Motion Part Regularization: Improving Action Recognition via Trajectory Group Selection

Bingbing Ni
ADSC Singapore
bingbing.ni@adsc.com.sg

Pierre Moulin
UIUC USA
moulin@ifp.uiuc.edu

Xiaokang Yang
SJTU China
xkyang@sjtu.edu.cn

Shuicheng Yan
NUS Singapore
eleyans@nus.edu.sg

Abstract

Dense local trajectories have been successfully used in action recognition. However, for most actions only a few local motion features (e.g., critical movement of hand, arm, leg etc.) are responsible *for the action label. Therefore, highlighting the local features which are associated with important motion parts will lead to a more discriminative action representation. Inspired by recent advances in sentence regularization for text classification, we introduce a* Motion Part Regularization *framework to mine for discriminative groups of dense trajectories which form important motion parts. First, motion part candidates are generated by spatio-temporal grouping of densely extracted trajectories. Second, an objective function which encourages sparse selection for these trajectory groups is formulated together with an action class discriminative term. Then, we propose an alternative optimization algorithm to efficiently solve this objective function by introducing a set of auxiliary variables which correspond to the discriminativeness weights of each motion part (trajectory group). These learned motion part weights are further utilized to form a discriminativeness weighted Fisher vector representation for each action sample for final classification. The proposed motion part regularization framework achieves the state-of-the-art performances on several action recognition benchmarks.*

1. Introduction

Video based action recognition is a challenging task due to large variation of human posture/movement and significant background/irrelavant motions. Spatio-temporal local motion features, such as spatio-temporal interest points (STIPs) [10], HOG3D [7], Cuboids [2] and especially the recent prevailing dense trajectories [23, 25], have shown great discriminative capability in action recognition. Typically, hundreds of thousands of dense local trajectories are extracted from a given video clip. After feature extraction, a bag-of-words [23] or Fisher vector [25] action representation vector is calculated by pooling the entire video. However, this processing pipeline has obvious drawbacks. For most actions, only a small subset of local motion features out of the entire video is relevant to the action label. For example, when a person is waving his hand while walking, only the movement around the arm/hand is *responsible* for the action *waving hand*. Another example is when there exist severe background motions, it would be demanding that the video-level action representation should make the foreground motion highlighted. Unfortunately, in either bag-of-words or Fisher vector, redundant and noisy features extracted over the entire video sequence may dominate the representation and downgrade the discriminative capability.

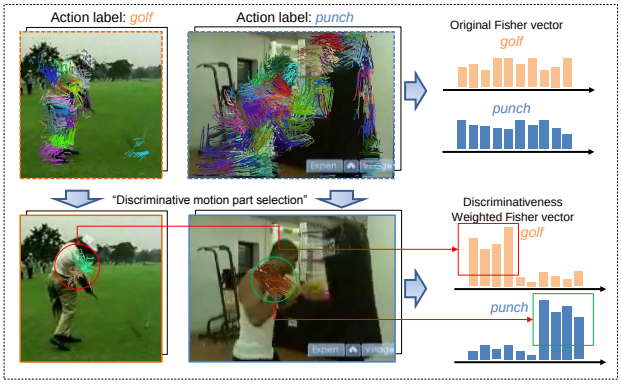

Figure 1. Motivation of our work. Note that our motion part regularization framework generate discriminativeness weighted Fisher vector representation, which is more discriminative than the unweighted traditional Fisher vector.

Based on the latent structural model [4], several local motion feature selection methods have been proposed for action recognition. Wang et al. [31] proposed a max-margin hidden conditional random field framework to select discriminative spatio-temporal interest points by considering the spatial relationship among the local features. Roo and Aggarwal [18] matched two local motion feature graphs for action detection. Graph-based approach is also adopted for

video annotation and classification [30, 29]. To recognize action, Yuan et al. [33] searched for video volumes that contain discriminative STIP features. Raptis et al. [17] utilized a latent model to select discriminative clusters of motion trajectories. Kong et al. [8] proposed to select interaction phrases for human interaction recognition. However, these selection method have several limitations. First, the local features selected by these methods are very sparse (*i.e.*, often less than 10 local features or feature clusters for each video sample). This is partly because of the fact that current optimization scheme for solving a latent structural SVM problem is very costly if the number of latent variables is large. As a consequence, realistic action videos have very large variations and reasonable action representation cannot be achieved with such sparse local features. Moreover, even a small amount of irrelevant background motion or variation of human movement will cause local feature constellation based action models to fail. A good evidence is that the action classification accuracies achieved by these selection methods [31, 17] on benchmarks are not comparable to those resulting from bag-of-words or Fisher vector representations based on dense trajectories [23, 25]. Second, training a good latent structural SVM model heavily depends on a good initialization or latent part annotation. Third, in some scenarios, full body motion might not be completely useless. For example, the statistics of the whole body motion features can be used to recognize actions such as *walking*, *running*, etc.

Motivated by these observations, we believe that a good trade-off between the above two types of methods, *i.e.*, global uniform pooling of dense local motion features or selecting a sparse set of discriminative local features, can achieve better action recognition performance. Namely, instead of selecting some local features, we can *measure* the discriminative score of each local feature and rely on these scores to pool the local features in a weighted manner. For such a purpose, we first cluster dense trajectories into spatio-temporal groups, which are called *motion parts* in this work. Then, we propose a simple yet effective learning approach which can select discriminative motion parts in a soft manner, *i.e.*, to assign a weight to each motion part to indicate its discriminativeness and use these learned weights for more discriminative action representation by attenuating the of effect of irrelevant motion parts. Our method is inspired by the recent work called *Sentence Regularization* for document classification [32]. In [32], the key observation is that the text words in only a few sentences are relevant to the document label, which is very similar to the action recognition scenario: only a few motion parts that are associated with important moving body parts such as hand, arm, leg, etc., convey high discriminative information. In this sense, each motion part can be regarded as a *sentence* that contains a set of local trajectory features which are indexed into some visual words. For an action video, this results in tens of thousands of motion parts (*sentences*), and the visual words within these sentences are shared, i.e., we have overlapping groups of visual words. To select discriminative local motion part/trajectory group, for each local motion feature in each group we introduce an auxiliary variable, which can be regarded as a local copy of the global weight of the visual word it belongs to. The introduction of these local copies helps to convert the overlapping group lasso feature selection problem (which is not efficient to solve) to a non-overlapping group lasso problem. We thus introduce a simple yet effective alternative optimization scheme to simultaneously optimize the global classifier model weights associated with the visual words and the local copies of these weights. Finally, we utilize these learned weights of local motion parts to compute the discriminativeness weighted action representation for each action video. The motivation of our method is illustrated in Figure 1. Experiments on several benchmarks including Hollywood2 [12], HMDB51 [9], and Olympic Sports [14] show that the proposed method improves action recognition performance.

The rest of this paper is organized as follows. First, we discuss the related work in Section 2. Section 3 presents the proposed motion part regularization framework and our action representation pipeline. Extensive experimental results and discussions on several benchmark datasets are given in Section 4. Section 5 concludes the paper.

2. Related Work

Besides selecting information local motion features, several researchers have explored the idea of selecting the most informative spatio-temporal volumes for human action recognition, mostly based on the latent structural SVM model [4]. Satkin and Hebert [19] proposed to select the most discriminative temporal cropping of training videos to improve action recognition performance. The best temporal cropping for each training video is inferred by iteratively mining data using a leave-one-video-out scheme followed by a latent structural SVM refinement. In Duchenne et al. [3], video is segmented into overlapping spatio-temporal volumes and a sliding-window SVM detector is utilized for action detection. Shi et al. [20] proposed a discriminative action segmentation and recognition using semi-Markov model. Niebles et al. [14] represented an activity as temporal compositions of motion segments. The entire video volume is first decomposed into several temporal sub-volumes and bag of STIP [10] features are computed within each sub-volume. A query video is matched to the best video sub-volume from the training videos. Tang et al. [21] developed variable-duration hidden Markov models for partitioning the video into variable-length temporal sub-volumes. A max-margin latent structural SVM is uti-

lized to automatically discover discriminative temporal partitions. Recently, Ni et al. [13] utilized the human key pose information to adaptively select the best video sub-volume for action recognition. Latent structural SVM model is also utilized.

On one hand, these methods mostly select spatio-temporal video sub-volumes and calculate the bag-of-word representations within the selected sub-volumes. However, these sub-volumes might not well correspond to informative moving parts of human, e.g., hand, arm, leg, head etc. In contrast, spatio-temporally grouped dense trajectory clusters (motion part) are more related to semantic parts of the human body. On the other hand, most of these methods are based on latent structural SVM model, and as mentioned in the introduction, the quality of the learned model heavily relies on model initialization. In contrast, our proposed training algorithm is simple, effective and stable which does not depend on the initialization. Also, these methods use sparse features/feature groups, which is less discriminative than dense sampling based methods like ours. Wang et al. [26] mine discriminative motion groups associated with human skeleton joints for RGB-D action recognition, however, this method heavily rely on the extra skeleton joints information.

3. Action Recognition via Motion Part Regularization

We introduce the details of our proposed action recognition pipeline based on motion part regularization in this section. The pipeline includes motion part generation, discriminative motion part selection via motion part regularization, and discriminativeness weighted action representation formulation based on the learned motion part weights.

We begin with the notations used in this paper. Assume we are given D video samples. For the video sample d, we denote the set of local motion feature vectors (*e.g.*, MBH, HOF, HOG, or trajectory shape descriptor for dense trajectories [23, 25]) as $\{\mathbf{x}_i\}_{i=1:N_d}$ where N_d is the number of local features extracted in the video sample d. In the bag-of-words context, each local feature is indexed to a visual word and we assume the size of the visual word vocabulary (dictionary) is V. In this work, we use $V = 4000$ visual words. Each video sample d is therefore represented as a V-dimensional vector of word occurrence frequencies $\mathbf{x}_d$, known as video level representation. The label associated with video sample d is denoted as y_d. In a binary classification (action detection) setting, $y_d \in \{+1, -1\}$.

3.1. Motion Part Generation

Our goal is to select discriminative mid-level visual parts to represent action. The mid-level representations are expected to correspond to semantic parts/regions of the human body which are conducting important movement related to the action label. Although there exist other options for mid-level action representations such as densely sampled video sub-volumes [27, 15], we believe that the dense trajectory cluster/group proposed in [17] is a more appropriate choice, since the dense trajectories within a local cluster present very similar motion characteristics and they are spatio-temporally very close to each other (most likely belong to the same body part).

Following [17], we first extract dense trajectories over the entire video volume [23]. We follow the default settings of [23] for dense trajectory extraction. To partition/cluster the dense local trajectories into a set of semi-local trajectory clusters, we compute the similarity among trajectory pairs and form an $N_d \times N_d$ affinity matrix for a video that contains N_d trajectories. We use the same distance measure for a pair of trajectories as in [17] by considering their temporal overlapping, spatial proximity and speed similarity. Following [17], we also enforce the affinity to be zero for trajectory pairs that are not spatially close (*i.e.*, distance larger than some threshold). We then apply the graph based clustering method used in [17] to partition the trajectories into groups. To minimize the risk of missing important motion parts, we run the partition algorithm using four scales (*i.e.*, maximum allowed number of trajectories within each cluster/group) to form trajectory clusters with different average sizes. We name each trajectory group as a *Motion Part*, *i.e.*, the trajectories within such a group is spatially nearby, temporally overlapping and present very similar motion characteristics.

3.2. Motion Part Regularization

We assume a linear classifier model, which is applied to predict the class label of d-th video sample:

$$f = \mathbf{w}^T \mathbf{x}_d + b. \tag{1}$$

Here $\mathbf{x}_d$ is the V-dimensional bag-of-words (histogram) representation of video sample d (we overload the notation of $\mathbf{x}$ to denote both local feature descriptor and video level representation). $\mathbf{w}$ is the corresponding V-dimensional classifier model coefficients and b is a bias. The primary goal of classifier learning is to estimate the optimal model parameter $\mathbf{w}$ which minimizes some pre-defined loss function. Although there are various options for the loss function, a very simple yet common choice is the logistic regression function:

$$\mathcal{L}(\mathbf{w}, b) = -\sum_{d=1}^{D} \log\Big(1 + \exp\big(-y_d(\mathbf{w}^T \mathbf{x}_d + b)\big)\Big). \tag{2}$$

Note that for a normalized input $\mathbf{x}_i$, the absolute value of each entry of the model coefficients $\mathbf{w}$ naturally *measures* the discriminative capability of its corresponding visual word in classification. If the j-th entry of $\mathbf{w}$ (w_j) has

a high value, it indicates that the corresponding visual word j, $j \in \{1, \cdots, V\}$ is very discriminative for the underlying classification task.

Besides the empirical loss minimization term $\mathcal{L}$, usually we also employ some regularization penalty to endow the solution with some specific property. For example, the well-known ℓ_1 regularizer enforces a sparse solution, *i.e.*, to encourage the number of nonzero entries of $\mathbf{w}$ to be small:

$$\Omega_l(\mathbf{w}) = \|\mathbf{w}\|_1. \tag{3}$$

In the case that features (*i.e.*, the entries of the histogram vector $\mathbf{x}_d$) can be assigned into groups, a group sparsity penalty encourages all of the weights in a group to either be zero or nonzero, which is known as group lasso [34]. We can represent the group lasso regularizer as:

$$\Omega_{gl}(\mathbf{w}) = \sum_{g=1}^{G} \|\mathbf{w}_g\|_2, \tag{4}$$

where we assume the entries of $\mathbf{w}$ form G groups.

Recalling that the objective in this work is to select discriminative motion parts (trajectory groups), we show that our problem well fits into the group sparsity (group lasso) regularization framework. In particular, dense trajectories are clustered into groups (motion parts), and selecting discriminative motion parts is equivalent to enforcing that only a few trajectory groups are selected during the classifier learning. Namely, some trajectory groups receive high weights and others recieve lower weights. A key observation is that our trajectory groups are formed *locally* and therefore they are heavily overlapping. In other words, as the visual word vocabulary is globally defined, each visual word may occur in many motion parts (trajectory groups). Mathematically, we use d to index over video samples and p to index over motion parts (trajectory groups) within a video sample. We further denote by P_d the number of motion parts in action video d. Equation (4) can be expanded as:

$$\Omega_{gl}(\mathbf{w}) = \sum_{d=1}^{D} \sum_{p=1}^{P_d} \|\mathbf{w}_{d,p}\|_2, \tag{5}$$

where $\mathbf{w}_{d,p}$ corresponds to the sub-vector of $\mathbf{w}$ such that the corresponding features (visual words) are present in motion part p of video d, *i.e.*, different $\mathbf{w}_{d,p}$ vectors may have heavy overlap.

Although there exist many general solvers for the above overlapping group lasso problem, they might not perform efficiently in our problem, since we have hundred to thousands of motion parts generated in one video. Moreover, if we sum up over the entire training video set, the total number of overlapping groups could be more than one million. To address this issue, inspired by [32], the key idea is to introduce a set of auxiliary variables $\mathbf{v}$ to *de-overlap* the groups $\{\mathbf{w}_{d,p}\}$.

Each entry of $\mathbf{v}$ defines a weight for each local trajectory feature, thus the length of the vector $\mathbf{v}$ is the total number (denoted by N) of dense trajectories extracted over the entire training video set. In other words, each v_j ($j \in \{1, \cdots, N\}$) can be regarded as a local copy of the associated entry in $\mathbf{w}$ according to the visual word that the j-th (of the entire training trajectory set) trajectory feature is indexed to. $\mathbf{v}$ can be also decomposed into $\{\mathbf{v}_{d,p}\}$. Each $\mathbf{v}_{d,p}$ is associated with the trajectory features in the p-the motion part (trajectory group) of the d-th video, in the similar way that $\mathbf{w}_{d,p}$ is defined. Namely, each $\mathbf{v}_{d,p}$ can also be regarded as a local copy of its corresponding $\mathbf{w}_{d,p}$. The dimensionality of $\mathbf{v}_{d,p}$ will be identical to the size (number of trajectories) of the motion part (d, p), with one dimension per word token.

Using the auxiliary variable $\mathbf{v}$, sparse group (motion part) selection could be enforced by the following regularizer:

$$\Omega_{gl}(\mathbf{v}) = \sum_{d=1}^{D} \sum_{p=1}^{P_d} \|\mathbf{v}_{d,p}\|_2, \tag{6}$$

and since each $\mathbf{v}_{d,p}$ is just a local copy of $\mathbf{w}_{d,p}$ and its elements are not shared by other $\mathbf{v}_{d',p'}$ ($d \neq d', p \neq p'$), the original overlapping lasso problem is converted into a non-overlapping one. What remains is to enforce each v_j ($j \in \{1, \cdots, N\}$) to *agree* with its corresponding entry in the global model coefficient vector $\mathbf{w}$. To achieve this, we introduce an assignment matrix $\mathbf{M}$. $\mathbf{M}$ is a $N \times V$ binary matrix, such that $M_{i,j} = 1$ if the local trajectory feature i is indexed to visual word j and 0 otherwise. Therefore, each row of the matrix $\mathbf{M}$ sums up to one. Using the assignment matrix $\mathbf{M}$, the agreement between the global model coefficient vector $\mathbf{w}$ and the local copy $\mathbf{v}$ can be enforced by the penalty $\|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2$. Based on the above definitions, the integrated learning objective for motion part selection is formulated as:

$$\min_{\mathbf{w},b,\mathbf{v}} \mathcal{L}(\mathbf{w}, b) + \lambda_l \|\mathbf{w}\|_1 + \lambda_{gl} \sum_{d=1}^{D} \sum_{p=1}^{P_d} \|\mathbf{v}_{d,p}\|_2 + \beta \|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2, \tag{7}$$

where λ_l, λ_{gl} and β are the weighting factors for the sparsity, group sparsity and global-local agreement terms, respectively. From the objective function (7), we note that to optimize it with respect to either $\mathbf{w}$, b, or $\mathbf{v}$ is quite straightforward if the other variable is fixed. We therefore introduce an alternative optimization scheme as follows.

Update $\mathbf{v}$. With fixed value of $(\mathbf{w}, b)$, minimizing the cost function (7) with respect to $\mathbf{v}$ reduces to:

$$\min_{\mathbf{v}} \lambda_{gl} \sum_{d=1}^{D} \sum_{p=1}^{P_d} \|\mathbf{v}_{d,p}\|_2 + \beta \|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2, \tag{8}$$

which is a standard unconstrained quadratic programming problem with ℓ_2 regularizer. The problem is also convex. Since $\{\mathbf{v}_{d,p}\}$ are no longer overlapping, we separately solve for each $\mathbf{v}_{d,p}$. We denote by $\mathbf{M}_{d,p}$ the sub-matrix of $\mathbf{M}$ corresponding to the motion part (d,p). By applying the proximal projection operator used in non-overlapping group lasso, the optimal value $\mathbf{v}_{d,p}^{opt}$ is easily derived as:

$$\begin{aligned}\mathbf{v}_{d,p}^{opt} &= \text{prox}_{\Omega_{gl},\frac{\lambda_{gl}}{\beta}}(\mathbf{M}_{d,p}\mathbf{w}) \\ &= \begin{cases} \mathbf{0}, & \|\mathbf{M}_{d,p}\mathbf{w}\|_2 < \frac{\lambda_{gl}}{\beta}; \\ \frac{\|\mathbf{M}_{d,p}\mathbf{w}\|_2 - \frac{\lambda_{gl}}{\beta}}{\|\mathbf{M}_{d,p}\mathbf{w}\|_2}\mathbf{M}_{d,p}\mathbf{w}, & \text{else.} \end{cases}\end{aligned} \tag{9}$$

Update $\mathbf{w}$ and b. We denote the entry of $\mathbf{v}$ which corresponds to n-th instance in the set of trajectories of the training videos by v_n, *i.e.*, $n \in \{1, \cdots, N\}$. We denote the occurrence frequency (count) of visual word i in the training videos by N_i. Let $v_{i,n}$ denote the entry of $\mathbf{v}$ corresponding to the n-th instance (token) of visual word i for $n \in \{1, \cdots, N_i\}$. We can rewrite the optimization problem with respect to $\mathbf{w}$ by fixing $\mathbf{v}$ as:

$$\min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b) + \lambda_l \|\mathbf{w}\|_1 + \beta \|\mathbf{v} - \mathbf{M}\mathbf{w}\|_2^2. \tag{10}$$

After simple mathematic manipulation, Equation (10) is equivalent to:

$$\min_{\mathbf{w},b} \mathcal{L}(\mathbf{w}, b) + \lambda_l \|\mathbf{w}\|_1 + \beta \|\mathbf{w} - \mathbf{u}\|_2^2 \tag{11}$$

where $\mathbf{u}$ is a vector with the same length of $\mathbf{w}$ and each $u_i = \frac{1}{N_i}\sum_{n=1}^{N_i} v_{i,n}$, where $i \in \{1, \cdots, V\}$. Intuitively, each entry of the global model coefficient vector $\mathbf{w}$ is regularized towards the mean of its corresponding local copy variables in $\mathbf{v}$. We note that an accelerated proximal gradient method (APG) [22] can be derived to solve $\mathbf{w}$, since $\mathcal{L}(\mathbf{w}, b) + \beta\|\mathbf{w} - \mathbf{u}\|_2^2$ is convex and continuous. The proximal operator for the ℓ_1 regularizer $\lambda_l\|\mathbf{w}\|_1$ (lasso) is given by the soft-thresholding operator:

$$[\text{prox}_{\ell_1,\lambda_l}(\mathbf{w})]_j = \begin{cases} w_j - \lambda_l, & w_j > \lambda_l; \\ 0, & |w_j| \leq \lambda_l; \\ w_j + \lambda_l, & w_j < -\lambda_l. \end{cases} \tag{12}$$

Note that Equation (11) is convex and continues with respect to b, we can therefore minimize it with any gradient descent based method. In this work, we also use AGP to solve for the optimal value of b.

We iteratively update $\mathbf{v}$, $\mathbf{w}$, and b with the other parameters fixed to their old values. In practice, our algorithm usually converges within 200 iterations during our experiments. We set the convergence condition as relative changes in the ℓ_2 norm of the parameter vector $\mathbf{w}$ is small than $\epsilon = 10^{-5}$, or the maximum number of iterations (we set 500) is reached. To choose the best parameter values of λ_l, λ_{gl} and β from the candidate set $\{10^{-5}, 10^{-4}, \cdots, 10^5\}$, we perform three-fold cross-validation on the training set. We note that in [32], alternating directions method of multipliers (ADMM) is employed for optimizing the group lasso problem; however, we found in experiment that ADMM is sometimes unstable. In practice, we found our simple optimization scheme is more stable and efficient.

3.3. Action Representation: Discriminativeness Weighted Fisher Vector

We apply the above learning framework to *softly* select the discriminative motion parts for each action category, *i.e.*, we treat this category as the positive class and the other training video samples which do not belong to it as negative training samples. Therefore, for each action class, we obtain a set of motion parts paired with learned discriminative weights (scores), *i.e.*, $\|\mathbf{v}_{d,p}\|_2$. We perform normalization for these motion part weights among difference classes.

To represent each motion part (to form a mid-level feature vector for each trajectory group), we use the improved Fisher vector (IFV) encoding [16] of dense trajectories descriptors including MBH, HOG, HOF and trajectory shape for the trajectories within each motion part. The number of Gaussian mixture models is set as 256. We also follow [16] to compute square-rooting and normalization for the improved Fisher vector. To reduce feature dimension, we further apply PCA and keep 95% energy. Different feature channels are concatenated. A motion part feature and its corresponding weight is then denoted as $(\mathbf{p}, s)$. We denote by s the discriminativeness score.

Given a testing video sample, since we do not have the class label information (which is to be predicted), it is not possible to learn the motion part weights (discriminativeness) using our proposed method. We therefore propose the following part matching and weight propagation scheme for computing the action representation for a testing video.

First, we construct a database of training motion parts, which consist of 1) the training motion parts with top 10% discriminative weights (denoted as discriminative set); and 2) randomly sampled training motion parts with lower scores (denoted as background set). We set the size of the background set two times of that of the discriminative set.

For each testing video sample, we first generate motion parts using the dense trajectory grouping algorithm introduced in Section 3. Then for each testing motion part, we match its mid-level representation vector (IFV) to those of the motion parts from the above mentioned motion part database and search for its best K (we set $K = 7$ in this work) matches. Since each training motion part is associated with a discriminative weight s, we can then transfer the mean value of the weights associated with the matched training parts towards the testing motion part. After this discriminative weight transfer, every motion part in the testing

video is endowed with a discriminative weight. We assume that all the dense trajectory features within each motion part inherit the motion part's weight. We can then form a discriminativeness weighted Fisher vector representation for the testing video as follows.

Assume the set of trajectory features paired with discriminative scores of the testing video is denoted as $X = \{\mathbf{x}_t, s_t\}_{t=1:T}$, where T denotes the number of local features of the testing video sample. We then recall that the gradient with respect to the Gaussian mean $\mathbf{u}_i$ of an improved Fisher vector [16] can be formulated as (we take the example of mean, and the gradient of variance calculation is similar):

$$\mathcal{G}^X_{\mathbf{u}_i} = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma_t(i) \Big(\frac{\mathbf{x}_t - \mathbf{u}_i}{\sigma_i} \Big), \tag{13}$$

where $\mathbf{u}_i$ and σ_i is the mean and variance of the i-th Gaussian component, respectively. ω_i is the weight of the Gaussian component $\mathbf{u}_i$. $\gamma_t(i)$ is the soft assignment of the local feature descriptor $\mathbf{x}_t$ to Gaussian component $\mathbf{u}_i$. Then, our discriminativeness weighted Fisher vector can be written as:

$$\mathcal{G}^X_{\mathbf{u}_i} = \frac{1}{T\sqrt{\omega_i}} \sum_{t=1}^{T} \gamma_t(i) s_t \Big(\frac{\mathbf{x}_t - \mathbf{u}_i}{\sigma_i} \Big). \tag{14}$$

Again, we use the square-rooting and normalization for the resulting Fisher vector representation. Note that important motion part (as well as the local dense trajectories within it) receives higher weight s_t, therefore its importance/significance in the resulting Fisher vector will be high.

What remains is how to match the testing motion part representation vector to database motion part vectors. Because of the high-dimensionality, direct matching using Euclidean distance cannot achieve good performance. Inspired by the recent success of exemplar-SVM [11], for each database motion part, we learn a detector, by taking itself as the only positive training sample and randomly sampling some negative samples from the database. Therefore, each database motion part is endowed with a part detector. For each testing part, we can therefore search its K best matches by simply applying all the database part detectors and ranking the response scores. The pipeline of our method is shown in Figure 2. For action classification, we use a linear SVM classifier based on LibSVM [1]. The penalty parameter is set as $C = 100$.

4. Experiment

4.1. Experimental Settings

We perform action recognition on several benchmark datasets to evaluate the effectiveness of our method as well as to study its algorithmic behavior. We compare our method with the state-of-the-art action recognition methods in terms of recognition accuracy. The datasets on which we test the algorithms include 1) Hollywood2 movie action dataset [12]; 2) HMDB51 YouTube action dataset [9]; and 3) Olympic Sports dataset [14]. The details about various dataset are briefly summarized as follows.

1. The Hollywood2 dataset [12]: It consists of 12 action classes such as *answering the phone*, *driving car*, *eating*, etc., with $1,707$ video samples in total. We follow the same experimental settings as in [23] [24]. The mean average precision (mAP) over all classes is reported.

2. The HMDB51 dataset [9]: Collected from YouTube, it contains 51 action categories and $6,766$ video sequences. The action categories include simple facial actions, general body movements and human interactions. We follow the experimental setting used in [24]. Average accuracy over the three train-test splits is reported.

3. The Olympic Sports dataset [14]: It consists of sports action videos collected from YouTube, which contains 16 sports actions (such as *high-jump*, *pole-vault*, *basketball lay-up*, *discus*, etc.) with a total of 783 video samples. Following [25], we also use 649 samples for training and 134 samples for testing as recommended by the authors. Mean average precision (mAP) over all action classes is reported.

4.2. Qualitative Results

First, we visualize the learned discriminative motion parts (trajectory groups). In Figure 3, we show several examples of the top 1% ranked (based on the learned discriminative scores/weights) motion parts from various action categories from the HMDB51 dataset. Different motion parts are color-coded. From Figure 3, we observe that: our algorithm can automatically discover important/discriminative/representative motion parts for various action categories. For example, our algorithm discovers the important motions around the lower/upper arms and shoulders for the golf swing motion; it discovers the *cup to mouth* movement which is the most representative motion for the action *drink*; and it also discovers the critical movement of hand/lower arm for the action category *punch*.

4.3. Quantitative Results

We also compare various state-of-the-art action recognition algorithms on these action video benchmarks. For our method, we use the recently proposed improved dense motion trajectories [25] for feature extraction (We use the implementation provided by the authors [25] and follow the same parameter settings). Following [25], global background motion compensation and human detection is also

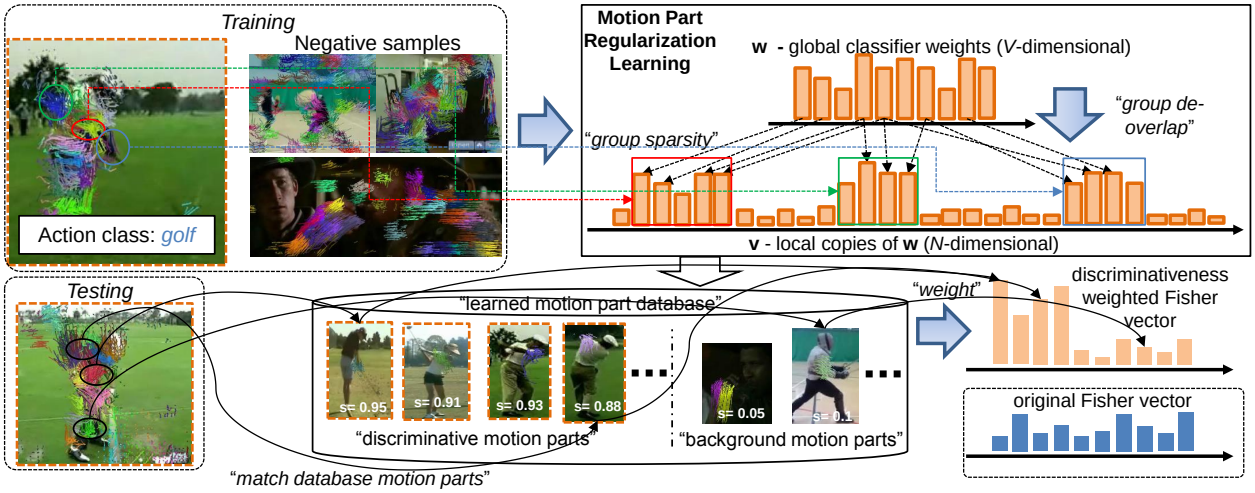

Figure 2. Illustration of the pipeline of our method, including motion part regularization learning, discriminative motion part database generation and discriminativeness weighted Fisher vector generation for a testing video.

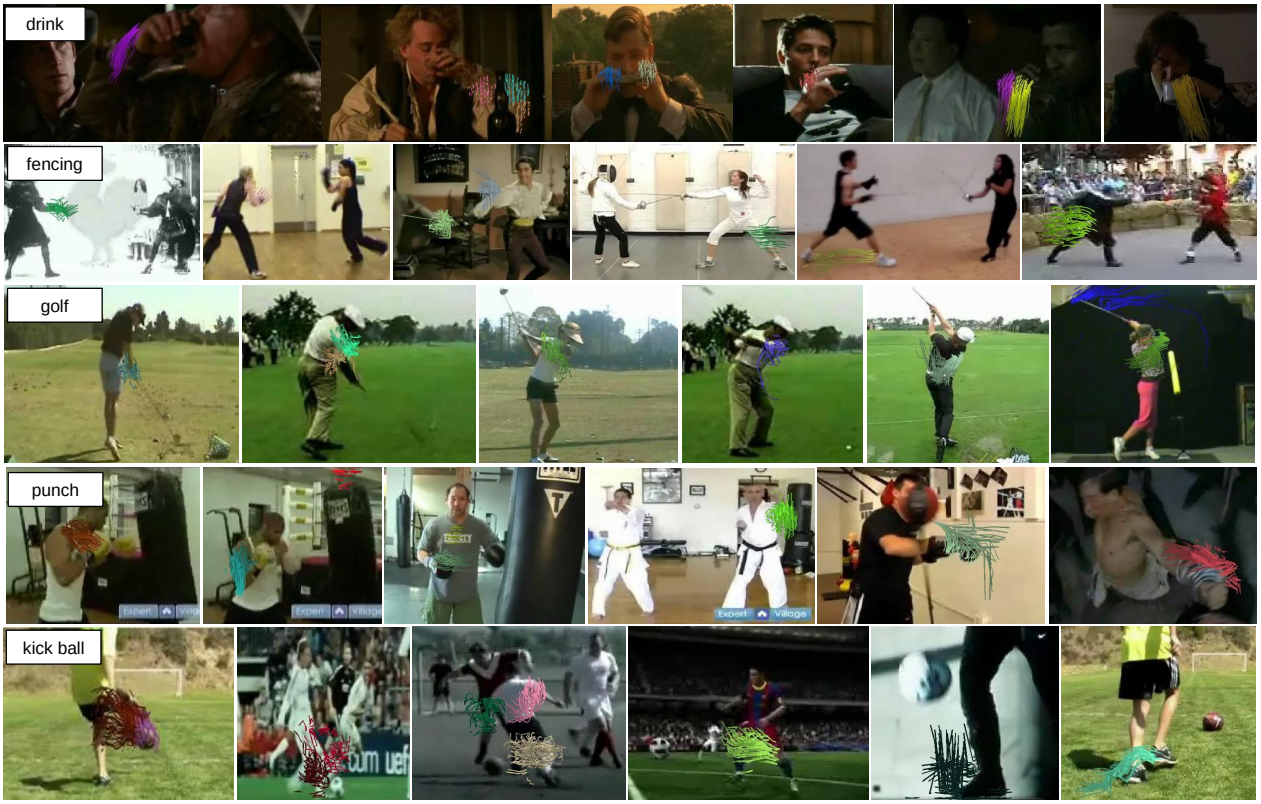

Figure 3. Examples of 1% ranked (based on the learned discriminative scores/weights) motion parts (trajectory groups) discovered by our motion part regularization learning framework. Each row corresponds to one class of action with the action label given at the top left of the first image.

applied, for the purpose of flow warping (stabilization) and outlier trajectory removal, respectively. For our algorithm, we retain the top 10% ranked discriminative motion parts from the training video to form the discriminative part database, which is used to compute discriminativeness weighted Fisher vector representation for each video sample.

We compare our method with the follows algorithms: 1) Harris3D (STIP) [10] + HOG/HOF; 2) improved dense trajectories [25] with both BOW and IFV feature encodings;

Table 1. Comparison of action recognition performance on the Hollywood2, HMDB51, and Olympic Sports datasets.

Method	Hollywood2 [12]	HMDB51 [9]	Olympic Sports [14]
Harris3D [10] + HOG/HOF	–	20.2% (from [9])	–
Improved Trajectory + BOW (IFV) [25]	62.2% (64.3%)	52.1%(57.2%)	83.3% (91.1%)
Jiang et al. [6]	59.5%	40.7%	80.6%
Jain et al. [5]	62.5%	52.1%	83.2%
Motion Atoms/Phrases [27] (+low level)	–	–	79.5% (84.9%)
LHM + Dense Trajectory [28]	59.9%	–	83.2%
Motion Actons [35]	61.4%	54.0%	–
Stacked Fisher Vector [15]	–	**66.8%**	–
Motion Part Regularization (ours)	**66.7%**	65.5%	**92.3%**

3) the method proposed by Jain et al. [5] which decomposes visual motion to stabilize dense trajectories; 4) the model proposed by Jiang et al. [6] which explores the the relationship among dense trajectory clusters; 5) the latent hierarchical model (LHM) proposed by Wang et al. [28]; 6) the motion atoms and phrases representation [27]; 7) the motion actons [35]; 8) the stacked Fisher vector encodings [15]. Table 1 shows the comparison results. We report the results of comparing algorithms from their original papers if applicable. From Table 1, we make the following observations. First, our proposed method either outperform the state-of-the art methods or achieves the state-of-the-art performances on the tested databases. The stacked Fisher vector [15] performs slightly better than our method on HMDB51 dataset; however, this result is achieved by combining two IFV vectors: one pooled from low level trajectory features and another pooled from mid-level features. In contrast, our method only use a single discriminativeness weighted IFV vector. This shows that finding the most discriminative and representative motion part is helpful for high performance action recognition. We also note that methods based on IFV always give high recognition performed, *e.g.*, [25, 15] and our method, which demonstrates that IFV is a high-performance feature coding scheme for action representation. Finally, our mid-level feature selection scheme is better than other mid-level mining methods, *e.g.*, [35] and [27].

To study the effect of the size of the discriminative motion part database (number of selected discriminative parts) on the final classification performance, we select the top 1%, 5%, 7%, 10%, and 20% discriminative motion parts from the training videos to compose the database, and compute the corresponding video level representationes (discriminativeness weighted Fisher vector) based on them. We show the corresponding classification performances (mAP) for the Hollywood2 and HMDB51 databases in Figure 4. We see that in general the classification performance increases with respect to the increase of the discriminative part database size. However, the accuracy improvement becomes negligible when more than top 10% discriminative parts are selected to compose the database. Therefore, in our experiment, we choose the top 10% discriminative motion parts to compose the discriminative motion part database if not otherwise specified.

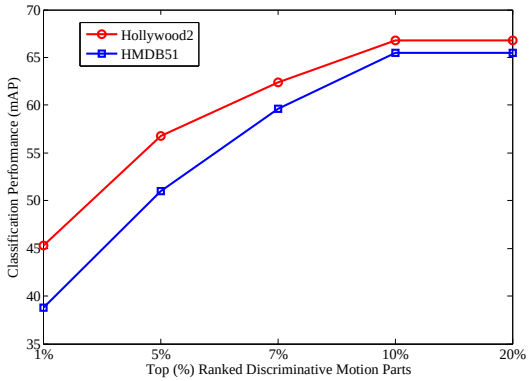

Figure 4. Classification performances w.r.t the number of top (%) ranked discriminative motion parts used for motion part database construction and video representation calculation.

5. Conclusion

In this paper, we propose a motion part regularization framework for discriminative mid-level motion representation (*i.e.*, trajectory group) selection, based on overlapping group lasso. We also propose a *de-overlap* scheme along with an efficient alternative optimization algorithm to solve the motion part selection problem. The selected discriminative motion parts are then utilized to form discriminativeness weighted Fisher vector action representation. Our experiments on several action video benchmarks demonstrate that our method can select discriminative motion part for action representation and it improves the state-of-the-art on several benchmarks.

Acknowledgment

This study is supported by the research grant for the human-centered cyber-physical systems (HCCS) at the Ad-

vanced Digital Sciences Center from Singapore's Agency for Science, Technology and Research (A*STAR).

References

[1] C.-C. Chang and C.-J. Lin. LIBSVM: A library for support vector machines. *ACM TIST*, 2(27):1–27, 2011. 6

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005. 1

[3] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, pages 1491–1498, 2009. 2

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE T-PAMI*, 32(9):1627–1645, 2010. 1, 2

[5] M. Jain, H. Jegou, and P. Bouthemy. Better exploiting motion for better action recognition. In *CVPR*, pages 2555–2562, 2013. 8

[6] Y.-G. Jiang, Q. Dai, X. Xue, W. Liu, and C.-W. Ngo. Trajectory-based modeling of human actions with motion reference points. In *ECCV*, pages 425–438, 2012. 8

[7] A. Klaser, M. Marszalek, and C. Schmid. A spatio-temporal descriptor based on 3d gradients. In *BMVC*, 2008. 1

[8] Y. Kong, Y. Jia, and Y. Fu. Interactive phrases: Semantic descriptionsfor human interaction recognition. *IEEE T-PAMI*, 36(9):1775–1788, 2014. 2

[9] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. HMDB: a large video database for human motion recognition. In *ICCV*, 2011. 2, 6, 8

[10] I. Laptev and T. Lindeberg. Space-time interest points. In *ICCV*, 2003. 1, 2, 7, 8

[11] T. Malisiewicz, A. Gupta, and A. A. Efros. Ensemble of exemplar-svms for object detection and beyond. In *ICCV*, 2011. 6

[12] M. Marszalek, I. Laptev, and C. Schmid. Actions in context. In *CVPR*, pages 2929–2936, 2009. 2, 6, 8

[13] B. Ni, P. Moulin, and S. Yan. Pose adaptive motion feature pooling for human action analysis. *IJCV*, 2014. 3

[14] J. C. Niebles, C. W. Chen, and L. Fei-fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, pages 392–405, 2010. 2, 6, 8

[15] X. Peng, C. Zou, Y. Qiao, and Q. Peng. Action recognition with stacked fisher vectors. In *ECCV*, 2014. 3, 8

[16] F. Perronnin, J. Sánchez, and T. Mensink. Improving the Fisher kernel for large-scale image classification. In *ECCV*, pages 143–156, 2010. 5, 6

[17] M. Raptis, I. Kokkinos, and S. Soatto. Discovering discriminative action parts from mid-level video representations. In *CVPR*, 2012. 2, 3

[18] M. S. Ryoo and J. Aggarwal. Spatio-temporal relationship match: Video structure comparison for recognition of complex human activities. In *ICCV*, pages 1593–1600, 2009. 1

[19] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, pages 536–548, 2010. 2

[20] Q. Shi, L. Wang, L. Cheng, and A. Smola. Discriminative human action segmentation and recognition using semi-markov models. *IJCV*, 93(1):22–32, 2011. 2

[21] K. Tang, L. Fei-fei, and D. Koller. Learning latent temporal structure for complex event detection. In *CVPR*, 2012. 2

[22] P. Tseng. On accelerated proximal gradient methods for convex-concave optimization. *SIAM Journal on Optimization*, 2008. 5

[23] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin. Action recognition by dense trajectories. In *CVPR*, pages 3169–3176, 2011. 1, 2, 3, 6

[24] H. Wang, A. Kläser, C. Schmid, and C.-L. Liu. Dense trajectories and motion boundary descriptors for action recognition. *IJCV*, 103(1):60–79, 2013. 6

[25] H. Wang and C. Schmid. Action recognition with improved trajectories. In *ICCV*, 2013. 1, 2, 3, 6, 7, 8

[26] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, pages 1290–1297, 2012. 3

[27] L. Wang, Y. Qiao, and X. Tang. Mining motion atoms and phrases for complex action recognition. In *ICCV*, pages 2680–2687, 2013. 3, 8

[28] L. Wang, Y. Qiao, and X. Tang. Latent hierarchical model of temporal structure for complex activity classification. *IEEE T-IP*, 23(2):810–822, 2014. 8

[29] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song. Unified video annotation via multi-graph learning. *IEEE T-CSVT*, 19(5):733–746, 2009. 2

[30] M. Wang, X.-S. Hua, J. Tang, and R. Hong. Beyond distance measurement: Constructing neighborhood similarity for video annotation. *IEEE T-MM*, 11(3):465–476, 2009. 2

[31] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *IEEE T-PAMI*, 33(7):1310–1323, 2011. 1, 2

[32] D. Yogatama and N. A. Smith. Making the most of bag of words: Sentence regularization with alternating direction method of multipliers. In *ICML*, 2014. 2, 4, 5

[33] J. Yuan, Z. Liu, and Y. Wu. Discriminative video pattern search for efficient action detection. *IEEE T-PAMI*, 33(9):1728–1743, 2011. 2

[34] M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 68(1):49–67, 2006. 4

[35] J. Zhu, B. Wang, X. Yang, W. Zhang, and Z. Tu. Action recognition with actons. In *ICCV*, pages 3559–3566, 2013. 8