\|_{2,1}$ and apply a proximal regularization of the linearized function of $f(\mathbf{R})$ at a given point $\mathbf{Z}$:

$$W_\eta(\mathbf{R}, \mathbf{Z}) := f(\mathbf{Z}) + \langle\nabla f(\mathbf{Z}), \mathbf{R} - \mathbf{Z}\rangle + \frac{\eta}{2}\|\mathbf{R} - \mathbf{Z}\|_F^2 + \kappa\|\mathbf{R}\|_{2,1}, \tag{5}$$

which has a unique minimizer $p_\eta(\mathbf{R}) := \arg\min_{\mathbf{R}}\{W_\eta(\mathbf{R}, \mathbf{Z})\}$. With simple algebra (ignoring constant terms in $\mathbf{Z}$), we can obtain

$$p_\eta(\mathbf{R}) = \arg\min_{\mathbf{R}} \left\{ \frac{1}{\eta} g(\mathbf{R}) + \frac{1}{2} \left\| \mathbf{R} - \left(\mathbf{Z} - \frac{1}{\eta}\nabla f(\mathbf{Z}) \right) \right\|_F^2 \right\}, \tag{6}$$

where η is a Lipschitz constant of the gradient $\nabla f(\mathbf{Z})$ and plays a role as a step size in optimization. We set η to $2\lambda_{\max}(\mathbf{S}^T\mathbf{S})$ according to [11], where $\lambda_{max}(\mathbf{A})$ is the maximum eigenvalue of $\mathbf{A}$. Finally, by representing $\left(\mathbf{Z} - \frac{1}{\eta}\nabla f(\mathbf{Z})\right)$ as $\mathbf{Q}$ and $\frac{\kappa}{\eta}$ as τ, the closed form solution of (6) can be obtained by the following theorem [18]:

Theorem 1. *Let* $\mathbf{Q} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_i, \ldots]^T$ *be a given matrix and* $\|\cdot\|_F$ *be the Frobenius norm. If the optimal solution of*

$$\min_{\mathbf{R}} \tau\|\mathbf{R}\|_{2,1} + \frac{1}{2}\|\mathbf{R} - \mathbf{Q}\|_F^2 \tag{7}$$

is $\mathbf{R}^*$*, then the i-th row of* $\mathbf{R}^*$ *is*

$$\mathbf{r}_i = \begin{cases} 0, & \|\mathbf{q}_i\| \le \tau \\ (1 - \tau/\|\mathbf{q}_i\|)\mathbf{q}_i, & \textit{otherwise.} \end{cases} \tag{8}$$

The motion clustering algorithm is summarized in Algorithm 1.

3.2. Local Motion Selection

In the preceding section, we have explained how to cluster similar trajectories in an unsupervised manner. In this section, we focus on the selection of local motion using the fact that trajectories corresponding to one local motion pattern are spatially concentrated, while trajectories with global motion patterns, e.g., camera motion, are spatially spread apart. A motion cluster is selected as local motion if it has more than one member trajectory and spatial standard deviations of member trajectories do not exceed user defined thresholds (β_x, β_y) which are chosen experimentally. In our example shown in Figure 2,

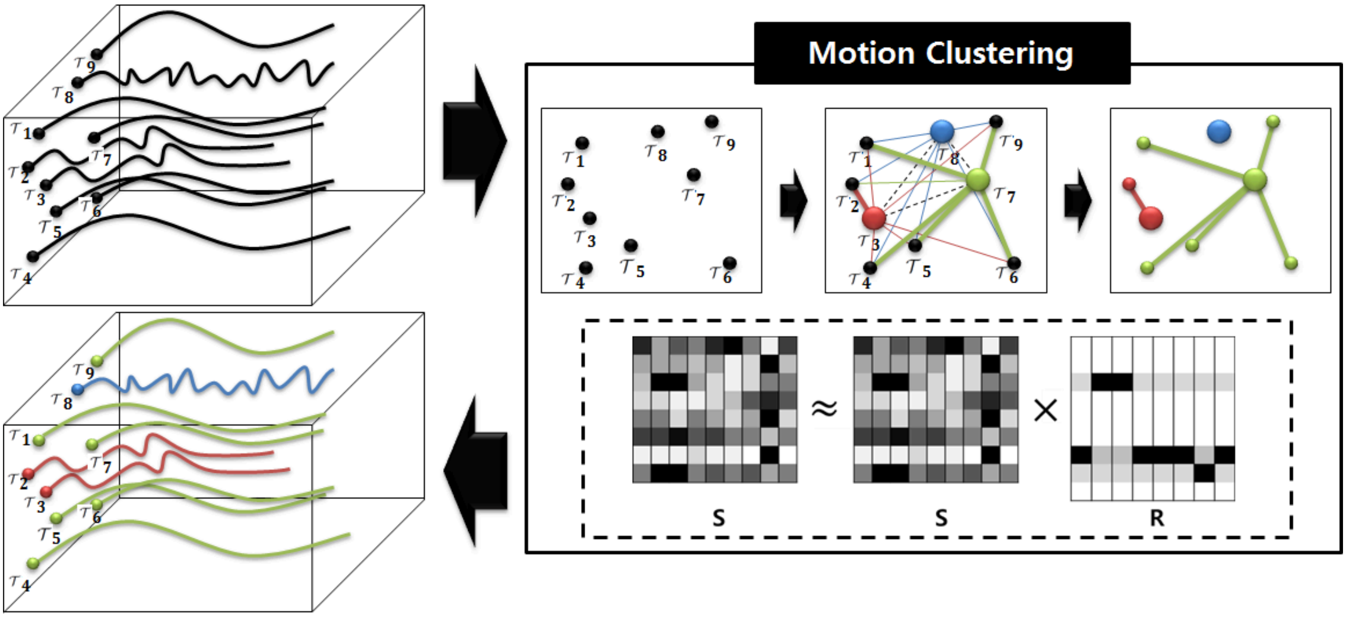


Figure 2: An illustration of the proposed unsupervised motion clustering method. The proposed motion clustering method selects a few key trajectories ($\mathcal{T}_3$, $\mathcal{T}_7$ and $\mathcal{T}_8$ in this example) by solving (4) and assigns the membership of remaining trajectories based on the similarity to the key trajectories. Different colors indicate different motion clusters. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

Algorithm 1 Motion Clustering Algorithm

```
Input: $\mathbf{S}, \mathbf{R}_0, \eta, \kappa > 0, K$
Output: A set of motion clusters $\mathcal{C} = \{\mathcal{C}^c | c = 1, \ldots, N_c\}$
Initialize $\mathbf{Z}_0 = \mathbf{R}_0, t_0 = 1, k = 0$.
for $k = 0, 1, 2, \ldots, K-1$ do
    Calculate $\mathbf{R}_{k+1}$ by (6), (7), and (8).
    $t_{k+1} = \frac{1+\sqrt{1+4t_k^2}}{2}$
    $\mathbf{Z}_{k+1} = \mathbf{R}_k + \left(\frac{t_k - 1}{t_{k+1}}\right)(\mathbf{R}_{k+1} - \mathbf{R}_k)$
end for
Assign the membership of trajectories using values of $\mathbf{R}_K$
as shown in Figure 2.
```

Algorithm 2 Local Motion Selection Algorithm

```
Input: Motion Clusters $\mathcal{C} = \{\mathcal{C}^c | c = 1, \ldots, N_c\}$, $\beta = (\beta_x, \beta_y)$.
Output: A set of local motion clusters $\mathcal{T}'$
$\mathcal{T}' = \emptyset$
for all $c \in \{1, \ldots, N_c\}$ do
    if $std(\mathcal{C}^c) \leq \beta$ and $|\mathcal{C}^c| > 1$ then
        $\mathcal{T}' = \mathcal{T}' \cup \mathcal{C}^c$.
    end if
end for
```

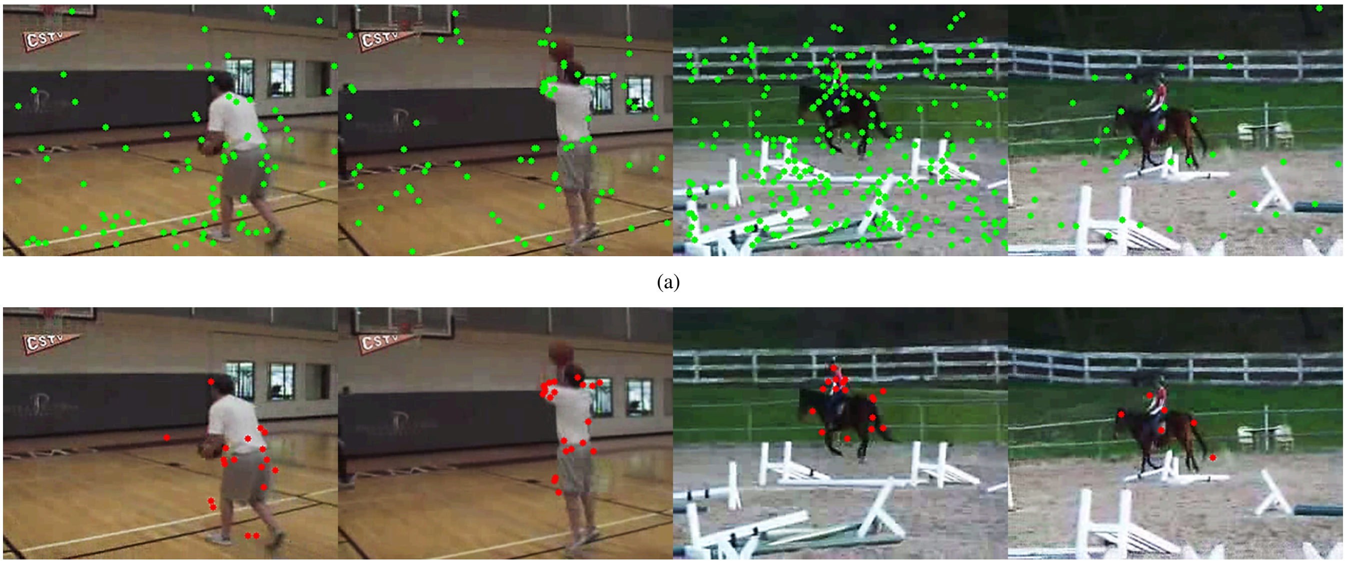
member trajectories of $\mathcal{C}^2 = \{\mathcal{T}_1, \mathcal{T}_4, \mathcal{T}_5, \mathcal{T}_6, \mathcal{T}_7\}$ are widely spread apart in the image plane. Therefore, $\mathcal{C}^2$ is regarded as a global motion cluster. Since $\mathcal{C}^3 = \{\mathcal{T}_8\}$ consists of a single trajectory, it is not considered as a local motion. Trajectories in $\mathcal{C}^1 = \{\mathcal{T}_2, \mathcal{T}_3\}$ are spatially concentrated around the key trajectory $\mathcal{T}_2$, hence, $\mathcal{C}^1$ is selected as a local motion cluster. The local motion selection process is described in Algorithm 2 and this local motion selection process is repeatedly performed at every frame.

Figure 3 shows examples of results from the local motion selection algorithm. Green points in Figure 3(a) represent positions of all trajectories. As shown in Figure 3, there are many trajectories generated by global motion. Although a bag of words (BoW) based approach is robust against outliers [19], it is clear that local motions characterizing an action are contaminated by indistinct global motions. Red points in Figure 3(b) represent positions of trajectories corresponding to local motion selected by Algorithm 2. As shown in Figure 3, the proposed local motion selection method can select important local motions.

4. Local Motion Emphasis and Kernel Group Sparse Representation

4.1. Local Motion Emphasis

We use the BoW approach which represents a video as an orderless distribution of visual words. We separately create a visual vocabulary for each descriptor type. We fix the number of visual words per descriptor to 1,000 or 4,000. To limit the complexity, we cluster a subset of 100,000 randomly selected training descriptors using k-means. We perform k-means five times with random initials and keep the result with the lowest error. Descriptors are assigned to their closest vocabulary word using the Euclidean distance. The resulting frequency histograms of visual word occurrences are used as features for action classification of a video clip. To preserve the information of local motion patterns, we separately create two types



(a)

(b)

Figure 3: Examples of local motion selection by Algorithm 2. Green points in (a) represent positions of full motion trajectories. Red points in (b) represent positions of local motion trajectories selected by the algorithm. (For interpretation of the references to color in this figure caption, the reader is referred to the web version of this article.)

of frequency histograms, i.e., full and local motions. To combine multiple frequency histograms, we use the multiple kernel method from [12]. Each frequency histogram for each descriptor type corresponds to one channel. That is, in our case, we have a total of eight channels and they are TrajShape, HOG, HOF, MBH descriptors of full and local motions, respectively.

The distinctive feature of our method is that, in addition to frequency histograms of full motion descriptors, we use frequency histograms of the selected local motion descriptors. Since full motion descriptors already include local motion descriptors, we are biasing the distance between samples by emphasizing the local motion information. By emphasizing local motion when calculating the distance between samples, it relatively alleviates the effect of global motion. We compare feature distributions using the exponential χ^2 distance with the multiple kernel method [12] as follows:

$$K(\mathbf{x}_i, \mathbf{x}_j) = \exp\left(-\frac{1}{N_c}\sum_{c=1}^{N_c}\frac{1}{\Omega_c}D_c(\mathbf{x}_i, \mathbf{x}_j)\right), \qquad (9)$$

where $D_c(\mathbf{x}_i, \mathbf{x}_j)$ is χ^2 distance [19] for channel c, and Ω_c is the mean value of χ^2 distances between training samples for the c-th channel.

4.2. Kernel Group Sparse (KGS) Representation for Classification

As shown in [6], sparse representation uses only basis vectors that can most compactly represent a signal and it is naturally discriminative. However, a general sparse representation has a limitation that it needs training samples to span the sample space well enough. It can perform poorly when there are a lot of intra-class variations. In this paper, we extend the group sparse representation approach to action recognition using the multiple kernel method.

Let $\mathbf{X} = [\mathbf{X}_1, \ldots, \mathbf{X}_J] \in \mathbb{R}^{m\times p}$ be a training feature matrix which is generated by concatenating training samples of J classes, i.e., $\mathbf{X}_1 \in \mathbb{R}^{m\times p_1}, \ldots, \mathbf{X}_J \in \mathbb{R}^{m\times p_J}$, where m is the dimensionality of a training sample, p_j is the number of training samples in class j, and $\sum_{j=1}^{J} p_j = p$ is the total number of training samples. Given a test sample $\mathbf{y} \in \mathbb{R}^m$, our group sparse representation is formulated as

$$\min_{\mathbf{w}} \frac{1}{2}\left\|\mathbf{y} - \sum_{j=1}^{J}\mathbf{X}_j\mathbf{w}_j\right\|_2^2 + \mu\sum_{j=1}^{J}\|\mathbf{w}_j\|_2, \qquad (10)$$

where $\mathbf{w}_j \in \mathbb{R}^{p_j}$, and μ is a regularization parameter. While (10) is similar to (4), there is a difference in the group structure. While a row of $\mathbf{R}$ makes one group in (4), the coefficient vector $\mathbf{w}_j$ for the j-th class in training samples makes one group in (10). We can solve (10) using the APG method as discussed in Section 3.1. However, this algorithm is developed for sparse representation of raw vector based features, while we have multiple features, i.e., TrajShapes, HOGs, HOFs, and MBHs. Hence, it requires a method to combine multiple features. For the purpose of combining multiple features, we modify the general APG method using the kernel trick as done in [7] for object recognition.

A kernel approach uses a non-linear kernel function $\phi(\cdot)$ to map training and test samples from the original space to a higher dimensional feature space. The kernel trick enable us to operate in the feature space by computing inner products using a kernel function, instead of performing operations in the high-dimensional feature space, i.e., $K(\mathbf{x}_i, \mathbf{x}_j) = \phi(\mathbf{x}_i)^T\phi(\mathbf{x}_j)$ for a given kernel function K. Many algorithms, such as a nonlinear SVM [20] and kernel principal component analysis [21], have used this kernel trick and demonstrated better performance compared to non-kernel methods. In this paper, we also apply the kernel trick to (10) with a kernel function $\phi(\cdot)$. It can be represented as

$$\min_{\mathbf{w}} \frac{1}{2} \left\| \phi(\mathbf{y}) - \sum_{j=1}^{J} \phi(\mathbf{X}_j)\mathbf{w}_j \right\|_2^2 + \mu \sum_{j=1}^{J} \|\mathbf{w}_j\|_2 , \qquad (11)$$

where $\phi(\mathbf{X}_j) = [\phi(\mathbf{X}_{j,p_1}), \dots, \phi(\mathbf{X}_{j,p_j})]$. When solving (11) using APG, there is a gradient mapping step, i.e., $\nabla f(\cdot) = -\phi(\mathbf{X})^T\phi(\mathbf{y}) + \phi(\mathbf{X})^T\phi(\mathbf{X})$ which involves inner products of features. We can straightforwardly apply the kernel trick here. Let $\mathbf{G} = \phi(\mathbf{X})^T\phi(\mathbf{X})$ with $\phi(\mathbf{X}) = [\phi(\mathbf{X}_1), \cdots, \phi(\mathbf{X}_J)]$ be the training kernel matrix, and $\mathbf{h} = \phi(\mathbf{X})^T\phi(\mathbf{y})$ be the test kernel vector. Then we can have $\nabla f(\cdot) = -\mathbf{h} + \mathbf{G}$ instead of inner products of features.

Using only the optimal coefficients $\widehat{\mathbf{w}}_j$ associated with the j-th class, one can approximate $\mathbf{y}$ of a test sample as $\phi(\mathbf{y}) = \phi(\mathbf{X}_j)\widehat{\mathbf{w}}_j$ and the reconstruction error using training samples in the j-th class is determined as

$$\begin{aligned} E_j = &\quad \|\phi(\mathbf{y}) - \phi(\mathbf{X}_j)\widehat{\mathbf{w}}_j\|_2^2 \\ = &\quad K_{\max} - 2\mathbf{h}_j\mathbf{w}_j + \mathbf{w}_j^T\mathbf{G}_j\mathbf{w}_j, \end{aligned} \qquad (12)$$

where $K_{\max}$ is the maximum value of the kernel function, $\mathbf{h}_j = \phi(\mathbf{y})^T\phi(\mathbf{X}_j)$ indicates elements of $\mathbf{h}$ associated with the j-th class, and $\mathbf{G}_j = \phi(\mathbf{X}_j)^T\phi(\mathbf{X}_j)$ is the block diagonal of $\mathbf{G}$ associated with the j-th class.

To use the spare representation method for video indexing and retrieval, we need a score function for a positive decision. For example, in some datasets in Section 5, each class is modeled by an independent binary classification problem, i.e., positive and negative classes, and the performance is evaluated by computing the average precision (AP), which approximates the area under a recall-precision curve for each action class [22, 23]. We define the score function as

$$f(E_j; E_{l\neq j}, \gamma) = \frac{K_{\max} - \min(E_j, \gamma)}{\sum_{l=1}^{J}(K_{\max} - \min(E_l, \gamma))}, \qquad (13)$$

where $\min(E_j, \gamma)$ is a truncated error function with $0 < \gamma < K_{\max}$ for limiting the maximum error and robust classification. The score function returns a relative score on the j-th class compared to all classes in training samples. That is, the decision score of the j-th class increases if the reconstruction error of the j-th class, E_j, decreases or the reconstruction error of the remaining classes, $E_{l\neq j}$, increases, and vice versa. Figure 4 shows an example of the score function for the positive class in a binary classification problem. As shown in this figure, the score is proportional to the reconstruction error of a negative class, E_N, and inversely proportional to the reconstruction error of a positive class, E_P, as expected. For a multi-class problem, the class with the highest score is chosen as a solution, i.e., $j^* = \arg\max_j f(E_j; \cdot)$. We demonstrate that the proposed score function shows good performance with group sparse representation in Section 5.

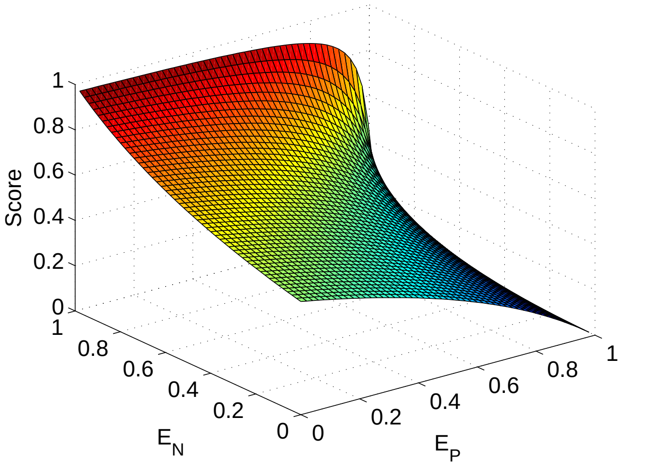


Figure 4: An example of a score function (13) for a binary classification. A binary classification can be done by thresholding the score. E_P is the reconstruction error using positive training samples and E_N is the reconstruction error using negative training samples.

5. Experiment

In this section, we demonstrate that the local motion selection and emphasis methods can improve the performance in dynamic action recognition and the group sparse representation with the multiple kernel method outperforms a nonlinear SVM method in most cases. We follow the experiment setup of [3] and extensively evaluate the proposed method on five popular benchmark datasets: Hollywood2 [22], Olympic Sports [13], UCF11 [24], UCF Sports [25], and KTH [26] datasets. We compare our method to the baseline method [3] and other state-of-the-art results. For experiments, we use default parameters as $N_\omega = 3$, $N_{min} = 3$, $N_{max} = 50$, $L = 15$, $N = 32$, $n_\sigma = 2$, $n_\tau = 3$, and the threshold α for removing a large displacement between two consecutive frames is 70% of the overall displacement of the trajectory. The sampling step size is set to $W = 10$ pixels, instead of $W = 5$ as in [3], because $W = 5$ is extremely dense and $W = 10$ does not make a big performance reduction as shown in [3]. For comparison, we report the results from [3] which is based on denser sampling with $W = 5$. The spatial threshold parameters (β_x, β_y) are chosen experimentally and they are set to 15% of the width and height of a frame, respectively. The regularization parameter κ in (4) is set to 0.1, μ in (10) is set to 0.001, and γ in (12) is set to 0.99.

We evaluate the performance of each individual descriptor, i.e., TrajShape, HOG, HOF, and MBH, and two different combinations of descriptors. The first combination (Comb. 1) is TrajShape, HOG, and MBH and the second combination (Comb. 2) is TrajShape, HOG, HOF, and MBH. Also, the proposed method can have three different types of descriptors, i.e., full motion descriptors, selected local motion descriptors, and emphasized motion descriptors obtained by combining local motion descriptors and full motion descriptors. We call them "F", "L", and "ME", respectively. Therefore, there are six different combinations when combined with two different classifi-

Table 1: Statistics of selected local motion descriptors

Dataset	No. of LMDs	No. of FMDs	ASR
Hollywood2	7,717,375	41,183,453	21.8%
Olympic Sports	2,349,940	15,705,422	21.6%
UCF11	2,524,486	14,976,537	26.9%
UCF Sports	470,461	1,998,232	32.2%
KTH	396,408	1,062,059	40.4%

cation methods, i.e., an SVM and kernel group sparsity (KGS) and they are F-SVM, L-SVM, ME-SVM, F-KGS, L-KGS, and ME-KGS. For example, “F-SVM” means that we have used full motion descriptors and an SVM classifier for action recognition.

5.1. *Statistics of Selected Local Motion Descriptors*

Before evaluating the performance, we investigate the portion of selected local motion descriptors compared to full motion descriptors. Some videos have a large number of global movements, e.g., Hollywood2 [22] and Olympic Sports [13] datasets. In this case, we uniformly select 50,000 trajectories among all trajectories obtained from the video and perform motion clustering. Table 1 shows a statistics of selected local motion descriptors on five datasets. In Table 1, the second column and the third column show the number of local motion descriptors (LMDs) selected by local motion selection and the number of full motion descriptors (FMDs) obtained from all samples in each dataset, respectively. We calculate an average selection ratio (ASR) of the number of selected local motion descriptors to the number of full motion descriptors obtained from each video. The third column shows ASRs on five datasets and we can find that the proposed local motion selection method selects only a small fraction of descriptors.

5.2. *Hollywood2 Dataset*

The Hollywood2 dataset [22] has been collected from 69 different Hollywood movies. There are 12 action classes: answering the phone, driving car, eating, fighting, getting out of car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up (see Figure 5). A total of 1,707 video sequences are divided into a training set (823 sequences) and a test set (884 sequences). Training and test sequences come from different movies. The performance is evaluated by computing the average precision (AP) for each action class and reporting the mean AP over all classes (mAP) as in [23].

We evaluate the performance with 1,000 visual words and 4,000 visual words, respectively. As shown in Table 2, ME-KGS, which is one of the proposed combinations, significantly outperforms F-SVM, regardless of the type of descriptors. Even some cases, the proposed method using local motion descriptors show better performance than F-SVM, despite the fact that only 21.8% of descriptors are used in our algorithm. It demonstrates that the proposed motion selection method can effectively select important local motions. Also, since the result shows that ME-SVM and ME-KGS are better than F-SVM and F-KGS, respectively, we can say that the proposed motion emphasis method makes the descriptors more robust. Looking at the impact according to classification methods, in most case the proposed KGS is better than an SVM, and ME-KGS is the best. Table 3 shows the performance for each action of ME-KGS and F-SVM with Comb. 2 and 4,000 visual words. Although the performance on “AnswerPhone” and “HugPerson” decreases, the amount of reduction is not significant. Surprisingly, ME-KGS outperforms the baseline method [3] which is obtained from more dense sampling with $W = 5$. As shown in Table 2, when we use the proposed KGS, the performance with Comb. 2 and 4,000 visual words in the local motion case is almost similar to that reported in [3] with $W = 5$. Considering the fact that we use $W = 10$, the results are very promising.

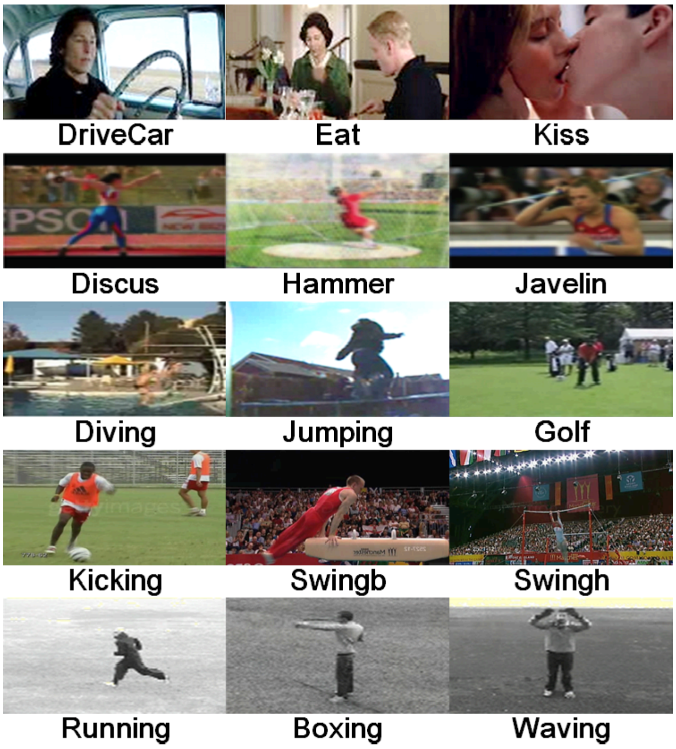


Figure 5: Examples of of Hollywood2 (first row), Olympic Sports (second row), UCF11 (third row), UCF Sports (fourth row), and KTH (last row) action datasets.

5.3. *Olympic Sports Dataset*

The Olympic Sports dataset [13] consists of athletes practicing different sports, which are collected from YouTube and annotated using Amazon Mechanical Turk. There are 16 sports actions: high-jump, long-jump, triple-jump, pole vault, gymnastics vault, shot put, snatch, clean-jerk, javelin throw, hammer throw, discus throw, diving platform, diving springboard, basketball lay-up, bowling, and tennis serve, represented by a total of 773 video sequences. We use 649 sequences for training and 134 sequences for testing as recommended by the authors [13]. We report the mean average precision over all classes (mAP) as in [23].

Table 2: Evaluation of the proposed method on the Hollywood2 dataset (mAP)

Descriptor	SVM, 1000			KGS, 1000			SVM, 4000			KGS, 4000		
	F(baseline)	L	ME	F	L	ME	F(baseline)	L	ME	F	L	ME
TrajShape	44.5	43.3	48.6	44.3	44.3	**49.5**	46.1	45.6	50.1	46.0	46.2	**51.0**
HOG	36.2	37.8	39.2	37.2	38.2	**39.8**	39.8	40.3	42.0	39.9	41.2	**43.2**
HOF	45.2	43.6	47.4	45.5	44.1	**48.1**	48.7	47.4	50.9	49.0	47.6	**51.3**
MBH	50.0	47.4	52.2	50.6	48.4	**52.8**	51.8	50.2	54.7	52.4	50.9	**55.7**
Comb. 1	53.9	54.1	57.0	55.1	55.8	**58.7**	55.0	55.3	57.8	56.1	57.6	**59.9**
Comb. 2	54.8	54.6	57.4	55.6	56.3	**59.2**	56.0	56.0	58.7	56.8	58.2	**60.5**

Table 3: Comparison to the method by Want et al. [3] on the Hollywood2 dataset for each action (AP for Comb. 2 and 4,000 visual words)

Action	ME-KGS ($W = 10$)	F-SVM ($W = 10$)	Wang et al. ($W = 5$) [3]	Action	ME-KGS ($W = 10$)	F-SVM ($W = 10$)	Wang et al. ($W = 5$) [3]
AnswerPhone	25.4	27.7	**32.6**	HugPerson	48.3	51.9	**54.2**
DriveCar	**93.5**	91.2	88.0	Kiss	65.7	64.5	**65.8**
Eat	**75.6**	68.2	65.2	Run	**85.9**	83.8	82.1
FightPerson	**83.9**	81.0	81.4	SitDown	**68.5**	63.2	62.5
GetOutCar	52.1	46.2	**52.7**	SitUp	**22.4**	4.2	20.0
HandShake	**34.6**	23.5	29.6	StandUp	**69.8**	66.4	65.2
				mAP	**60.5**	56.0	58.3

On this dataset, the proposed method shows more improvements than the Hollywood2 dataset. As shown in Table 4, in case of Comb. 1 with 1,000 visual words, ME-KGS outperforms F-SVM by 9.5% and with 4,000 visual words the improvement is 6.8%. Except for cases of HOG and Comb. 1 based on local motion, all combinations of the proposed methods show better performance than that of F-SVM and the decrease for HOG and Comb. 1 is only $0.1\% \sim 1\%$. Especially, considering that the number of local motion descriptors is small, i.e., 21.6%, and many video clips in this dataset are heavily polluted by camera motion, this result shows that the proposed methods are robust for a realistic video with a dynamic scene. Table 5 is the comparison of ME-KGS, F-SVM and the method by Niebles et al. [13], where Olympic Sports dataset is first used. ME-KGS achieves the best performance in terms of mAP and the performance gap between ME-KGS and Niebles's method is 10.7%.

5.4. *UCF11 Dataset*

The UCF11 dataset [24] contains 11 action categories: biking/cycling, diving, golf swinging, soccer juggling, trampoline jumping, horse riding, basketball shooting, volleyball spiking, swinging, tennis swinging, and walking with a dog (see Figure 5). This dataset is challenging due to large variations in camera motion, object appearance and pose, object scale, viewpoint, cluttered background, and illumination conditions. The dataset contains a total of 1,597 sequences. We follow the original setup [24] using the leave one out cross validation for a pre-defined set of 25 folds. Average accuracy over all classes is reported as a performance measure.

From Table 6, we can reach the same conclusion that the proposed motion selection and emphasis methods make descriptors robust and KGS significantly outperforms an SVM. Figure 6 shows confusion matrices on the UCF11 dataset. The confusion matrix in the left is obtained from F-SVM with Comb. 2 and 4,000 visual words and the confusion matrix in the right is obtained using ME-KGS with Comb. 2 and 4,000 visual words. ME-KGS improves the performance for *Riding*, *Biking*, *Spiking*, and *Tennis* classes which have a lot of global motion due to global movements of actors in a video. It demonstrates our claim that in general the information about local motion is more important than global motion, hence, local motion has to be emphasized for robust action recognition.

5.5. *UCF Sports Dataset*

The UCF Sports dataset [25] contains ten human actions: diving, golf swinging, kicking, weight-lifting, horse-riding, running, skateboarding, swinging on the pommel horse (swing-bench), swinging at the high bar (swing-side), and walking. The dataset consists of 150 video samples which show a large intra-class variability. To increase the amount of data samples, we extend the dataset by adding a horizontally flipped version of each sequence to the dataset. We train a multi-class classifier and report average accuracy over all classes. We use the leave-one-out setup, i.e., testing on each original sequence while training on all the other sequences together with their flipped versions.

Table 4: Evaluation of the proposed method on the Olympic Sports dataset (mAP)

Descriptor	SVM, 1000			KGS, 1000			SVM, 4000			KGS, 4000		
	F(baseline)	L	ME	F	L	ME	F(baseline)	L	ME	F	L	ME
TrajShape	54.7	58.7	60.1	59.5	64.8	**66.2**	56.4	60.2	61.5	61.9	64.8	**67.9**
HOG	65.6	63.8	64.8	67.2	68.9	**71.5**	65.4	64.4	66.9	70.1	69.1	**72.2**
HOF	53.5	57.9	56.9	55.8	60.7	**61.2**	54.3	57.9	56.9	57.4	**62.1**	60.7
MBH	67.5	70.2	71.6	69.8	**77.1**	76.0	67.8	71.3	71.1	71.9	**77.8**	76.6
Comb. 1	71.5	75.3	75.7	78.3	**81.9**	81.0	74.0	73.9	75.9	81.0	82.0	**82.8**
Comb. 2	70.4	74.7	73.2	77.3	**81.5**	80.7	71.7	73.3	74.9	79.6	80.9	**81.3**

Table 5: Comparison to the method by Niebles et al. [13] on the Olympic Sports dataset (AP for Comb. 1 and 4,000 visual words)

Action	ME-KGS	F-SVM	Niebles et al. [13]	Action	ME-KGS	F-SVM	Niebles et al. [13]
high jump	60.9	59.9	**68.9**	javelin throw	**100**	**100**	74.6
long jump	**95.2**	85.7	74.8	hammer throw	**84.6**	78.5	77.5
triple jump	**67.7**	60.7	52.3	discus throw	88.5	**91.1**	58.5
pole vault	64.9	63.8	**82.0**	diving platform	**98.9**	92.2	87.2
gymnastics vault	80.2	80.7	**86.1**	diving springboard	**100**	**100**	77.2
shot put	**68.9**	62.8	62.1	basketball layup	**97.1**	7.5	77.9
snatch	66.9	67.5	**69.2**	bowling	**77.0**	73.3	72.7
clean and jerk	77.9	73.4	**84.1**	tennis serve	**96.4**	87.2	49.1
				mAP	**82.8**	74.0	72.1

Note that the flipped version of the tested sequence is removed from the training set as in [3].

As can be seen in Table 7, the difference is rather small on UCF Sports, which is largely due to the leave-one-out setting, e.g., 298 videos are used for training and only two are used for testing. Nevertheless, with Comb. 2 and 4,000 visual words, ME-KGS outperforms F-SVM by 1.7 %. The best is 90.3% when we use L-KGS with Comb. 1 and 1,000 visual words. Interestingly, when comparing two confusion matrices of ME-KGS and F-SVM with Comb. 2 and 4,000 visual words, we can find that there is no performance degradation in all action classes as shown in Figure 7.

5.6. *KTH Dataset*

The KTH dataset [26] consists of six human action classes: walking, jogging, running, boxing, waving and clapping. Each action is performed several times by 25 subjects. The sequences were recorded in four different scenarios: outdoors, outdoors with scale variation, outdoors with different clothes, and indoors. The background is homogeneous and static in most sequences. In total, the dataset consists of 2,391 video samples. We follow the original experimental setup of the authors, i.e., divide the samples into a test set (9 subjects) and a training set (the remaining 16 subjects). As in the initial paper [26], we report average accuracy over all classes as a performance measure.

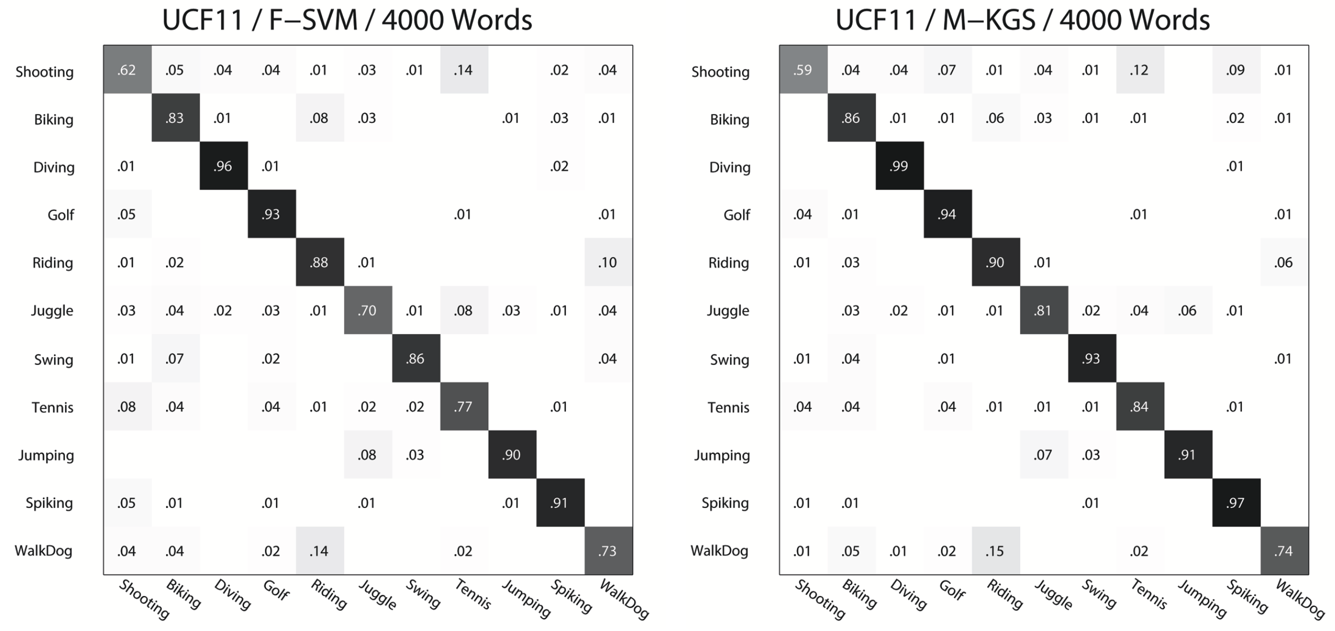
Unlike other four datasets, the performance gain by the proposed method is not significant (see Table 8). Since the KTH dataset has very similar artificial camera motions in both training and test samples and the background is clear, all descriptors obtained from a video have discriminative information. Therefore, selecting a small number of descriptors makes information loss and it causes a reduction in performance. Nevertheless, the local motion emphasis which uses local motion descriptors with full motion descriptors can prevent the information loss. The proposed motion emphasis method alleviates the risk coming from incomplete motion separation and it makes action recognition more robust.

5.7. *Comparison to Other State-of-the-Art Algorithms*

In this section, we compare our results using ME-KGS with Comb. 2 and 4,000 words to the state-of-the-art algorithm for each dataset. We can observe that the proposed method outperforms the state-of-the-art algorithms on all datasets except KTH. Table 9 shows mAP reported in the literature. On the Hollywood2 dataset, we obtain 2.2% gain over [3]. This performance gain is not trivial considering that we use descriptors obtained from trajectories at a sparser sampling rate. On the Olympic Sports dataset, we attain better results than all the state-of-the-arts and the improvement is 4% (when we use Comb. 1, a significant gain of 5.5% is achieved as shown in Table 4). Table 10 shows average accuracies reported in the

Table 6: Evaluation of the proposed method on the UCF11 dataset (Average Accuracy)

Descriptor	SVM, 1000			KGS, 1000			SVM, 4000			KGS, 4000		
	F(baseline)	L	ME	F	L	ME	F(baseline)	L	ME	F	L	ME
TrajShape	65.9	68.9	**70.6**	64.5	66.9	70.5	65.5	67.6	**69.5**	65.1	66.9	69.2
HOG	72.8	73.5	75.4	73.7	75.9	**77.1**	74.0	74.6	76.1	75.0	77.2	**78.5**
HOF	68.7	72.2	72.2	69.9	72.7	**73.5**	70.3	72.4	72.5	72.7	74.0	**75.7**
MBH	76.7	77.9	79.6	80.1	79.2	**82.2**	77.2	77.5	79.2	80.8	80.1	**81.3**
Comb. 1	83.2	85.9	86.3	84.7	87.3	**88.0**	83.3	84.4	85.0	85.6	86.8	**87.3**
Comb. 2	82.7	84.5	85.9	84.5	86.4	**86.6**	82.6	83.2	84.5	84.8	**86.2**	86.1



UCF11 / F–SVM / 4000 Words

	Shooting	Biking	Diving	Golf	Riding	Juggle	Swing	Tennis	Jumping	Spiking	WalkDog
Shooting	.62	.05	.04	.04	.01	.03	.01	.14		.02	.04
Biking		.83	.01		.08	.03			.01	.03	.01
Diving	.01		.96	.01						.02	
Golf	.05			.93				.01			.01
Riding	.01	.02			.88	.01					.10
Juggle	.03	.04	.02	.03	.01	.70	.01	.08	.03	.01	.04
Swing	.01	.07		.02			.86				.04
Tennis	.08	.04		.04	.01	.02	.02	.77		.01	
Jumping						.08	.03		.90		
Spiking	.05	.01		.01		.01			.01	.91	
WalkDog	.04	.04		.02	.14			.02			.73

(a)

UCF11 / M–KGS / 4000 Words

	Shooting	Biking	Diving	Golf	Riding	Juggle	Swing	Tennis	Jumping	Spiking	WalkDog
Shooting	.59	.04	.04	.07	.01	.04	.01	.12		.09	.01
Biking		.86	.01	.01	.06	.03	.01	.01		.02	.01
Diving			.99							.01	
Golf	.04	.01		.94				.01			.01
Riding	.01	.03			.90	.01					.06
Juggle		.03	.02	.01	.01	.81	.02	.04	.06	.01	
Swing	.01	.04		.01			.93				.01
Tennis	.04	.04		.04	.01	.01	.01	.84		.01	
Jumping						.07	.03		.91		
Spiking	.01	.01					.01			.97	
WalkDog	.01	.05	.01	.02	.15			.02			.74

(b)

Figure 6: Two confusion matrices on the UCF11 dataset with Comb. 2 and 4,000 visual words. By comparing two confusion matrices, we can observe that ME-KGS improves the performance for *Riding*, *Biking*, *Spiking*, and *Tennis* classes.

literature. On the UCF11 dataset, we outperform the state-of-the-art [3] by over 1.9% (when we use Comb. 1, a significant gain of 3.8% is achieved as shown in Table 6). On the UCF Sports dataset, the proposed method achieves 89.7% which is 1.5% better than the state-of-the-art [3]. On this dataset, our best performance is 90.3% with L-KGS using Comb. 1 and 1,000 words and the gain over [3] is 2.1%. Considering that the experiment setting is the leave-one-out setting, the gain is significantly meaningful. On the KTH dataset, the best is [27], but the difference is only 0.3%.

Table 9: Comparison to other existing algorithms (mAP)

Hollywood2		Olympic Sports	
Taylor et al. [28]	46.6	Laptev et al. [29]	62.0
Wang et al. [19]	47.7	Niebles et al. [13]	72.1
Gilbert et al. [27]	50.9	Liu et al. [30]	74.4
Wang et al. [3]	58.3	Brendel et al. [31]	77.3
Our method	**60.5**	Our method	**81.3**

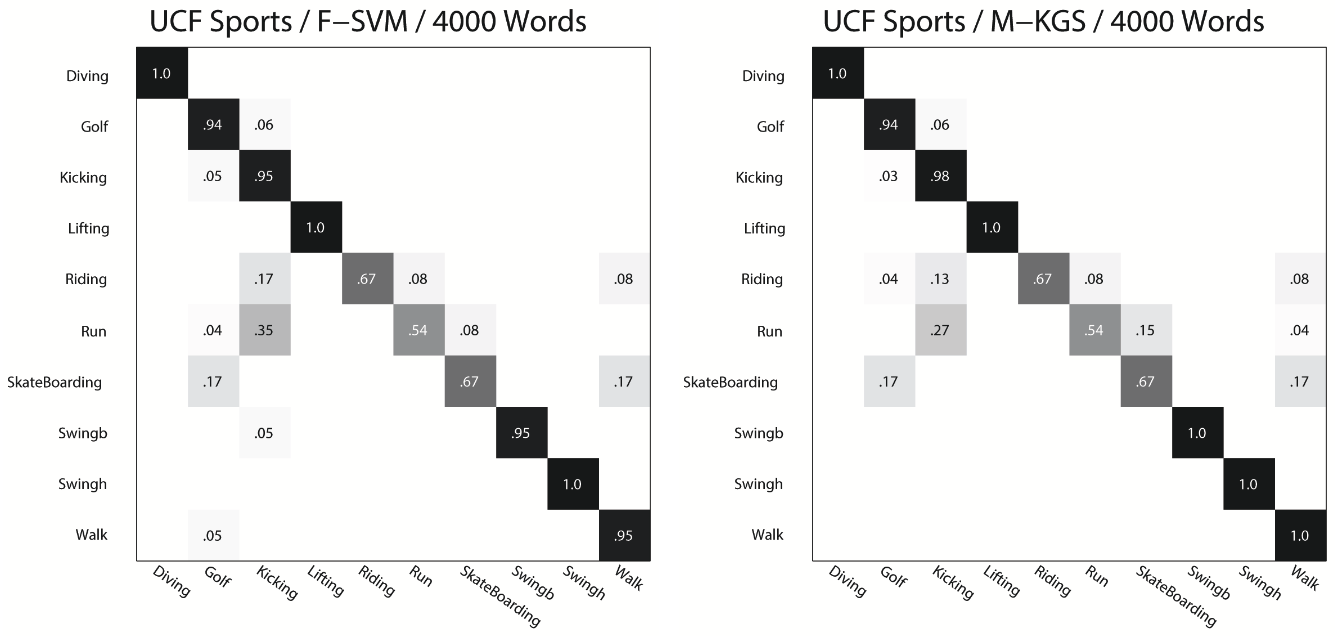
5.8. Computational Complexity

Table 11 shows a detailed analysis of the computation time of the proposed algorithm on the UCF Sport dataset with Comb. 2 and 4,000 words. Since the same descriptors are used in both the baseline method and the proposed method, we focus on the time spent for four major steps: generation of a full motion histogram (Hist. for FMDs), local motion selection (LMS), generation of a local motion histogram (Hist. for LMDs), and performance evaluation (Classification) using the leave-one-out setup explained in Section 5.5. We implemented the proposed algorithm using MATLAB on a PC with a 3.4GHz quad-core Intel i7-2600 CPU. Also, we used a single core of the CPU for better estimation of computation times.

As shown in Table 11, the relative time consumption for

Table 7: Evaluation of the proposed method on the UCF Sports dataset (Average Accuracy)

Descriptor	SVM, 1000			KGS, 1000			SVM, 4000			KGS, 4000		
	F(baseline)	L	ME	F	L	ME	F(baseline)	L	ME	F	L	ME
TrajShape	73.3	75.0	76.0	74.3	75.0	**76.3**	71.0	75.0	75.0	76.7	**78.3**	77.0
HOG	81.7	82.0	83.3	82.0	82.3	**84.7**	81.7	81.7	82.7	83.0	84.3	**85.0**
HOF	78.7	79.7	80.7	78.3	83.3	**83.7**	78.3	76.3	81.0	80.0	81.0	**81.7**
MBH	83.3	87.7	87.0	87.0	**88.0**	87.3	84.0	84.3	85.0	85.3	**88.3**	87.3
Comb. 1	87.3	86.7	89.0	86.7	89.3	**90.0**	86.7	86.7	86.3	88.0	88.7	**89.0**
Comb. 2	88.0	87.3	88.7	87.3	**90.3**	89.7	88.0	86.0	87.7	88.7	88.7	**89.7**



(a)

(b)

Figure 7: Two confusion matrices on the UCF Sports dataset with Comb. 2 and 4,000 visual words. Interestingly, when comparing two confusion matrices of ME-KGS and F-SVM, we can find that there are no performance degradation in all action classes. (Swingb indicates swing-bench and Swingh indicates swing-side.)

classification is negligible. Although the proposed motion emphasis method requires two additional steps for local motion selection and a local motion histogram, a local motion histogram can be computed simultaneously with the generation of a full motion histogram and the extra computation time is only $0.03s$ for a sample. Therefore, only additional cost of our algorithm compared to the baseline algorithm [3] is for the local motion selection step and it costs about 16.08% of the total computation time. In addition, if only local motion descriptors are used, the average computation time for the histogram generation can be reduced from $172.95s$ to $44.25s$, making the proposed method faster than [3] even if we consider the time required for local motion selection ($33.17s$). Note that the proposed method using only local motion features still outperforms the baseline algorithm and competes well against other state-of-the-art algorithms.

6. Conclusion

In this paper, we have proposed a novel action recognition method for a video with a dynamic scene. To recognize an action in a dynamic scene, it is important to emphasize local motion, since camera motion can dilute the motion characterizing the action of interest. The proposed algorithm automatically selects a small number of descriptors corresponding to local motion using group sparsity and emphasizes them by the multiple kernel method. For robust classification, we have applied the group sparse representation with the multiple kernel method. Through extensive experiments, we have demonstrated that the proposed local motion selection and emphasis can improve the performance of action recognition by correcting the distance between samples contaminated by global motion. In most cases, the selected local motion descriptors show better performance than that of using full motion descriptors. Fur-

Table 8: Evaluation of the proposed method on the KTH dataset (Average Accuracy)

Descriptor	SVM, 1000			KGS, 1000			SVM, 4000			KGS, 4000		
	F(baseline)	L	ME	F	L	ME	F(baseline)	L	ME	F	L	ME
TrajShape	**88.6**	84.5	88.2	**88.6**	86.0	88.3	86.8	84.9	86.8	87.7	87.0	**87.9**
HOG	82.6	81.8	83.2	83.8	82.4	**84.5**	84.0	84.5	85.0	86.4	85.5	**87.6**
HOF	91.3	87.7	90.1	**92.0**	88.6	90.6	92.2	90.0	**92.7**	92.3	89.7	91.8
MBH	93.0	90.6	93.0	**94.3**	90.8	93.5	94.2	92.6	94.3	**94.4**	92.6	94.0
Comb. 1	92.8	92.5	92.9	93.7	92.9	**94.0**	93.4	93.0	93.0	93.3	**94.1**	93.7
Comb. 2	93.2	92.1	92.8	92.8	92.8	**93.6**	93.4	93.0	93.2	93.3	93.4	**94.2**

Table 10: Comparison to other existing algorithms (Average Accuracy)

UCF11		UCF Sports		KTH	
Liu et al. [24]	71.2	Wang et al. [19]	85.6	Laptev et al. [29]	91.8
Ikizler-Cinbis et al. [4]	75.21	Kovashke et al. [32]	87.3	Yuan et al. [33]	93.3
Wang et al. [3]	84.2	Kiäser et al. [34]	86.7	Gilbert et al. [27]	**94.5**
		Wang et al. [3]	88.2	Wang et al. [3]	94.2
Our method	**86.1**	Our method	**89.7**	Our method	94.2

thermore, the classification method using the group sparse representation with the multiple kernel method dramatically improves the action recognition performance.

7. Acknowledgement

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (NRF-2013R1A1A2065551).

References

[1] R. Messing, C. Pal, H. Kautz, Activity recognition using the velocity histories of tracked keypoints, in: Proc. of IEEE International Conference on Computer Vision, 2009.

[2] J. Sun, X. Wu, S. Yan, L.-F. Cheong, T.-S. Chua, J. Li, Hierarchical spatio-temporal context modeling for action recognition, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[3] H. Wang, A. Klaser, C. Schmid, C.-L. Liu, Action recognition by dense trajectories, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[4] N. Ikizler-Cinbis, S. Sclaroff, Object, scene and actions: Combining multiple features for human action recognition, in: Proc. of European Conference on Computer Vision, 2010.

[5] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2005.

[6] J. Wright, A. Yang, A. Ganesh, S. Sastry, Y. Ma, Robust face recognition via sparse representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 31 (2) (2009) 210–227.

[7] X.-T. Yuan, S. Yan, Visual classification with multi-task joint sparse representation, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[8] T. Guha, R. Ward, Learning sparse representations for human action recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (8) (2012) 1576–1588.

[9] B. A. Olshausen, D. J. Field, Emergence of simple-cell receptive field properties by learning a sparse code for natural images, Nature 381 (6583) (1996) 607–609.

[10] W. Deng, W. Yin, Y. Zhang, Group sparse optimization by alternating direction method, Technical Report 11-06, Department of Computational and Applied Mathematics, Rice University, 2011.

[11] A. Beck, M. Teboulle, A fast iterative shrinkage-thresholding algorithm for linear inverse problems, SIAM Journal on Imaging Sciences 2 (1) (2009) 183–202.

[12] J. Zhang, M. Marszaek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, International Journal of Computer Vision 73 (2) (2007) 213–238.

[13] J. C. Niebles, C.-W. Chen, L. Fei-Fei, Modeling temporal structure of decomposable motion segments for activity classification, in: Proc. of European Conference on Computer Vision, 2010.

[14] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. of International Joint Conference on Artificial Intelligence, 1981.

[15] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[16] J. Shi, J. Malik, Normalized cuts and image segmentation, IEEE Transactions on Pattern Analysis and Machine Intelligence 22 (8) (2000) 888–905.

[17] G. Farneback, Two-frame motion estimation based on polynomial expansion, in: Proc. of Scandinavian Conference on Image Analysis, 2003.

[18] Y. Cong, J. Yuan, J. Liu, Sparse reconstruction cost for abnormal event detection, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[19] H. Wang, M. M. Ullah, A. Klaser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, in: Proc. of British Machine Vision Conference, 2009.

[20] C.-C. Chang, C.-J. Lin, Libsvm: A library for support vector machines, ACM Transactions on Intelligent Systems and Technology 2 (3) (2011) 27:1–27:27.

[21] B. Scholkopf, A. Smola, K.-R. Muller, Nonlinear component analysis as a kernel eigenvalue problem, Neural Computation 10 (5) (1998) 1299–1319.

[22] M. Marszałek, I. Laptev, C. Schmid, Actions in context, in: IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[23] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, A. Zisserman, The PASCAL Visual Object Classes Challenge

Table 11: Computation time of the proposed algorithm (ME-KGS)

UCF Sport dataset with Comb. 2 and 4,000 words	Hist. for FMDs (LMDs)	LMS	Classification	Hist. for LMDs
Computation time for all samples (sec)	51885.82 (8.15)	9951.62	55.78	13275.94
Average computation time per sample (sec)	172.95 (0.03)	33.17	0.19	44.25
Percentage (%)	83.83 (0.01)	16.08	0.09	-

2011 (VOC2011) Results, `http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html`.

[24] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos in the wild, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[25] M. Rodriguez, J. Ahmed, M. Shah, Action mach: a spatio-temporal maximum average correlation height filter for action recognition, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[26] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local svm approach, in: Proc. of International Conference on Pattern Recognition, 2004.

[27] A. Gilbert, J. Illingworth, R. Bowden, Action recognition using mined hierarchical compound features, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 883–897.

[28] G. W. Taylor, R. Fergus, Y. LeCun, C. Bregler, Convolutional learning of spatio-temporal features, in: Proc. of European Conference on Computer Vision, 2010.

[29] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning realistic human actions from movies, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2008.

[30] J. Liu, B. Kuipers, S. Savarese, Recognizing human actions by attributes, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2011.

[31] W. Brendel, S. Todorovic, Learning spatiotemporal graphs of human activities, in: Proc. of IEEE International Conference on Computer Vision, 2011.

[32] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2010.

[33] J. Yuan, Z. Liu, Y. Wu, Discriminative subvolume search for efficient action detection, in: Proc. of IEEE Conference on Computer Vision and Pattern Recognition, 2009.

[34] A. Klaser, M. Marszałek, I. Laptev, C. Schmid, et al., Will person detection help bag-of-features action recognition?, Technical Report, INRIA Grenobel-Rhone-Alpes, 2010.

About the Author- Jungchan Cho received the B.S. degree in the School of Electrical and Electronics Engineering from Chung-Ang University, Seoul, Korea, in 2010. He is currently pursuing the Ph.D. degree in the Department of Electrical and Computer Engineering at the Seoul National University. His research interests include pattern recognition and computer vision.

About the Author- Minsik Lee received the B.S. and Ph.D. degrees from the School of Electrical Engineering and Computer Science, Seoul National University, Korea in 2006 and 2012, respectively. He is currently a senior research engineer in ASRI, Seoul National University. His research interests include 3-D reconstruction, deformable models, pattern analysis, and their applications.

About the Author- Hyung Jin Chang received the B.S. and Ph.D. degrees from Seoul National University, Seoul, Korea, in 2006 and 2013, respectively. He is currently a research associate at Department of Electrical and Electronic Engineering of Imperial College London. His research interests include pattern recognition, moving object detection, object recognition and action recognition.

About the Author- Songhwai Oh is an associate professor in the Department of Electrical and Computer Engineering at the Seoul National University. He received all his degrees in Electrical Engineering and Computer Sciences (EECS) from UC Berkeley (B.S. with highest honors in 1995, M.S. in 2003, and Ph.D. in 2006). His research interests include cyber-physical systems, wireless sensor networks, robotics, estimation and control of stochastic systems, multi-modal sensor fusion, and machine learning.