Recognizing Realistic Actions from Videos "in the Wild"

Jingen Liu
Computer Vision Lab
University of Central Florida
liujg@cs.ucf.edu

Jiebo Luo
Kodak Research Labs
Eastman Kodak Company
jiebo.luo@kodak.com

Mubarak Shah
Computer Vision Lab
University of Central Florida
shah@cs.ucf.edu

Abstract

In this paper, we present a systematic framework for recognizing realistic actions from videos "in the wild." Such unconstrained videos are abundant in personal collections as well as on the web. Recognizing action from such videos has not been addressed extensively, primarily due to the tremendous variations that result from camera motion, background clutter, changes in object appearance, and scale, etc. The main challenge is how to extract reliable and informative features from the unconstrained videos. We extract both motion and static features from the videos. Since the raw features of both types are dense yet noisy, we propose strategies to prune these features. We use motion statistics to acquire stable motion features and clean static features. Furthermore, PageRank is used to mine the most informative static features. In order to further construct compact yet discriminative visual vocabularies, a divisive information-theoretic algorithm is employed to group semantically related features. Finally, AdaBoost is chosen to integrate all the heterogeneous yet complementary features for recognition. We have tested the framework on the KTH dataset and our own dataset consisting of 11 categories of actions collected from YouTube and personal videos, and have obtained impressive results for action recognition and action localization.

1. Introduction

Automatically recognizing human actions is receiving increasing attention due to its wide range of applications such as video indexing and retrieval, human-computer interaction, and activity monitoring. Although a large amount of research has been reported on action categorization, recognizing actions from realistic video still remains a quite challenging problem due to the significant intra-class variations, occlusion, and background clutter. In order to obtain reliable features, most early work made a number of strong assumptions about the videos, such as the availability of reliable human body tracking, slight or no camera motion, and limited number of viewpoints [3 ,5]. The commonly used KTH dataset contains relatively complicated scenarios, and many methods employing this dataset have been reported [8,9,10]. However, very few attempts have been made to recognize actions from videos

Figure 1: Examples of our YouTube action dataset consist of 11 categories with about 1160 videos.

"in the wild," as shown by the examples in Fig.1. Here, a video "in the wild" refers to a video captured under uncontrolled conditions, such as videos recoded by an amateur using a hand-held camera. Owing to the diverse video sources such as YouTube, TV broadcast and personal video collections, this type of video generally contains significant camera motion, background clutter, and changes in object appearance, scale, illumination conditions, and viewpoint. In this paper, our goal is to offer a generic framework for recognizing this type of realistic actions. Since we collected most of these videos from YouTube, hereafter, YouTube videos refer to videos "in the wild."

To the best of our knowledge, not much work has been reported on action recognition from unconstrained videos due to the difficulty in extracting good features from these videos. One related work is by Laptev *et al.* on recognizing actions from movies [19]. They collected a large and complicated action dataset from movies and employed

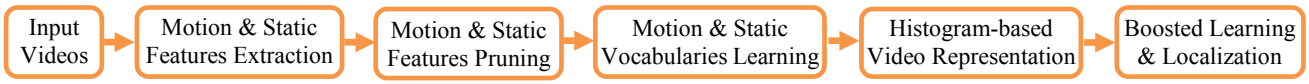


Figure 2: The flowchart of the training phase in our system.

various local motion features for recognition. Compared to the YouTube videos, the movie videos are of better quality, and contain no unintended camera motion. In addition, most actions in these videos are non-periodic. Other datasets include the UCF sports dataset [2] and Mikolajczyk's sports dataset [16]. Both of them contain a small number of actions and simple backgrounds.

Successful extraction of good features from videos is crucial to action recognition. Considering the large variation in realistic videos, local motion and static pose or posture features are more feasible to extract compared to 3D volumes, shapes, trajectories, etc. The bag of features (BOF) model can be used to capture the statistical information of the local motion or static features. BOF has been widely employed in object, scene, and action recognition [4,6,7,8,9,10,14,26,27,29] due to its simplicity and surprisingly good performance. Below, we briefly review the work on action and posture recognition using BOF.

Local Motion Features: Since the bag of video-words (BOV) approach does not require background subtraction and object tracking [3, 5], and can cope with certain camera motion and illumination changes, it is receiving increasing attention in generic action recognition [8,9,10,11,12,29]. Typically, spatiotemporal interest points are first detected either by a 3D Harris corner detector [11] or Gabor filters [12], and the descriptor vectors around those interest points are then computed and quantized into video-words whose statistical distributions are used to represent the entire video sequence. Beyond the BOV, the discriminative learning model such as SVM [11] and the generative model such as pLSA [10] have achieved excellent performance on the KTH dataset. Since BOV does not provide a spatiotemporal distribution of the features, the spatial correlogram [8], spatiotemporal pyramid matching [8, 19], and pLSA-ISM [10] were proposed to capture the spatial and temporal relationship between the local motion features and further improve the results.

Local Static Features: It is well known that the human vision system can recognize many types of human actions from a sequence of instantaneous postures or poses of a person in still images without motion information. Therefore, we believe the static pose in a single image can be useful for action categorization. Recently, pose recognition using local shape features such as shape context [13,15], histogram of gradients of the local patches[26], appearance, and position context [14,27] have obtained good results. Since a single pose only provides instantaneous information at a single instant, it is important to select the right pose in order to determine an action correctly. Instead of using a single pose, we can employ a sequence of poses, in order to make up for the lack of motion information. This is particularly useful for realistic videos where the motion features are unreliable due to unpredictable and often unintended camera motion (camera shake).

Hybrid Features: We strongly feel that local motion and static features are complementary for action recognition in unrestricted videos. For instance, suppose we want to differentiate *cycling* from *horseback riding* (see Fig. 1). Our observation is that videos of both actions may contain similar camera motion such as panning, which can result in similar motion features in both the background and the foreground. So it can be difficult to distinguish between the background and foreground based on only motion features. Yet, we humans can easily tell *bicycles* from *horses* based on their local shapes or appearance, thus static features may provide better recognition results in this case. On the other hand we may not be able to distinguish *jogging* from *running* based only on the pose information, and therefore must use motion features. To exploit the synergy, we choose to use hybrid features composed of both static features (local shape and appearance) and spatiotemporal motion features (local motion) to develop an effective recognition framework.

Little work has been reported on the combination of static and motion features for action recognition in realistic videos until recently. Fanti *et al.* [28] utilized a mixture of static features (local appearance) and dynamic features (simple velocity descriptors) for action recognition. Neibles *et al.* [15] proposed a generative model to learn a hierarchical model using both static and dynamic features for action recognition, and their results verified the hybrid features are useful. Liu *et al.* [4] proposed Fiedler Embedding to combine local motion features and spin image features that capture the global pose information. However, these methods may not be applicable for realistic videos due to the difficulty in acquiring good features in unconstrained videos. Instead of detecting spatiotemporal interest points, Mikolajczyk *et al.* [16] detected local static features with associated motion vectors from every single frame, and used motion vectors as a filter in recognition. Their action recognition method is akin to object recognition, and requires extra training images and object bounding boxes. Schindler *et al.* [17] combine different types of ST (spatiotemporal) features by simply concatenating the feature vectors.

1.1. Overview of the proposed framework

We present a systematic framework for action recognition in unrestricted videos based on BOF integrating both static and motion features. Fig. 2 depicts the flowchart of the proposed system. We make three main contributions in this work:

Motion and static feature pruning. Generally, in YouTube videos many motion features are detected in the

cluttered background due to the unpredicted camera motion. These motion features can adversely affect the recognition accuracy. In order to reduce their effects, we adopt an effective method to prune the motion features using spatial and temporal statistics.

The background in certain types of professional sports videos (e.g. football) can provide useful contextual information for action recognition since videos of the same sports tend to contain similar backgrounds. However, the background can vary significantly even for the same type of actions in unconstrained videos. Therefore, the background of unrestricted videos may not be that informative. We propose to utilize the spatial and temporal distribution of the motion to coarsely localize the region of interests (ROI). We believe the local shape or appearance information in the ROI, combined with (weak) contextual information from the background, provides the best opportunity for action recognition in unconstrained videos. In addition, we build a vast feature similarity graph by pair-wise image matching, and use PageRank (PR) [22] to select the significant static features. This method is capable of mining the features in the foreground in video sequences containing changing background.

Semantic visual vocabulary learning. The semantic visual vocabulary learning has two phases. In the first phase, we use k-means to create an initial vocabulary by grouping similar features based on their appearance. The initial vocabulary has two drawbacks. First, the performance is sensitive to the size of the vocabulary. Generally, larger vocabulary size performs better since the features are better quantized. Second, the visual words are not necessarily semantically meaningful, because k-means only considers the appearance similarity. For the sake of efficiency and effectiveness, compact yet discriminative semantic vocabularies are preferred. Liu, *et al.* [8] and Fulkerson *et al.* [1] used Information Bottleneck (IB) to obtain meaningful feature clusters. In the IB approach clusters are greedily merged in each iteration, this usually results in a suboptimal solution, so this makes it computationally expensive. Instead, we employ the divisive algorithm based on KL-divergence [20]. For each loop, it attempts to maintain the global optimal solution, and is more effective and efficient.

Heterogeneous features boosting. In the classification phase, we apply Adaboost to construct an effective final classifier through boosting of the heterogeneous features including motion and static features.

Our method can also localize the actions without explicit object detection and tracking. In summary, we propose a systematic framework for action recognition based on the following four steps: 1) use of motion statistics for feature pruning, and PR to further select important static features; 2) an information-theoretic divisive algorithm to learn the discriminative semantic visual vocabularies in order to make feature representation more compact and meaningful; 3) representation of the action videos by the histogram of bag of visual words; 4) combination of heterogeneous features by boosting for action recognition.

The proposed framework has been tested on both the standard KTH dataset and our unconstrained YouTube video dataset. Moreover, action localization can be provided both spatially and temporally thanks to the high quality of the motion and static features.

2. Visual Feature Extraction

2.1. Static feature detection

For every temporally sampled frame, we first apply three interest point detectors: Harris-Laplacian (HAR), Hessian-Laplacian (HES), and MSER detectors [25]. The three detectors produce complementary features: HAR locates corner features, and both HES and MSER extract blob features that are complementary to corner features. Next, each feature is described by its location (x, y), scale σ, and a 128-dimensional SIFT descriptor.

2.2. Motion feature detection and pruning

We use the spatiotemporal interest point detector proposed by Dollar *et al.* [11]. Compared to the 3D Harris-Corner detector, it produces dense features that can significantly improve the recognition performance in most cases. It utilizes two separate filters in spatial and temporal directions: 2-D Gaussian filter in space and 1-D Gabor filter in time. This detector produces high response to temporal intensity changes. The interest points are selected at the locations of local maximal responses of this detector, and 3D cuboids are extracted around them. For simplicity, we use the flat gradient vectors to describe the cuboids and then use PCA to reduce the descriptor dimension.

This motion feature extractor is effective and efficient. However, we have noted that in realistic videos, many features from the background may also be detected due to often unintended camera shake motion. One way to remove the camera motion is to perform motion compensation by registering frames using homography [16]. However, this method assumes scene with one dominating plane, while many of these videos normally contain multiple planes. Therefore, motion compensation may not work that well and is also computationally expensive. Instead of motion compensation, we employ an efficient feature pruning approach to remove irrelevant features corresponding to the background.

The major difficulty in detecting robust motion features captured by a moving or shaking camera is jittery motion, which may last only for a few frames. If those frames can be identified, then noisy features in those frames can be removed. Therefore, we propose to use feature statistics and the distribution of spatial locations to prescreen the features. Suppose a video has T frames, and frame F_t has N_t features. We propose two rules to prescreen the features.

Rule 1: If N_t > Mean (N) + Ω·Var(N), discard frame F_t , where Ω (e.g. 0.5) is a empirical parameter, and variable N represents the number of features in a frame.

This rule helps discard the frames with large unpredicted (unintended) camera motion and in turn helps efficiently remove the irrelevant motion features.

Let C(F_t) represent the mean location of all features (x_i^t, y_i^t) $(1 \le t \le T, 1 \le i \le N_t)$ present in F_t , $\delta(F_t)$ represents the neighboring frames of F_t, say F_{t-1} and F_{t+1}, and *Dist* be the distance between two locations, we define the Rule 2 as,

Rule 2: If *Dist*(C(F_t),C($\delta(F_t)$)) > η & *Diff*($N(F_t)$, $N(\delta(F_t))$) > γ, then select M/T number of features which are located close to C($\delta(F_t)$), where M is a predefined number of total features to be obtained.

Rule 2 can be used to predict the good features using the information about the neighboring frames. Our scheme to acquire good motion features is straightforward, but is very efficient and effective. Our experiments on the YouTube videos have shown that the average recognition accuracy can be improved by almost 8% by this measure.

3. Static Feature Pruning

In this section, we describe how to use motion cues and the PageRank to extract good static features from the foreground (i.e., region of interest).

3.1. ROI estimation by motion

Static features can help action recognition since they capture the pose or posture information in a sequence of frames. However, we are only interested in those static features that are located in the regions of interest identified by motion information. In realistic videos, the static features in the background may not be distinct. Therefore, we need to detect the region(s) of interest. The computation of ROI is as follows. Let $W = \{w_i(x_i, y_i, t_i) | 1 \le i \le n, t - \sigma \le t_i \le t + \sigma\}$ be a set of motion features extracted from frame $t - \sigma$ to frame $t + \sigma$ with time span of $2\sigma + 1$ (e.g. $\sigma = 10$). We can estimate the centroid of the ROI as,

$$\hat{x} = \frac{1}{n}\sum_i x_i \ , \hat{y} = \frac{1}{n}\sum_i y_i, \qquad (1)$$

and its dimensions are given by,

$$D_x = 2\sqrt{3c_{xx}} \ , D_y = 2\sqrt{3c_{yy}}, \qquad (2)$$

where c_{xx} and c_{yy} are the second central moments of the corresponding centroid. This approach works well for videos with relatively stable backgrounds.

3.2. Significant feature mining by PageRank

Some videos may have a constantly changing background, thus the motion information is not reliable. For this type of video, we propose to use PageRank (PR) techniques to discover the relatively important features. This is inspired by the successes of PR in the Google search engine and unsupervised object categorization [18]. In the case of a given video, we build a large directed graph of features. Here, a vertex denotes a feature, and an edge represents a match with another feature. If a feature is consistently matched with many other features, we consider it more significant than others. Since the background changes throughout the video, we consider a consistent feature foreground feature. The idea is similar to consistent feature tracking. PR is the right technique to analyze the interaction between the features, by assigning a ranking score to each feature as its relative significance in the feature network. It contains two major steps: visual similarity graph constructed by image matching and visual feature ranking by PR. We describe the procedure briefly as follows.

Figure 3: Two examples from *riding* (top) and *cycling* (bottom) demonstrate the effects of feature acquisition. The first row shows the original static features, and the second row shows the selected features. The top 10% features in PR values are retrieved.

3.2.1. Construction of feature similarity graph (FSG)

The FSG is a directed graph $G = (V, E, W)$, where V is the vertex set (the visual feature set), E is the edge set, and W is the associated adjacency matrix with weight representing the degree of match between the linked features. W is not necessarily symmetric. For instance, feature i in frame F_{t1} is selected to match feature j in frame F_{t2} when matching frame F_{t1} to F_{t2}, however when we try to match frame F_{t2} to F_{t1}, feature j' in F_{t2} may be the better match for feature i. Any image matching technique can be used here. We choose the spectral image matching technique in [24], which can find good matches with geometric consistency constraints.

In order to discover the discriminative foreground features, image matching is only performed on a pair of frames F_t and $F_{t'}$ with the constraint of $| t - t' | > \tau$ (τ, an empirical value, is set to 30 to 50 depending on the length of the video sequence). For matching frame F_t to $F_{t'}$, we initially re-

trieve n matched candidate feature pairs estimated by comparing the Euclidean distance between a pair of features represented by SIFT descriptor. Next, a graph with weighted adjacency matrix **P** (nxn) is constructed, where a node represents a pair of matched features (i, j), and edge weights are computed to measure the geometric consistency of two matches. For instance, if (i, j) and (i', j') are two pairs of candidates, then the entry **P**(ij,$i'j'$) contains the geometric consistency score between them. We think all the correct matches should have strong correlations with each other, while the incorrect ones are random outliers. The problem now is to cluster all the matches into two groups of S^* and R corresponding to inliers and outliers, respectively. This problem can be solved by spectral clustering to find the principle eigenvector of **P** matrix.

Once all the good matches (S^*) are obtained, we continue to re-estimate their matching score. For each pair of matches (i, j) (it corresponds to one edge in G) in S^*, we estimate the weight as $w_{ij} = \alpha w_{ij}^g + \beta w_{ij}^p$. The w_{ij}^g represents the geometric consistency, and w_{ij}^p measures the appearance similarity. The geometric consistency is computed by $S_{ij} = \sum_{i'j' \in S^*} P(ij, i'j')/|S^*|$. We then rank them by assigning them into different levels of matching (see paper [24] for details). The weight is estimated from the consistency level value with consideration of the total number of features. After matching all pairs of frames, we obtain an nxn sparse matrix W.

3.2.2. Feature ranking via PageRank

Given the constructed large graph G with its vertices and a set of edge weights, we want to measure the relative importance of the vertices using PR. Here, we can treat the VSG as a graph of linked WebPages. Each vertex is similar to a webpage and all the edge weights associated with a vertex can be considered as the *votes* cast by the linked vertices. Since the features from the foreground have more consistent matches throughout the entire video sequence, they get higher *votes*. The features in the background are, however, unstable due to the changing background, so their *votes* are lower. This is why we can discover the significant features using PR.

Suppose Pr is a $1xn$ PR vector with each entry corresponding to the PR value of the feature, we can solve the problem using the following equation:

$$Pr = \alpha * Pr * W + (\alpha * Pr * b + 1 - \alpha) * v,$$

where α is the scaling factor (α=0.85 in our experiment), b is an indicator vector indentifying the vertices with zero out-degree, W is the weights matrix, and v is an $nx1$ transport vector with uniform probability distribution over the vertices. The initial PR value for each vertex is $1/n$. For each frame F_t, we compute its PR vector Pr_t. Based on the rank of Pr values, we select the top μ features as the informative ones. Fig. 3 shows two examples of qualitative performance of our approach.

4. Learning Semantic Visual Vocabularies

In this section, we address the problem of obtaining compact yet discriminative visual vocabularies for motion and static features. We first create initial visual vocabularies with a relatively larger size. In general, a larger visual vocabulary performs better, but over-specific visual words may eventually over-fit the data. In addition, the initial vocabulary does not necessarily capture the semantic relations between the features. Therefore, we further use information-theoretic measure to refine the initial vocabularies by feature grouping. That is why very small vocabularies (e.g., two in [1]) can still achieve good performance [1,8]. Another motivation for vocabulary reconstruction is the fact that the combination of two features may be more useful than when used individually [21].

Given two distributions $p_1(x)$ and $p_2(x)$, the "distance" can be measured by Jensen-Shannon (JS) divergence as $JS_\pi(p_1, p_2) = \sum_{i=\{1,2\}} \pi_i KL\left(p_i, \sum_{j=\{1,2\}} \pi_j p_j\right)$, where $\pi_1 + \pi_2 = 1$, *and* $KL(p_1, p_2) = \sum_{x \in X} p_1(x) log(p_1(x)/p_2(x))$. JS-divergence is symmetric and finite compared to KL-divergence. Suppose variables $C = \{c_1, \ldots, c_L\}$ and $X = \{x_1, \ldots, x_M\}$represent classes and visual words respectively, then the information about C captured by X can be measured by mutual information (MI) I(C;X). Let $\hat{X} = \{\hat{x}_1, \ldots, \hat{x}_K\}$ be the visual words clusters of X; we can measure the quality of the new vocabulary as the loss of MI as, Q($\hat{X}$) = I(C;X) – I(C; $\hat{X}$), which can be computed as,

$$Q(\hat{X}) = \sum_{i=1}^{K} \pi(\hat{x}_i) JS(\{p(C|x_t) : x_t \in \hat{x}_i\}),$$

where $\pi(\hat{x}_i) = \sum_{x_t \in \hat{x}_i} \pi_t$, $\pi_t = p(x_t)$ is the prior, $\pi'_t = \pi_t / \pi(\hat{x}_i)$ for $x_t \in \hat{x}_i$, and $1 \le i \le K$, $1 \le t \le M$. From the equation, we can see the quality of the new cluster $\hat{x}_i$ is measured by the JS-divergence of every $p(C|w_t)$ in it. After some derivation, the new quality can also be written as,

$$Q(\hat{X}) = \sum_{i=1}^{K} \pi(\hat{x}_i) \sum_{x_t \in \hat{x}_i} \pi_t KL(p(C|x_t), p(C|\hat{x}_i)).$$

This equation suggests that the loss of MI due to vocabulary reconstruction can be considered as the dispersion of all the members ($p(C|x_t)$) to the new cluster center ($p(C|\hat{x}_i)$). Hence, we can use an iterative procedure like k-means algorithm to obtain the optimal new vocabulary using two major steps as follows:

1. For each cluster $\hat{x}_i$, compute the prior and "centers": $\pi(\hat{x}_i) = \sum_{x_t \in \hat{x}_i} \pi_t$ and $p(C|\hat{x}_i) = \sum_{x_t \in \hat{x}_i} \frac{\pi_t}{\pi(\hat{x}_i)} p(C|x_t)$;
2. Update clusters: for each x_t, find the new cluster: $i^*(x_t) = argmin_j\ KL(p(C|x_t), p(C|\hat{x}_j))$.

This iteration stops when $Q(\hat{X}) < \varepsilon$ (e.g. 10^{-3}). Compared to agglomerative IB (Information Bottleneck), this algorithm optimizes the global criteria, and is also more efficient with a complexity of $O(MKLS)$ compared to $O(M^3L)$ of IB, where S is the number of iterations (normally small).

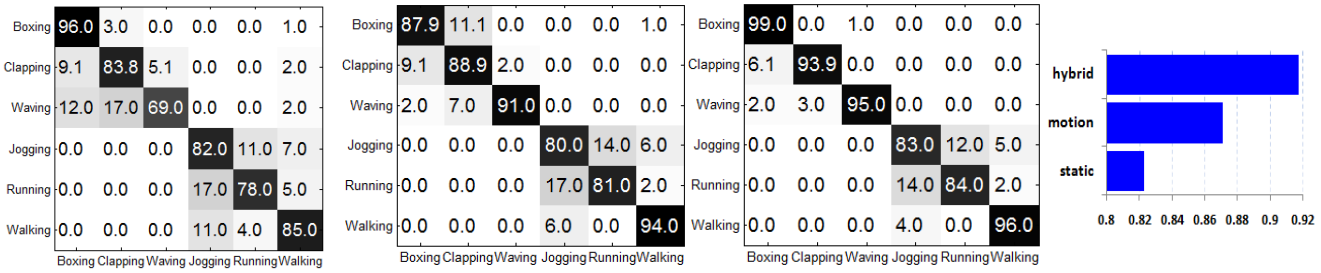


Figure 4: (a-c) Confusion tables for classification using static, motion and hybrid features, (d) the comparison of average accuracy.

5. Experiments and Discussion

We choose Adaboost with C.45 [23] as the classifier to combine the heterogeneous motion and static features. In the training phase, leave one out cross validation (LOOCV) scheme is used.

5.1. Datasets

The KTH dataset was recorded in a controlled setting with slight camera motion and "clean" background. However, it can still be used to test the benefit of using both motion and static features. The dataset contains six categories of actions: *boxing*, *clapping*, *waving*, *jogging*, *walking* and *running*. There were 25 actors performing each action four times in four different environments, resulting in about 600 video sequences in total.

Since the KTH dataset is relatively simple, both the motion and static features are mostly detected on the actors. We collected a more complex and challenging dataset based on YouTube videos and our personal video collections. Given that we do not have control over the video capturing process, the dataset has the following properties: 1) a mix of steady cameras and shaky cameras, 2) cluttered background, 3) variation in object scale, 4) varied viewpoint, 5) varied illumination, and 6) low resolution. This action dataset contains 11 categories: basketball *shooting (b_shooting)*, volleyball *spiking (v_spiking)*, trampoline *jumping* (*t_jumping*), soccer *juggling (s_juggling)*, horseback *riding(h_riding)*, cycling, diving, swinging, golf *swinging (g_swinging)*, tennis *swinging (t_swinging)*, and *walking (with a dog)*. The first four actions are easily confused with "*jumping*", the next two may have similar camera motion, and all the "swing" actions share some common motions. Some actions are also performed with objects such as a horse, bike or dog. Both static features and local contextual features can help in recognition. In order to remove the unfair effect of the same background in recognition, we organize the video sequences into 25 relatively independent groups, where separate groups are either taken in different environments or by different photographers. The dataset contains 1168 video sequences in total. To the best of our knowledge, this is the most extensive realistic action dataset in the vision community. We believe that the experimental results on this dataset will be very valuable considering that most previous research experiments were conducted within human-controlled settings to certain degrees. Fig. 1 shows some examples of the Youtube dataset.

5.2. Experiments on KTH dataset

We extracted about 400 cuboids (spatiotemporal volumes) and about 3,000 static features from each video, and applied PCA to reduce the dimension of the ST (spatiotemporal) feature descriptor to 100. Since the KTH dataset is relatively "clean", feature pruning is not necessary. We performed two groups of experiments. The objective of the first experiment is to demonstrate the benefit of combination of static and motion features. The objective of the second experiment is to show how compact and discriminative our learnt semantic visual vocabularies are. Fig. 4 shows the classification results for static features, motion features with initial vocabulary size of 600 respectively and the hybrid of them. The average accuracies are 82.3%, 87.1% and 91.8%, respectively. The improvement of using hybrid features is 4.7% over motion features alone. Note that the improvement is observed in *all* actions. It is surprising to note that we obtained much better results using the static features for *boxing* than using motion features. The reason is the fact that *boxing* has enough unique instantaneous poses. Both *clapping* and *waving* have some poses which overlap with *boxing*, so it is easy to misclassify them as *boxing*.

Table 1 Performance comparison between vocabularies generated by k-means and our information-theoretic method (%).

Size(N_w)	20	40	60	80	100	200	400
k-means	66.9	70.2	76.9	80.4	82.3	83.8	86.3
Proposed	84.1	85.1	86.8	87.6	88.8	90.8	89.1

Table 1 lists the performance comparison between the visual vocabularies generated by *k*-means and our semantic visual vocabularies (both are for motion features). The semantic visual vocabularies are learnt from an initial visual vocabulary of size 2000. The results support our conjecture that using mutual information between features and actions can result in a compact yet discriminative vocabulary, especially when the size is small. While we can only learn appearance similarity using k-means, we can further learn the semantic correlation between the features using our proposed method. In other words, the learnt vocabulary is semantically meaningful. This is consistent with the findings in [8]. We also learnt a semantic visual vocabulary of the static features from an initial visual vocabulary of size 2000, and achieved 84.3%. The hybrid combination of two resulted in **93.8%**. This is better than 91.8% reported in

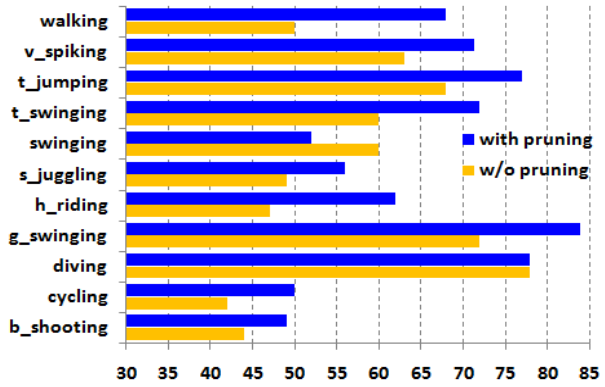


Figure 5: Performance comparison between systems with motion feature pruning and without feature pruning.

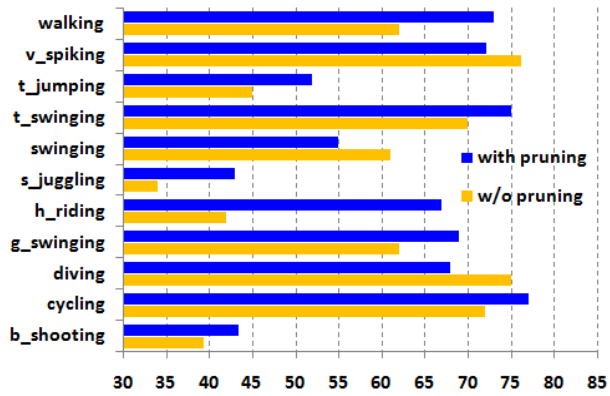


Figure 6: Performance comparison between systems with static feature mining and without feature mining.

[19], where multiple motion features with spatiotemporal information are combined for recognition. Our results are also comparable to those reported in [8] (i.e. 94.3%).

5.3. Experiments on YouTube Dataset

We extracted about 400 cuboids and a variable number of static features (about 7,000 ~ 14,000) from each video.

We first verified the effectiveness of our motion feature pruning technique. We used the initial vocabulary of size 2,000, and then learnt different sizes of semantic visual vocabularies, and selected 350 for experiments. Fig. 5 shows the comparison of the recognition accuracy between the experiments with and without using our motion feature pruning technique. The average accuracies are about 57.5% and 65.4%, respectively. The improvement is about 8%, which is impressive. For each individual action, the improvement varied between 4% to 18% with the exceptions of *swinging* and *diving*. One explanation is that the motion features from the background can help distinguish *swinging* from other similar *jumping* actions like *v_spiking and t_jumping*. Once these features are removed, it is easier to confuse *swinging* with other *jumping*. To verify our analysis, we checked the classification details. After feature pruning, about 16% and 8% *swinging* actions are misclassified into *t_jumping* and *v_spiking,* respectively, while the numbers are 13% and 4% before feature pruning.

We further conducted experiments to verify how good our static feature mining techniques are. The experimental results are reported on the initial vocabulary with a size of 2,000 and the learnt semantic vocabulary of size 400 is selected for further experiments. Fig. 6 shows the performance comparison between the experiments with and without applying feature mining. We obtained 5% to 25% improvement in recognition accuracy on eight actions. Specifically, we achieved 25% improvement for *h_riding*. This shows that PR can effectively discover the informative features. However, there are three categories: *v_spiking*, *swinging*, and *diving*, for which the performance decreased. The reason is that actions in one category took place in very similar environments. For instance, *v_spiking* normally happens in a crowd of people, and *diving* happens in a pool. This is common for professional sport actions which take place in highly structured environments. Overall, the average accuracy of all the categories improved from 58.1% to 63.0%.

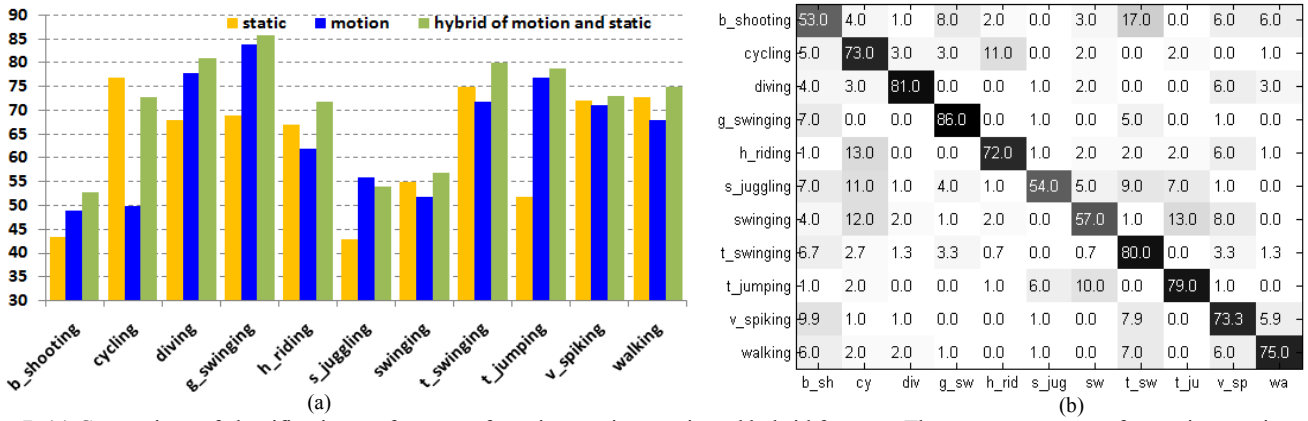
Finally, we verified the benefit of hybrid features. The initial vocabulary size is 2,000 for both motion and static features, and the size of the reconstructed vocabularies using our divisive information-theoretic approach is 350 and 400 for motion and static feature, respectively. Fig. 7 (a) shows the comparison of the classification accuracies using motion, static, and the hybrid features. The improvement using hybrid features is about 5.8% over motion features alone. Most categories obtained improvement in terms of recognition accuracy except for *s_juggling* and *cycling*. Fig. 7 (b) shows the confusion table for classification using the hybrid features. We can see that a lot of *b_shooting* is misclassified into *t_swing*, and *h_riding*, *s_juggling* and *swinging* are easier to be misclassified into *cycling*.

5.4. Action localization

By analyzing the location distribution of the good features (static or motion) that are discovered by using our motion and static feature pruning techniques, we can employ Equation 1 to estimate the centroid of the features in 2δ continued frames. This centroid is taken as the estimated center of the moving object with dimensions estimated by Equation 2. Fig. 8 shows a few recognition results with quite accurate action localization. As for the temporal localization, we employed a temporal sliding window and include a few examples in the supplemental material (due to limited space here).

6. Conclusions

We present a systematic framework for recognizing realistic actions from videos "in the wild," such as YouTube videos. In order to acquire good features, we use motion cues to prune motion and static features. In addition, we employ PageRank technique for informative static feature mining. We further use information-theoretic based divisive clustering to reconstruct compact yet discriminative semantic visual vocabularies. All the heterogeneous



(a) (b)

Figure 7: (a) Comparison of classification performance for using motion, static and hybrid features. The average accuracy for motion, static and hybrid features are 65.4%, 63.1% and 71.2%, respectively. (b) The confusion table for classification using hybrid features.

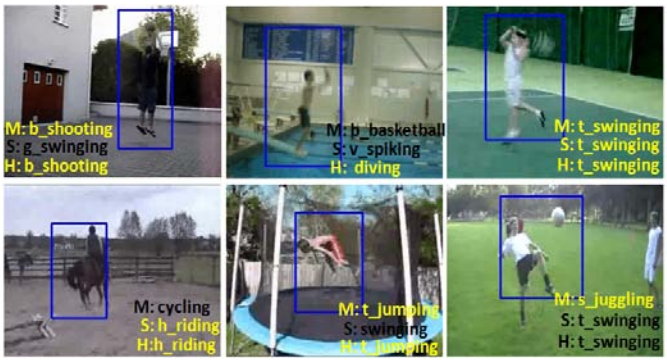


Figure 8: Some recognition results with localization. "M", "S" and "H" in the images means the following judgments are made on the "motion", "static" and "hybrid" features, respectively.

features are combined effectively by a boosting classifier. The experiments verified that our framework is effective for recognizing the realistic actions, and using hybrid features of motion and static can improve the average recognition accuracy.

7. References

[1] B. Fulkerson, A. Vedaldi and S. Soatto. Localizing objects with smart dictionaries, ECCV 2008.

[2] M. Sullivan and M. Shah. Action MACH: Maximum average correlation height filter for action recognition, CVPR 2008.

[3] D. Weinland, E. Boyer and R. Ronfard. Action recognition from arbitrary views using 3D examplars, ICCV 2007.

[4] J. Liu, S. Ali and M. Shah, Recognizing human actions using multiple features, CVPR 2008.

[5] V. Parameswaran and R. Chellappa. View invariance for human action recognition, IJCV, 66(1), 2006.

[6] F. Jurie and B. Triggs. Creating efficient codebooks for visual recognition, ICCV 2005.

[7] J. Sivic, B. Russell, A. Zisserman and W. Freeman. Discovering objects and their location in Images, ICCV 2005.

[8] J. Liu and M. Shah. Learning human action via information maximization, CVPR 2008.

[9] A. Gilbert and J. Illingworth and R. Bowden. Scale invariant action recognition using compound features mined from dense spatiotemporal corners, ECCV 2008.

[10] S. Wong, T. Kim and R. Cipolla. Learning motion categories using both semantics and structural information, CVPR 2007.

[11] P. Dollar, V. Rabaud, G. Cottrell and S. Belongie. Behavior recognition via sparse spatiotemporal features, VS-PETS 2005.

[12] I. Laptev and T. Lindeberg. Space-time interest points, ICCV 2003.

[13] Y. Wang, H. Jiang, M.S. Drew, Z. Li and G. Mori. Unsupervised discovery of action classes, CVPR 2006.

[14] H. Ning, Y. Hu and T.S. Huang. Discriminative learning of visual words for 3D human pose estimation, CVPR 2008.

[15] J. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification, CVPR 2007.

[16] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest, CVPR 2008.

[17] G. Schindler, L. Zitnick and M. Brown. Internet video category recognition, Internet Vision 2008.

[18] G. Kim, C. Faloutsos and M. Hebert. Unsupervised modeling of object categories using link analysis techniques, CVPR 2008.

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

[20] I. Dhillon, S. Mallsa and R. Kumar. A divisive information-theoretic feature clustering algorithm for text classification, JMLR, 3:1265-1287, 2003.

[21] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. JMLR, 3:1157-1182, 2003.

[22] S. Brin and L. Page. The anatomy of a large-scale hyper-textual web search engine, WWW 1998.

[23] J. R. Quinlan, Bagging, boosting, and c4.5, the 13th National Conference on AI, 1996.

[24] M. Leordeanu and M. Hebert. A spectral technique for correspondence problem using pairwise constraint, ICCV 2005.

[25] K. Mikolajczyk, *et. al.* A comparison of affine region detectors, IJCV 65:43-72, 2005.

[26] A. Bissacco, M. H. Yang and S. Soatto. Detecting humans with their pose, NIPS 2007.

[27] H. Ying, W. Xu, Y. Gong and T. S. Huang, Latent pose estimator for continuous action recognition, ECCV 2008.

[28] C Fanti, L. Zelnik-Manor and P. Perona. Hybrid models for human motion recognition, CVPR 2005.

[29] J. Yuan, Z. Liu and Y. Wu, Discriminative sub volume searches for efficient action detection, CVPR 2009.