# Motionlets: Mid-Level 3D Parts for Human Motion Recognition *

LiMin Wang[1,2], Yu Qiao[2] †, and Xiaoou Tang[1,2]
[1]Department of Information Engineering, The Chinese University of Hong Kong
[2]Shenzhen key lab of Comp. Vis. & Pat. Rec., Shenzhen Institutes of Advanced Technology, CAS, China
07wanglimin@gmail.com, yu.qiao@siat.ac.cn, xtang@ie.cuhk.edu.hk

## Abstract

*This paper proposes* motionlet*, a mid-level and spatiotemporal part, for human motion recognition. Motionlet can be seen as a tight cluster in motion and appearance space, corresponding to the moving process of different body parts. We postulate three key properties of motionlet for action recognition: high motion saliency, multiple scale representation, and representative-discriminative ability. Towards this goal, we develop a data-driven approach to learn motionlets from training videos. First, we extract 3D regions with high motion saliency. Then we cluster these regions and preserve the centers as candidate templates for motionlet. Finally, we examine the representative and discriminative power of the candidates, and introduce a greedy method to select effective candidates. With motionlets, we present a mid-level representation for video, called* motionlet activation vector*. We conduct experiments on three datasets, KTH, HMDB51, and UCF50. The results show that the proposed methods significantly outperform state-of-the-art methods.*

## 1. Introduction

Due to the popularization of surveillance cameras and personal video devices, video based human motion analysis and recognition have become a highly active area in computer vision [2]. Human action recognition is difficult for many reasons, such as high-dimension of video data, intra-class variability caused by scale, viewpoint and illumination changes, low resolution and video quality. Like many computer vision problems, an effective *visual representation* of video data is vital to deal with these problems.

*This work is partly supported by National Natural Science Foundation of China (61002042), Shenzhen Basic Research Program (JC201005270350A, JCYJ20120903092050890, JCYJ20120617114614438), 100 Talents Programme of Chinese Academy of Sciences, and Guangdong Innovative Research Team Program (No.201001D0104648280).
†Corresponding Author.

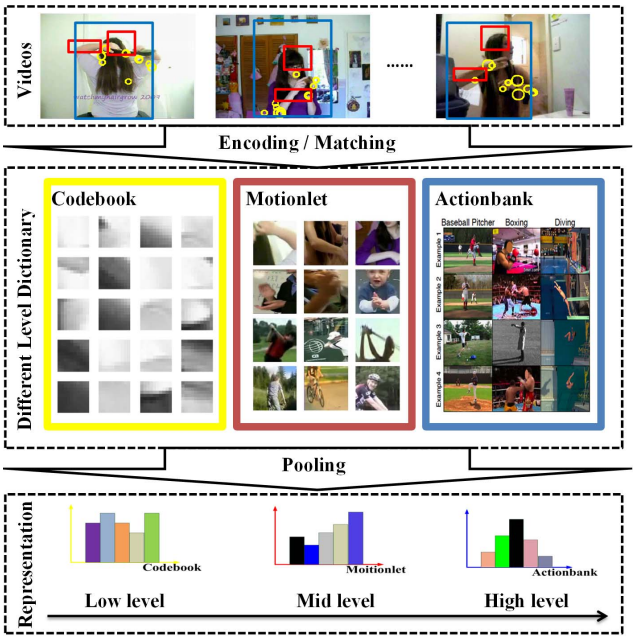


Figure 1. Three types of video representation: Low-level representation is based on local features around small regions (indicated by yellow circles) and maps video into " codebook space"; High-level representation is based on a global template covering the whole action (indicated by blue boxes) and encode the video into " action space"; Our motionlet is a mid-level 3D part (indicated by red boxes) and share advantages of both low and high level representation.

Currently, the most popular methods of video representation are based on local spatio-temporal features. These methods use detectors such as Harris3D [20], Cuboids [9] to locate spatio-temporal interested points, and extract descriptors like HOG/HOF [21], HOG3D [17]. Then the extracted descriptors are quantified with a pre-learned codebook, and input videos are modeled with Bag of Visual Words (BoVW) [31] (Figure 1). These local descriptors share some desired properties for video representation, such as *local* and *repeatable*. They usually describe motion and appearance information of a local cuboid around interest

point. Thus it tends to be easy to find repeatable patterns among different videos. These properties make local descriptors robust to intra-class variability and deformation to a certain degree. However, they only capture low-level information, and may lack discriminative power for high-level motion recognition. Several recent works attempt to solve this problem by introducing high-level features or models, such as Motion Energy and History Image [3], Silhouette [7], Space-time Shape [14]. Among them, Action Bank [27] applies a large set of action detectors on input video, and use the responses of these detectors as a semantically rich representation (Figure 1). These high-level detectors are made up of global templates of actions, and have desired properties such as *global* and *discriminative*. But the global nature makes them sensitive to intra-class variation and deformation.

To balance between low-level and high-level representations, we propose a mid-level 3D (spatio-temporal) part, called *motionlet*. The concept of " Part " originates from the area of object recognition, and has been widely explored in object detection and recognition recently. Different from 2D image, video is represented by a 3D volume data with an additional dimension of time, which exhibits properties different from 2D spatial dimensions. In addition to appearance, motion is an important visual cues for action recognition. Moreover, it is more ambiguous and difficult to define parts for human motion than for objects. In this paper, we define Motionlet as a spatio-temporal part with stable features in both appearance space and motion space. It corresponds to the moving process of parts, objects, visual phrases (See Figure 1).

We expect motionlets to have three desired properties: 1) *high-motion saliency*, which means it is able to capture the part with strong motion cues in videos; 2) *multiple scale representation*, which means it is a balance between with local features and global template and can capture motions at different scales; 3) *representative* and *discriminative ability*, "representative" implies it should occur frequently enough in action video, and "discriminative" indicates it can provide rich information for classifying motions.

To achieve the above goals , we propose a learning based approach to extract motionlets from training videos. Specifically, we first estimate motion saliency using spatiotemporal orientation energies [1], and extract 3D regions with high motion saliency. Then we tightly cluster these 3D regions into candidate motionlets, and keep the medians for each cluster as the templates. Finally, we examine the representative and discriminative power of these candidates, and introduce a greedy search algorithm to select effective candidates as motionlets. We represent a video by *motionlet activation vector*, which measures the strength of each motionlet occurring in the video. We conduct experiments on human motion recognition on three public datasets: KTH [28], UCF50 [26], and HMDB51 [19]. The proposed methods achieve significant improvements compared with state-of-the-art methods. We obtain accuracy rates of 78.4% on UFC50 and 42.1% on HMDB51, which are the best results reported on these datasets so far.

## 2. Related Work

The concept of "part" has been widely and successfully used in image based object detection and recognition. In [11], Felzenszwalb *et al.* propose Pictorial Structure Model and use a tree to model the relationship among different body parts. In a more recent work [10], Felzenszwalb *et al.* propose Deformable Part Model (DPM) for object detection and achieve success on identifying very difficult objects. DPM uses root detector to find a match of the whole object, and then uses part detectors to fine-tune the result. Perhaps the most similar work to ours is Poselet proposed by Bourdev *et al.* [4]. They construct Poselet based on annotations of human pose in 2D image. Motionlet differs from Poselet in two ways: 1) Motionlet is a 3D part constructed from video and designed for human motion recognition; 2) we construct motionlet in an unsupervised way without using human annotations of pose.

Several recent action recognition methods also make use of the concept of "part", either explicitly or implicitly. In [23], Niebles *et al.* propose to decompose the whole video into several segments, which can be regarded as temporal "parts". In [5], Brenderl *et al.* firstly over-segment the whole video into tubes corresponding to action "part" and adopt spatiotemporal graphs to learn the relationship among the parts. In [25], Raptis *et al.* group the trajectories into clusters, each of which can be seen as an action part. Then they use graphical model to incorporate motion/appearance information of each part and pairwise constraints between parts. All these part related methods rely on complex models and iterative optimization algorithms to learn model parameters. Different from these methods, our motionlets are motion templates and provide mid-level representation of video. Moreover, motionlets do not rely on specific inference algorithms in recognition step, which makes it easy to be combined with other methods.

## 3. Low-Level Features

There are many low level features designed for video data, such as STIP [20] and Cuboids [9]. Most of these local features are based on interest points, and are suitable for BoVW based video representation. To construct motionlets, however, we need features for describing and matching templates. In this paper, we use spatiotemporal orientation energy (SOE) [1] as low level features. SOEs have been used for action recognition in [8, 27]. They compute SOE for each pixel, thus making feature sensitive to small shift

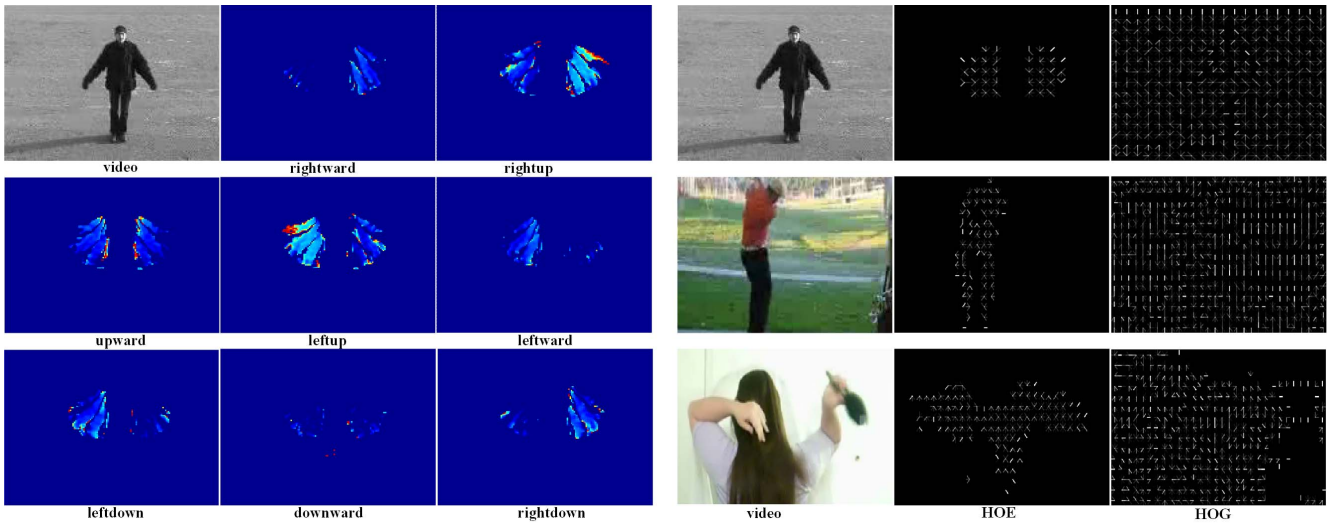


Figure 2. Low-level Features, left: the motion saliency of eight orientations; right: dense HOE and HOG features of three video clips from KTH, HMDB51 and UCF 50.

and adding the computational cost. Instead, we resort to a dense sampling strategy and compute SOE histogram as features. This approach not only fits motionlet representation of video very well, but also reduce time cost in template matching.

**Spatiotemporal Orientation Energy.** Video can be seen as a 3D volume data. We use 3D steerable filter to estimate local spatiotemporal orientation energy to represent the strength of motion along 3D spatiotemporal directions. Specially, we use third derivatives of 3D Gaussian as local filters, $G^3_{\hat{\theta}}(\mathbf{x})$, where $\mathbf{x} = (x, y, t)$ represents a position in spatiotemporal space, and unit vector $\hat{\theta}$ denotes a 3D direction. We can estimate the spatiotemporal orientation energy at each pixel as follows,

$$E_{\hat{\theta}}(\mathbf{x}) = \sum_{\mathbf{x}' \in \Omega(\mathbf{x})} (G^3_{\hat{\theta}} * V)^2, \tag{1}$$

where $\Omega(\mathbf{x})$ represents a local region around $\mathbf{x}$, $V \equiv V(\mathbf{x})$ is the input video, and $(*)$ denotes convolution. Note the separable property of steerable filters allows us to estimate SOE efficiently without conducting convolution for all directions [12]. In order to remove the influence of spatial orientation, the energy is usually processed by a "marginalization" step [8]. Specifically, energy along a frequency domain plane with normal $\hat{\mathbf{n}}$ is calculated by summing a set of measurements $E_{\hat{\theta}_i(\hat{\mathbf{n}})}$,

$$\widetilde{E}_{\hat{\mathbf{n}}}(\mathbf{x}) = \sum_{i=0}^{N} E_{\hat{\theta}_i(\hat{\mathbf{n}})}(\mathbf{x}), \tag{2}$$

where $N$ is the order of Gaussian derivatives, $\hat{\theta}_i$ is one of $N + 1 = 4$ directions needed to span orientation plane. Formally $\theta_i(\hat{\mathbf{n}})$ is given by,

$$\hat{\theta}_i(\hat{\mathbf{n}}) = \cos\left(\frac{\pi i}{N+1}\right)\hat{\theta}_a(\hat{\mathbf{n}}) + \sin\left(\frac{\pi i}{N+1}\right)\hat{\theta}_b(\hat{\mathbf{n}}), \tag{3}$$

with $\hat{\theta}_a(\hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \hat{\mathbf{e}}_x / \|\hat{\mathbf{n}} \times \hat{\mathbf{e}}_x\|$, $\hat{\theta}_b(\hat{\mathbf{n}}) = \hat{\mathbf{n}} \times \hat{\theta}_a(\hat{\mathbf{n}})$, and $\hat{\mathbf{e}}_x$ is the unit vector along the $\omega_x$ axis. When spacetime orientation is defined by image velocity $(u, v)^\top$, the normal vector is given by $\hat{\mathbf{n}} = (u, v, 1)^\top / \|(u, v, 1)^\top\|$.

In our implementation of motionlet, we use nine spatiotemporal energies with different image velocities $(u, v)^\top$ as shown in Table 1. In addition, we define another energy called lack of structure $\widetilde{E}_o$ which is computed as a function of the nine energies and has peaks when none of the other nine energies has strong response. This energy is introduced to avoid instabilities at points where overall energy is small. As observed in our experiments and in [27], it is better to use the energy of $\widetilde{E}_s$ and $\widetilde{E}_o$ which separate the pure orientation energies from the background and noise influence:

$$\overline{E}_i = \max(\widetilde{E}_i - \widetilde{E}_s - \widetilde{E}_o, 0), \quad \forall i \in \text{All} - \{s, o\}. \tag{4}$$

The resulting eight energies can be seen as measures of motion saliency along eight different orientations (Figure 2). Finally, the eight pure energies are normalized to avoid influence of contrast and illumination change.

**Dense features.** We extract dense histogram of spatiotemporal orientation energy (HOE) and histogram of gradient (HOG) for video representation. As shown in Figure 3, we first divide the video into volumes of size $W \times H \times L$. To incorporate the detailed spatiotemporal information, each volume is further divided into $w \times h \times l$ grids. In our experiments, these parameters are set as $W = H = L = 8$ and $w = h = l = 2$. For each grid, we extract two kinds of

| Energy | leftward: $\widetilde{E}_l$ | rightward: $\widetilde{E}_r$ | downward:$\widetilde{E}_d$ |
|---|---|---|---|
| $(u, v)$ | $(-1, 0)$ | $(1, 0)$ | $(0, 1)$ |
| Energy | upward: $\widetilde{E}_u$ | left-up: $\widetilde{E}_{lu}$ | left-down: $\widetilde{E}_{ld}$ |
| $(u, v)$ | $(0, -1)$ | $(-1, -1)$ | $(-1, 1)$ |
| Energy | right-up: $\widetilde{E}_{ru}$ | right-down: $\widetilde{E}_{rd}$ | static: $\widetilde{E}_s$ |
| $(u, v)$ | $(1, -1)$ | $(1, 1)$ | $(0, 0)$ |

Table 1. Image velocities along different orientations.

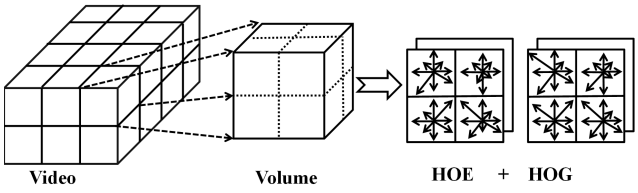


Figure 3. The illustration of dense HOE and HOE extraction.

histogram features to capture motion and appearance information. For motion information, we compute histogram of eight pure energies by Equation (4). For appearance information, we compute histogram of oriented gradient, where the orientation are quantized into eight bins. Thus the total feature dimension for a grid is $8 + 8 = 16$, and the dimension for a volume is $2 \times 2 \times 2 \times 16 = 128$. Both descriptors are normalized with their $L_1$ norm. Figure 2 shows examples of HOE and HOG. By the dense sampling strategies, dense HOE and HOG is more suitable for template matching than interested point based local features such as STIP [20] and Cuboids [9]. Being histogram features, dense HOE and HOG is more compact and efficient than the spatiotemporal orientation energy features used in [27, 8], where they compute a feature vector for each pixel. We define the similarity between two volumes as follows,

$$m(V_i, V_j) = \sum_{k=1}^{128} \sqrt{\mathbf{h}(V_i)^k \mathbf{h}(V_j)^k}\,, \tag{5}$$

where $\mathbf{h}(V_i)$ is a vector of HOE and HOG, and $\mathbf{h}(V_i)^k$ denotes the $k^{th}$ element of $\mathbf{h}(V_i)$. Root function in Equation (5) originates from the definition of Hellinger distance, and proves to be effective for histogram features.

# 4. Motionlet Construction

This section describes how to construct motionlet for video representation. As shown in Figure 4, the whole process consists of three steps, 1) extracting motion salient regions, 2) finding motionlet candidates, and 3) ranking motionlets.

## 4.1. Extraction of Motion Salient Regions

In the first step, we extract 3D video regions with high motion saliency as seeds for constructing motionlets. Like the step for calculating dense features, we divided the video into volumes of size $W \times H \times L$. For each volume $\Omega$, we use the summation of spatiotemporal orientation energies as a measure of motion saliency (See Left of Figure 4),

$$s(\Omega) = \sum_{\mathbf{x} \in \Omega} \sum_{i \in \mathrm{All} - \{s,o\}} \overline{E}_i(\mathbf{x}). \tag{6}$$

Then we use a threshold $\alpha$ to convert motion saliency map into a binary one,

$$\mathcal{B}(\Omega) = \begin{cases} 1 & \text{if } s(\Omega) > \alpha, \\ 0 & \text{otherwis.} \end{cases} \tag{7}$$

Empirically, we set $\alpha$ as 0.9 times of saliency maximum. To obtain region with different sizes, we conduct component analysis based on these binary maps. In spatial dimension, we extract its 8-connected motion regions. For temporal dimension, we fix time duration of each volume. In this way, we can obtain a large pool of 3D regions with different sizes $\{\mathcal{R}_1, \cdots, \mathcal{R}_M\}$. For each 3D region, we extract its dense HOE and HOG $\{h(\mathcal{R}_1), \cdots, h(\mathcal{R}_M))\}$ , where $h(\mathcal{R}_i) \in \mathbb{R}^{w_i \times h_i \times 128}$ ,$w_i$ and $h_i$ represent the spatial sizes of 3D region $\mathcal{R}_i$, and $M$ is the number of 3D regions.

## 4.2. Finding Motionlet Candidates

The 3D regions generated from motion saliency serve as the seeds for constructing motionlet. In this section, we identify representative ones from all 3D regions by using clustering method. Since these 3D regions have different sizes and the associate features $h(\mathcal{R}_i)$ have different dimensions, we cannot compare them directly. Here we design a two-step approach. We first group the 3D regions according to their spatial sizes. This step ensures regions in the same group share similar shapes, and reduces the computational cost in the next step. Then, for each group, we cluster the 3D regions according to motion and appearance information. The key issue is how to measure the similarity between two regions $\mathcal{R}_i$ and $\mathcal{R}_j$. The difficulty comes from that they may have different sizes. We define the similarity as the maximum of the correlation between their two subregions (shown in Middle of Figure 4):

$$\mathrm{Sim}(\mathcal{R}_i, \mathcal{R}_j) = \max_{\mathbf{x}} \left\{ \sum_{\mathbf{u}} m(\mathcal{R}_i(\mathbf{x} + \mathbf{u}), \mathcal{R}_j(\mathbf{u})) \right\}, \tag{8}$$

where $\mathcal{R}_i(\mathbf{x} + \mathbf{u})$ and $\mathcal{R}_j(\mathbf{u})$ denote two volumes started at $\mathbf{x} + \mathbf{u}$ and $\mathbf{u}$ respectively, and $m()$ represents the similarity function defined by Equation 5. $\mathbf{u}$ ranges such that $\mathbf{x} + \mathbf{u} \in \mathrm{Scale}(\mathcal{R}_i)$ and $\mathbf{u} \in \mathrm{Scale}(\mathcal{R}_j)$, and $\mathbf{x}$ ranges over the scale of $\mathcal{R}_i$. The above equation will also be used for matching templates (montionlets) in the recognition step, thus is called *template matching similarity*.

With similarity measures, we use Affinity Propagation [13] to cluster 3D regions. Affinity Propagation is an exem-

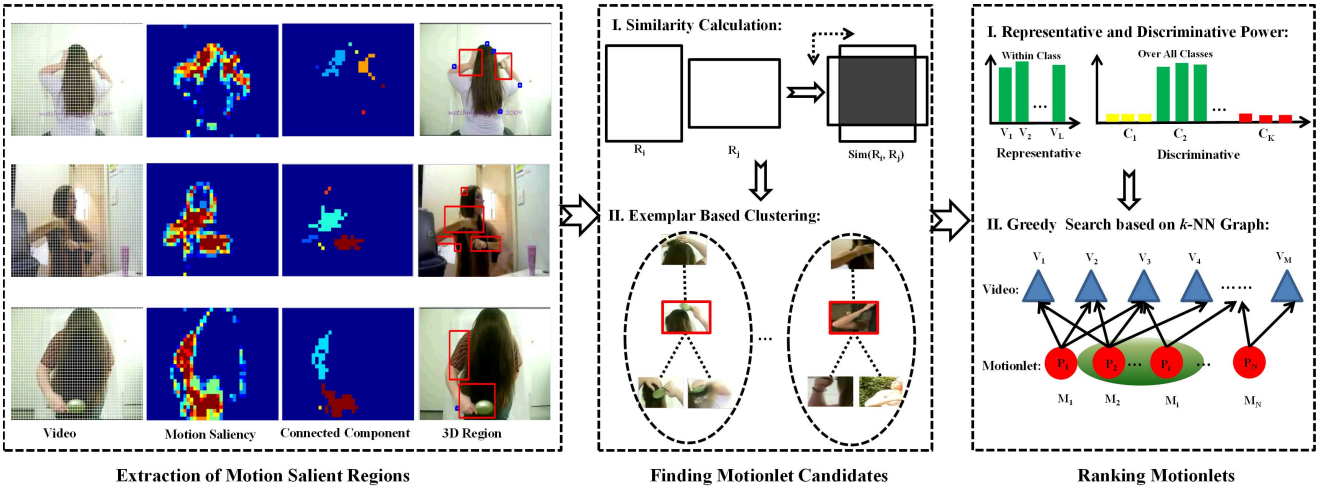


Figure 4. The pipeline of motionlet construction: we first generate a large pool of 3D regions using motion saliency; then, we tightly cluster 3D regions into candidate motionlets; finally, we rank and select motionlets based on their representative and discriminative ability.

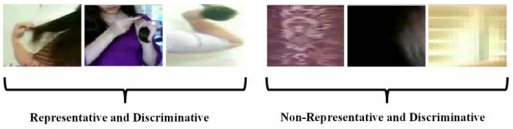


Figure 5. Some examples of representative-discriminative and non representative-discriminative motionlets for brush hair.

plar based cluster algorithm whose input is similarity matrix. It simultaneously considers all data points as potential exemplars and exchanges real-valued messages between data points until obtaining a high-quality set of exemplars. Due to the great variance of video data, the preference parameters of Affinity Propagation are set to be larger than the median to make sure 3D regions within the same cluster very similar. Each cluster corresponds to a candidate motionlet, and some examples are shown in Figure 6. The median of each cluster can be seen as a template of motionlet. The construction of motionlet is conducted for each action category separately. For each action category, we generate about $3,000$ 3D regions and cluster them into $500$ templates.

### 4.3. Ranking Motionlet

The motionlet templates constructed above mainly takes account of the low level features captured by HOE and HOG. As a consequence, it is still uncertain whether these templates are *representative* and *discriminative* for high-level action classification. To be representative, a motionlet should occur frequently and distribute widely in different videos (See Figure 4). To be discriminative, a motionlet should provide information to distinguish one action class from the others (See Figure 4). Some examples are shown in Figure 5.

**Algorithm 1:** Greedy Selection Algorithm of Motionlet.

**Input** : Representative and Discriminative power: $P$. Coverage table: $T$. Selecting number $l$.
**Output**: Selected motionlets: $S$

**Init**: coverage counter $C \leftarrow 0$, selected set $S \leftarrow \emptyset$;
**for** $i \leftarrow 1$ **to** $l$ **do**
1. videoset $\leftarrow$ `FindLeastCoverage(`$C$`)`;
2. motionletset $\leftarrow$ `FindActive(`$T$,videoset`)`;
3. bestmot $\leftarrow$ `FindBest(`$P$,$S$,motionletset`)`;
4. `Update(`$S$,$C$,$T$,bestmot`)`;

**end**

To measure the representative and discriminative ability of motionlets, we use each motionlet as template to scan over all training videos, and analyze their matching response values. Specifically, let $s_i^j$ denote motionlet activation value which is calculated as the max pooling result of matching motionlet $\mathcal{M}_j$ with video $\mathcal{V}_i$, .

$$s_i^j = \max_{\mathbf{x}} \left\{ \sum_{\mathbf{u}} m(\mathcal{V}_i(\mathbf{x}+\mathbf{u}), \mathcal{M}_j(\mathbf{u})) \right\}, \qquad (9)$$

where $\mathcal{V}_i(\mathbf{x}+\mathbf{u}), \mathcal{M}_j(\mathbf{u})$ denotes two volumes and $m(,)$ is defined by Equation (5). $s_i^j$ indicates the strength of motionlet $\mathcal{M}_j$ occurring in $\mathcal{V}_i$. We define the representative and discriminative measure of $\mathcal{M}_j$ as the ratio of between-class variance to within-class variance,

$$P_j = \frac{\sum_{k=1}^{K} N_k (\overline{s}_k^j - \overline{s}^j)^2}{\sum_{k=1}^{K} \sum_{\mathcal{V}_i \in C_k} (s_i^j - \overline{s}_k^j)^2}, \qquad (10)$$

where $K$ is the total number of action classes, $N_k$ is the

number of videos in action class $C_k$, and $\overline{s}_k^j$ and $\overline{s}^j$ are the means within class $C_k$ and over all classes,

$$\overline{s}_k^j = \frac{1}{N_k} \sum_{\mathcal{V}_i \in C_k} s_i^j \ , \quad \overline{s}^j = \frac{1}{\sum_{k=1}^K N_k} \sum_{k=1}^K N_k \overline{s}_k^j. \ (11)$$

With the measures above, we can rank motionlets and select those with high measures. However, this method treats each motionlet independently, and ignore the correlation between motionlets. This may result in a redundant set just covering a subset of training classes. We overcome this limitation by exploring the $k$ nearest videos of each motionlet in training samples, as shown in Figure 4. We call video $\mathcal{V}_i$ is '$k$ nearest' to motionlet $\mathcal{M}_j$, if matching result $s_i^j$ belongs to the $k$ largest value of $\{s_n^j\}$ ($n = 1, ..., N$ is index of training videos). We say motionlet $\mathcal{M}_j$ covers video $\mathcal{V}_i$ if $\mathcal{V}_i$ is in the $k$ nearest neighbors of $\mathcal{M}_j$.

Our goal is to find a subset of motionlets satisfying two requirements, the sum of representative and discriminative power should be as large as possible; the coverage percentage of training samples should be as high as possible. We design a greedy algorithm to select motionlets sequentially as shown in Algorithm 1. In each iteration, we first find the least covered training samples. Then, we search for the set of motionlets that cover these training samples. Finally, we greedily select the motionlet that has highest representative and discriminative power in this set.

### 4.4. Video Representation using Motionlet

With a set of motionlets $\mathbb{M} = \{\mathcal{M}_1, \mathcal{M}_2, ..., \mathcal{M}_m\}$, we can represent an action video $\mathcal{V}$ by a *motionlet activation vector* $f = [s^1, ..., s^m]$, where activation $s^j$ is the max pooling result for matching motionlet $\mathcal{M}_j$ with $\mathcal{V}$ (Equation (9)). We use a spatial-temporal pyramid representation of video for matching which has three layers $1 \times 1 \times 1$, $2 \times 2 \times 2$, and $1 \times 1 \times 4$. Thus the dimension of motionlet activation vector is $m \times 13$, where $m$ is number of motionlets. For classifier, we use linear SVM implemented by LIBSVM [6], and adopt one-versus-rest scheme to select the class with highest score for multi-class classification.

## 5. Experiment

We evaluate the effectiveness of motionlet on three datasets, one small scale dataset KTH [28] and two large scale datasets UCF50 [26] and HMDB51 [19]. KTH [28] consists of six human action classes and each action is performed several times by 25 subjects. The videos are recorded in a controlled settings with homogeneous background and static camera. In total, the data consists of 2,391 video clips and we follow the original experimental setup [28], i.e. 16 subjects for training and 9 subjects for testing, each long video is split into several short clips. UCF50 [26] and HMDB51 [19] are two large datasets for human action recognition. UCF50 has 50 action classes with total 6,618 videos, and each action class is divided into 25 groups with at least 100 videos for each class. HMDB51 has 51 action classes with total 6,766 videos and each action class has at least 100 videos. Videos in these two sets are from realistic environment such as, YouTube, Movies and Sports Video. For UCF50, we conduct experiments according to two kinds schemes: 5-fold group-wise cross validation (GV) [27] and Leave One Group Out cross validation (LOGO) [26]. For HMDB51, we use the original settings in [19] which include three training-testing splits. The final results are reported as the average of three splits.

**Visualization of Motionlets.** Some examples of motionelts are shown in Figure 6. From the results, we can see that video parts belonging to the same motionlets exhibit similar motion and appearance features. Motionlets can correspond to the motion of body part (such as upper body, leg) or visual phase (person-horse, gun-hand), and thus can yield important cues to recognize human motion category.

| Method | Accuracy (%) |
|---|---|
| Harris3D [20] + HOG/HOF [21] (from [30]) | 91.8 |
| Cuboids [9] + HOF3D [17] (from [30]) | 90.0 |
| Dense + HOF [21] (from [30] ) | 88.0 |
| Hessian [32]+ ESURF [32] (from [30]) | 81.4 |
| HMAX(C2) [15] | 91.7 |
| 3D CNN [16] | 90.2 |
| GRBM [29] | 90.0 |
| ISA (dense sampling) [22] | 91.4 |
| ISA (norm thresholding) [22] | 93.9 |
| ActionBank [27] | **98.0**[1] |
| Motionlet (1000) | 92.1 |
| Motionlet (3000) | 93.3 |

Table 2. Recognition accuracy in KTH [28]. We compare motionlet with low-level representation, high-level representation, deep learning based method.

**Recognition Results.** The experimental results on three datasets are shown in Table 2, Table 3, and Table 4. From these results, we see that the proposed motionlets achieve a comparable result on the simple dataset and high performance on the two large scale datasets. For KTH, our method get recognition accuracy $92.1\%$ for $1,000$ motionlets and $93.3\%$ for $3,000$ motionlets. Our result is comparable to low-level features [20, 9, 32] and deep learning based methods [15, 16, 29, 22]. Action bank [27] use a different testing settings and get the best result on KTH. For HMDB51, our method obtains a classification accuracy $32.1\%$ using $1,000$ motionlets and $33.7\%$ using 3,000 motionlets. These results yield 13 percents improvement over a baseline HOG/HOF (low-level representation), 7 percents

[1]They remove part of testing videos in the bank and they do not split each video into short clips according to [28], thus their testing settings is different from the other methods and ours.

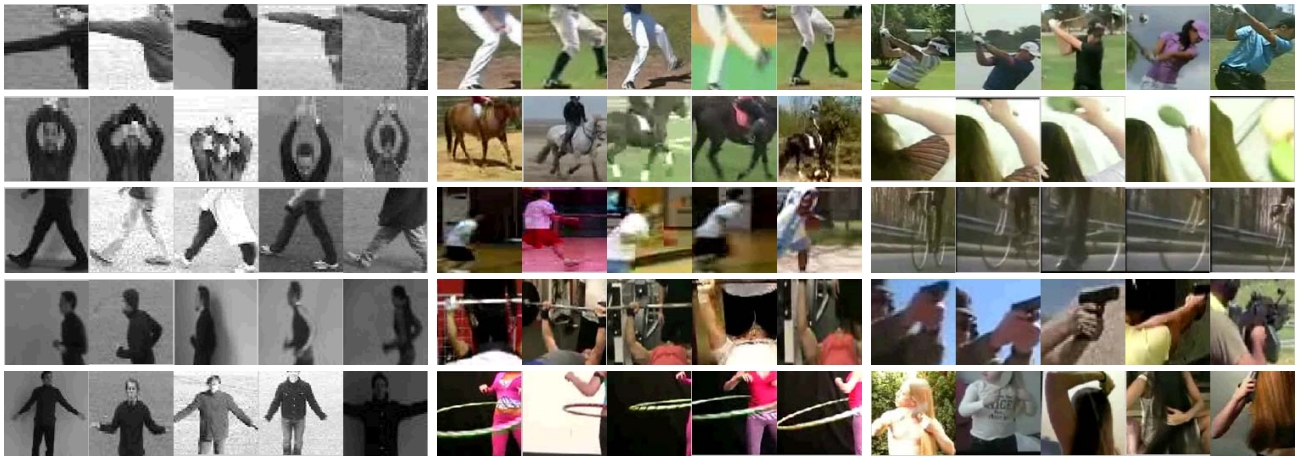

Figure 6. Examples of motionlet from three datasets: KTH (left), UCF50 (middle) and HMDB51 (right). We find each motionlet is a tight cluster both in motion and appearance space.

| Method | Accuracy (%) |
|---|---|
| Gist [24] (from [27]) | 13.4 |
| Harris3D [20] + HOG/HOF [21] (from [19]) | 20.2 |
| HMAX(C2) [15] (from [19]) | 23.2 |
| Motion Interchange Pattern [18] | 29.2 |
| Action Bank [27] | 26.9 |
| Motionlet (1000) | 32.1 |
| Motionlet (3000) | **33.7** |

Table 3. Recognition accuracy in HMDB51 [19]. We compare motionlet with low-level representation, high-level representation.

| Method | GV | LOGO |
|---|---|---|
| Gist [24] (from [27]) | 38.8 | - |
| Harris3D [20] + HOG/HOF [21] (from [27]) | 47.9 | - |
| Motion Interchange Pattern [18] | 68.5 | 72.7 |
| Action Bank [27] | 57.9 | - |
| Motionlet (1000) | 67.9 | 70.2 |
| Motionlet (3000) | **71.7** | **73.9** |

Table 4. Recognition accuracy in UCF50 [26]. We compare motionlet with low-level representation, high-level representation.

improvement over a recent method of action bank (high-level representation), and 4 percents improvement over a recent feature of motion interchange pattern (low-level representation) [18]. For UCF50, the proposed method obtains recognition accuracy of $67.9\%$ (1000 motionlets) and $71.7\%$ ($3,000$ motionlets) for GV. For LOGO, we obtain results of $70.2\%$ (1000 motionlets) and $73.9\%$ ($3,000$ motionlets). Our method outperforms HOG/HOF, Action Bank, and motion interchange pattern for both group wise cross validation (GV) and leave one group out cross validation schemes (LOGO). For computational cost, we extract motion saliency for about $30s$ and 3000 motionlets match for about $40s$ for each video on average on HMDB51 and UCF50 on a PC with E5645 CPU(2.4 GHZ) and 8G RAM.

From these comparisons, we can conclude that motionlet is effective in dealing with realistic videos. The diversity of realistic videos is much higher than the controlled videos of KTH. Local features like HOG/HOF cannot describe the complex motion information in realistic videos, while high level templates like action bank fail to deal with the large deformation among video samples very well. Due to the mid-level nature, motionlets yield a good tradeoff between low-level and high-level representation, and provide rich and robust information for classification.

**Varying number of motionlets.** We explore the influence of motionlet number and the effectiveness of motionlet selection algorithm using HMDB51 and UCF50 (GV). For HMDB51, there are totally $500 \times 51 = 25,500$ candidate motionlets, and for UCF50 there are totally $500 \times 50 = 25,000$ candidates. The results are shown in Figure 7, from which we can see that the accuracy increases little when the number of motionlets is larger than 2,000. These results indicates high redundancy within candidates, and thus it is necessary to conduct motionlet selection. We also make comparison between our motionlet selection method and random selection (we randomly select motionlets and repeat the random experiments 50 times). The results show that our method significantly outperforms the random ones. Besides, we can achieve a bit higher classification accuracy using selected motionlets than using all candidate motionlets. All these results imply that our greedy algorithm is effective in motionlet selection.

**Combined with other representations.** We use motionlets to obtain a mid-level representation of video. Since mid-level representation is complementary to low-level and high-level ones, we consider to combine these representations to further improve the performance. For low-level representation, we use traditional BoVW of STIP + HOG/HOF with 4,000 codewords. For high-level representation, we

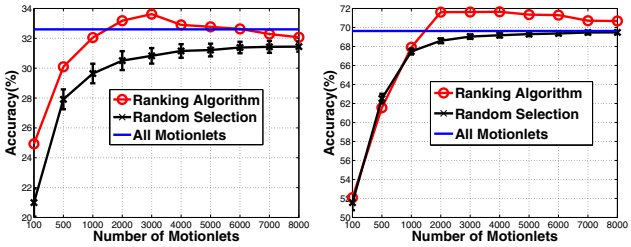


Figure 7. Results of varying motionlet size and compare ranking algorithm with random selection, Left: HMDB51 and Right: UCF50.

| Method | HMDB51 | UCF50 |
|---|---|---|
| Combined with Harris3D + HOG/HOF | 35.5 | 73.6 |
| Combined with Action Bank | 39.0 | 74.0 |
| Combine All | **42.1** | **78.4** |

Table 5. Recognition accuracy of combined representation in HMDB51 [19] and UCF50 [26].

use action bank representation with 205 detectors[1]. The number of motionlets are set as 3,000 in this combination. We simply concatenate the feature vectors of each representation and use linear SVM for classification. The results are shown in Table 5. We see that combination of any two representation can improve the performance. Combination of all three representations yields the state-of-the-art results on the two large scale datasets, $78.4\%$ for UCF50 (GV) and $42.1\%$ for HMDB51.

## 6. Conclusion

In this paper, we propose a mid-level video representation for motion recognition using motionlet. Motionlet are defined as a spatiotemporal part with coherent appearance and motion features. We develop a data-driven approach to learn motionlets by considering three properties, high motion saliency, multiple scale representation, and representative-discriminative ability. Compared with local features (such as STIP) and global template (such as action bank), motionlets are a mid-level parts and provide a good tradeoff between repeatability and discriminative ability. We evaluate the performance of motionlet on three public datasets, KTH, HMDB51 and UCF50. The experimental results show that our methods achieve significant improvements over recent published methods.

## References

[1] E. Adelson and J. Bergen. Spatiotemporal energy models for the perception of motion. *J. Opt. Soc. Am. A*, 2(2):284–299, 1985.

[2] J. K. Aggarwal and M. S. Ryoo. Human activity analysis: A review. *ACM Comput. Surv.*, 43(3):16, 2011.

[3] A. F. Bobick and J. W. Davis. The recognition of human movement using temporal templates. *TPAMI*, 23(3):257–267, 2001.

[1] Available at http://www.cse.buffalo.edu/ jcorso/r/actionbank/

[4] L. D. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009.

[5] W. Brendel and S. Todorovic. Learning spatiotemporal graphs of human activities. In *ICCV*, 2011.

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

[7] G. K. M. Cheung, S. Baker, and T. Kanade. Shape-from-silhouette of articulated objects and its use for human body kinematics estimation and motion capture. In *CVPR*, 2003.

[8] K. G. Derpanis, M. Sizintsev, K. J. Cannons, and R. P. Wildes. Efficient action spotting based on a spacetime oriented structure representation. In *CVPR*, 2010.

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

[10] P. F. Felzenszwalb, R. B. Girshick, D. A. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *TPAMI*, 32(9):1627–1645, 2010.

[11] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *IJCV*, 61(1):55–79, 2005.

[12] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *TPAMI*, 13(9):891–906, 1991.

[13] B. J. Frey and D. Dueck. Clustering by passing messages between data points. *Science*, 315:972–976, 2007.

[14] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *TPAMI*, 29(12):2247–2253, 2007.

[15] H. Jhuang, T. Serre, L. Wolf, and T. Poggio. A biologically inspired system for action recognition. In *ICCV*, 2007.

[16] S. Ji, W. Xu, M. Yang, and K. Yu. 3d convolutional neural networks for human action recognition. In *ICML*, 2010.

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

[18] O. Kliper-Gross, Y. Gurovich, T. Hassner, and L. Wolf. Motion interchange patterns for action recognition in unconstrained videos. In *ECCV*, 2012.

[19] H. Kuehne, H. Jhuang, E. Garrote, T. Poggio, and T. Serre. Hmdb: A large video database for human motion recognition. In *ICCV*, 2011.

[20] I. Laptev. On space-time interest points. *IJCV*, 64(2-3):107–123, 2005.

[21] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

[22] Q. V. Le, W. Y. Zou, S. Y. Yeung, and A. Y. Ng. Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis. In *CVPR*, 2011.

[23] J. C. Niebles, C.-W. Chen, and F.-F. Li. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010.

[24] A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001.

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

[26] K. K. Reddy and M. Shah. Recognizing 50 human action categories of web videos. *MVAJ*, 2012.

[27] S. Sadanand and J. J. Corso. Action bank: A high-level representation of activity in video. In *CVPR*, 2012.

[28] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: A local svm approach. In *ICPR*, 2004.

[29] G. W. Taylor, R. Fergus, Y. LeCun, and C. Bregler. Convolutional learning of spatio-temporal features. In *ECCV*, 2010.

[30] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009.

[31] X. Wang, L. Wang, and Y. Qiao. A comparative study of encoding, pooling and normalization methods for action recognition. In *ACCV*, 2012.

[32] G. Willems, T. Tuytelaars, and L. J. V. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *ECCV*, 2008.