Action Recognition with Motion-Appearance Vocabulary Forest

Krystian Mikolajczyk
University of Surrey
Guildford, UK
K.Mikolajczyk@surrey.ac.uk

Hirofumi Uemura
Kyushu Institute of Technology
Kitakyushu, Japan
H.Uemura@surrey.ac.uk

Abstract

In this paper we propose an approach for action recognition based on a vocabulary forest of local motion-appearance features. Large numbers of features with associated motion vectors are extracted from action data and are represented by many vocabulary trees. Features from a query sequence are matched to the trees and vote for action categories and their locations. Large number of trees make the process efficient and robust. The system is capable of simultaneous categorization and localization of actions using only a few frames per sequence. The approach obtains excellent performance on standard action recognition sequences. We perform large scale experiments on 17 challenging real action categories from olympic games[1] *. We demonstrate the robustness of our method to appearance variations, camera motion, scale change, asymmetric actions, background clutter and occlusion.*

1. Introduction

Significant progress has been made in classification of static scenes and action recognition is receiving more and more attention in computer vision community. Many existing methods [2, 5, 8, 18, 21, 24, 25] obtain high classification score for simple action sequences with exaggerated motion, static and uniform background in controlled environment. Example of a category used by these methods is displayed in Figure 1(left). It is however hard to make a visual correspondence to the real action of the same category displayed in Figure 1(right) as the appearance, motion and clutter of the scene is very different. Such scenes represent a real challenge which is rarely addressed in the literature. Our main goal in this paper is to propose a generic solution which could handle these type of actions but also to demonstrate how the performance for the controlled environment and the real one can differ.

The need for using real world data is argued in image recognition community [3]. The same direction should be followed in action classification, since the solutions proposed in both fields start to converge. Recently, a boosted space-time window classifier from [8] was applied to real movie sequences in [10]. However, boosting systems are known to require weak classifiers and large number of training examples to generalize, otherwise the performance is low. Other frequently followed class of approaches is based on spatio-temporal features computed globally [1, 4, 26] or locally [2, 5, 19, 24]. Both methods suffer from various drawbacks. Global methods cannot recognize multiple actions simultaneously or localize them spatially. In these methods recognition can be done by computing similarity between globally represented actions using cross-correlation [4] or histograms of spatio-temporal gradients [26]. Spatio-temporal interest points [9] result in a very compact representation but are too sparse to build action models robust to camera motion, background clutter, occlusion, motion blur etc. Moreover, local features are often used to represent the entire sequence as a distribution, which results in a global representation at the end. It was demonstrated in [25] that as few as 5 to 25 spatio-temporal interest points give high recognition performance on standard test data. We argue that this number is insufficient for real actions. The need for more features has been observed in [2], where Harris interest point detector was combined with Gabor filter to extract more spatio-temporal points. This argument was also emphasized by [5, 19], which propose a hybrid of spatio-temporal and static features to improve the recognition performance. This shifts the attention from motion towards the appearance of objects performing actions. In this context it seems more appropriate to address object-action categorization problem rather than action via motion only.

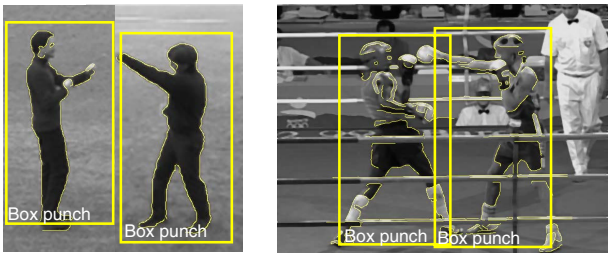

Figure 1. Examples of an object-action category in different environments.

A different class of approaches rely on a strong as-

[1] Video footage covering olympic games in Barcelona.

978-1-4244-2243-2/08/$25.00 ©2008 IEEE

sumption that body parts can be reliably tracked [23], even though existing tracking tools often fail in real video data. These methods use relatively large temporal extent and recognize more complex actions often viewed from multiple cameras, thus are less relevant to this work.

In this paper we address the problem of recognizing object-actions with a data driven method, which does not require long sequences or high level reasoning. The main contribution is a generic solution to action classification including localization of objects performing actions. We draw from existing work recently done in recognition and retrieval of static images [11, 15, 20]. Our approach follows the standard paradigm, which is the use of local features, vocabulary based representation and voting. Such systems have been very successful in retrieval and recognition of static images. However, recognition of actions is a distinct problem and a number of modifications must be proposed to adopt it to the new application scenario. Compared to existing approaches which usually focus on one of the issues associated with action recognition and make strong assumptions, our system can deal with appearance variations, camera motion, scale change, asymmetric actions, background clutter and occlusion. So far, very little was done to address all these issues simultaneously. The key idea explored here is the use of large number of features represented in many vocabulary trees in contrast to many existing action classification methods based on a single, small and flat codebook [2, 19, 24]. This message also comes from the static object recognition [3], where efficient search methods using many different features from a lot of data provide the best results. The advantage of using multiple trees has been demonstrated in image retrieval [22]. In this paper the trees are build various types of features, represent appearance-action models and are learnt efficiently from videos as well as from static images. Moreover, we use a simple NN classifier unlike the other methods based on SVM [2, 19, 24].

Among other contributions, we adopt Linear Discriminant Projections [7, 16] to the categorization problem. We implement an object-action representation which allows to hypothesize an action category, its location and pose from a single feature. We show how to make use of static training data and static features to support action hypothesis. In contrast to all the other systems our method can simultaneously classify the entire sequence as well as recognize and localize multiple actions within the sequence. Finally, we consider the use of new action categories and recognition results reported in this paper as one of our major contributions.

2. Overview of the recognition system

The main components of the system are illustrated in Fig. 2. The representation of object-action categories is based on multiple vocabulary trees. Training of the trees starts with feature extraction which includes scale invariant feature detection, motion estimation, and region description discussed in Sec. 3. The dimensionality of features is reduced with Linear Discriminant Projections [16]. Sec. 4 explains how the vocabulary forest is build from subsets of low dimensional features. Sec. 4.2 discusses the recognition where features and their motion vectors are first extracted from the query sequence. The descriptors are projected in low dimensional spaces and matched to the vocabulary trees. The features that match to the tree nodes accumulate scores for different categories and vote for their locations and scales within a frame. The learning and recognition process is very efficient due to the use of many trees and highly parallelized architecture discussed in Sec. 4.3. Finally, experimental results are presented in Sec. 5.

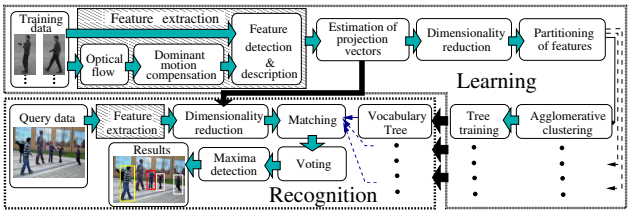

Figure 2. Overview of learning and recognition of object-action categories.

3. Motion-appearance representation

This section discusses feature extraction methods, dimensionality reduction as well as local and global motion estimation.

3.1. Appearance

Local features. The central part of our object-action representation are local features with associated motion vectors. Given the frames of action sequences we apply various state-of-the-art interest point detectors: MSER [14], Harris-Laplace and Hessian-Laplace [17]. These features proved very powerful in many recognition systems [11, 15, 20]. Inspired by pairs of adjacent segments from [6] we use similar method to extract edge segments but based on more efficient Canny detector. These features are robust to background clutter as only the connected edges are used to compute the descriptors. To obtain more features from the MSER detector we run it at multiple image scales and on red-green as well as blue-yellow projected images if color is available. Thus we obtain 5 types of image features which represent complementary patterns. MSER and Hessian-Laplace extract various types of blobs, Harris-Laplace finds corners and other junctions, pairs and triplets of edge segments represent contours. Each feature is described by a set of parameters : (x, y) - location, σ - scale, which determines the size of the measurement region, ϕ - dominant orientation angle, which is estimated from gradient orientations within the measurement region. Given these parameters we compute

GLOH features from [17]. Interest points are described with 17 bins in log-polar location grid and 8 orientation bins over 2π range of angles, thus 136 dimensional descriptor. The segment features use 17 bins in log-polar location grid and 6 orientation bins over π range of angles, resulting in 102 dimensions. There are 100s up to 1000s of features per frame in contrast to other action recognition methods [2, 24, 25] which extract only 10s of spatio-temporal features but do not deal with sequences containing more than one action, camera motion or complex background.

Dimensionality reduction. High dimensional features are very discriminative, slow to compute the similarity distance and make data structures for fast nearest neighbor search ineffective. Recently, a dimensionality reduction techniques more effective than PCA, yet based on global statistics was introduced in [7, 16]. Two global covariance matrices C and $\hat{C}$ are estimated for correctly matched and non-matched features, respectively, on image pairs representing the same scenes from different viewpoints. The matrices are then used to compute a set of projection vectors $\mathbf{v}$ by maximizing $J(\mathbf{v}) = \frac{\mathbf{v}^T \hat{C} \mathbf{v}}{\mathbf{v}^T C \mathbf{v}}$. Since correctly matched features are difficult to identify in category recognition we adopt a different strategy. Given the features extracted by a combination of a detector-descriptor (e.g. MSER and GLOH) we perform efficient agglomerative clustering [12] until the number of clusters is equal 10% of the number of features. In other words we expect a local image pattern represented by the cluster to occur on 10 training examples on average. Most of the resulting clusters are very compact and contain few features with only a few large clusters of indistinctive patterns. To prevent the domination of large clusters we use only those with less than 10 feature members, which is typically more than 90% of all clusters. Next, for each cluster member we generate 10 additional features by rotating, scaling and blurring its measurement region and computing new descriptors, which further populate the cluster. The covariance matrices are then estimated and we obtain the projection vectors $\mathbf{v}$. We select a number of eigenvectors associated with e largest eigenvalues to form the basis for linear projections. The parameter e is different for various feature types and automatically determined by the sum of eigenvalues which is equal to 80% of the sum of all eigenvalues. This typically results in 10 to 30 dimensions out of original 102 and 136 of GLOH. It leads to great reduction of memory requirements, increase of efficiency and most of all it makes the tree structures effective.

3.2. Motion

Motion maps are computed using standard implementation of Lucas-Kanade optical flow in image pyramids [13]. The motion is represented by velocity maps between pairs of frames. To remove erroneous vectors we run a median filter on each map.

Dominant motion compensation. Action sequences are often shot with significant camera motion or zoom. We found that the similarity transformation provides sufficient approximation for the dominant motion between two consecutive frames in most of the sequences we dealt with. Our dominant motion estimation starts by sampling points within the frames with the interval of 8 pixels. We select the interest point nearest to the sample, if there is one within 8x8 pixels, otherwise we extract an 8x8 patch centered on the sample point. These patches in addition to the interest points provide good coverage of the image and often contain sufficient texture or an edge to verify a match. Standard RANSAC is then applied to find the parameters of the global similarity transformation between frames using the motion vectors of the selected points. We accept the dominant motion vector if more than 20% of sparsely distributed points follow that motion, otherwise the dominant motion is assumed to be zero. Finally, the dominant motion is subtracted from the motion maps.

Local motion. For each appearance feature, dominant motion orientation angle is estimated within the measurement region using the motion maps. This is done by building motion orientation histogram and selecting the angle corresponding to the largest bin. We found that a single motion orientation angle per feature is sufficient as the interest point regions usually cover parts moving in the same direction. In a similar way we estimate the motion magnitude.

3.3. Action representation

The object-action categories are represented by appearance features with associated motion vectors extracted from pairs of consecutive frames. By using only pairs we avoid tracking issues with fast moving objects on complex background. Fig. 3 shows the representation with the parameters which allow to recognize and localize an object-action category. We use a star shape model to capture the global structure of an object and its moving parts. Similar model was successfully used for object detection in [11, 12, 15] and it is adapted here to actions. The training frames are annotated by bounding boxes which allow to estimate the size and the center of the training example. Each feature contains occurrence parameter vector $\mathbf{r} = [a, x, y, d, \sigma, \beta, \gamma, \phi, \mu]$; a - label of the action category, (x, y) - feature coordinates, d - distance to the object center, σ - scale (blue circles in Fig. 3), β - dominant orientation angle, γ - angle between the vector to the object center and the gradient orientation, ϕ - angle between the motion direction and the gradient orientation, and μ - motion magnitude. Angle γ is invariant to similarity transformations. With these parameters we can construct a local reference frame for every query feature and hypothesize pose parameters of an object-action category. A query feature can draw a hypothesis if its appearance and motion is similar to the model feature. The center

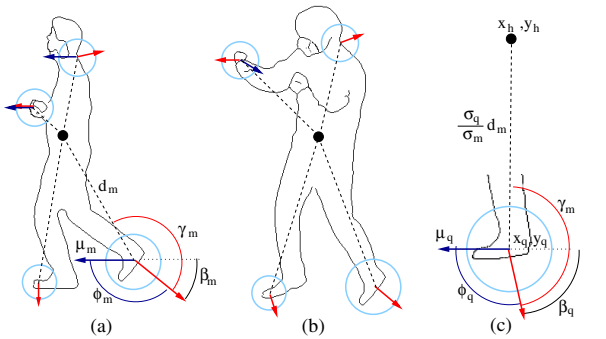

Figure 3. Object-action representation. Examples of features with motion-appearance parameters for jogging (a) and boxing (b). (c) Hypothesized object center based on a single feature.

of the hypothesis is computed by:

$$\begin{bmatrix} x_h \\ y_h \end{bmatrix} = \begin{bmatrix} x_q \\ y_q \end{bmatrix} + \frac{\sigma_q}{\sigma_m} d_m \begin{bmatrix} \cos(\beta_q - \gamma_m) \\ \sin(\beta_q - \gamma_m) \end{bmatrix} \quad (1)$$

where indexes q and m indicate the query and the model features, respectively. (x_h, y_h) is the location of the hypothesis within the image, σ_q/σ_m is the scale of the hypothesis, and $\beta_q - \beta_m$ is its orientation angle (see Fig. 3(c)).

The angle between the dominant gradient orientation and the dominant motion orientation of a feature is characteristic for a given time instance of an object-action category and it is used during recognition to validate a match between a query feature and a model feature. Note that some features do not move. These features are labeled static and will serve for refinement of object-action hypotheses, which is discussed in Sec. 4.2. In this representation many of the appearance features can be shared among various categories (see Fig. 3).

4. Vocabulary forest

The number of action examples is usually not large enough to build a generic model, in particular for object category appearance. To improve that we augment the training set by static images of our category objects, which are easier to obtain from the Internet or existing datasets than the videos. The features extracted from still images are labeled static. Once features all examples are extracted we build a set of trees. The features are first separated according to different types, which are combinations of detector-descriptor. Note that each type is projected with different set of vectors to reduce the number of dimensions and the features can be compared only within the same type. We start by partitioning the features from each type into subsets with kmeans. The argument to use kmeans instead of random splits exploited in [22] is that our objective is to cluster and compress the amount of information within each subset and not only to search for the nearest neighbors. A tree is constructed from each subset of features.

4.1. Tree construction

Clustering. The kmeans is initialized such that the subset contains less than $F = 200\,000$ features. A vocabulary tree is built with the agglomerative clustering which can handle this number of features within reasonable time. Initially each feature forms a cluster. Two nearest clusters in the whole set are merged at each iteration based on their Euclidean distance. We continue merging until one large cluster remains. This results in a binary tree of clusters where each node is represented by the average of its children and the size. The size of the node is given by the distance from the node center to its $(0.9F_n)$-th leaf child, where F_n is the number of the node's leaf children ordered by the distance. Factor 0.9 discards 10% of outliers when estimating the node size and makes the tree more compact. Finally, to compress the volume of the tree we remove the smallest clusters from the bottom of the tree until the remailing number of leaf nodes is 10% of the initial number of features F. See Fig. 4(a) for illustration.

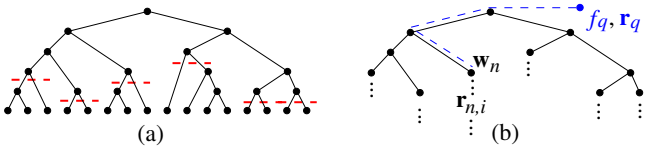

Figure 4. Vocabulary tree. (a) Binary clustering tree with cuts removing small clusters. (b) Vocabulary tree with node weights $\mathbf{w}_n$, parameters $\mathbf{r}_n$ and the path of query feature f_q.

Fast matching. The leaf nodes of the tree can be considered a codebook, which is similar to many other approaches based on codebooks. However, matching with a vocabulary tree is much more efficient than with flat codebooks in [2, 11, 18, 19, 24]. A query feature is first compared to the top node. If the distance from the query descriptor to the node center is less than the size of the node then the feature is accepted and compared to the children of this node. This continues until the query feature reaches the leaf nodes of the tree. We use this simple technique during training and recognition.

Training. We estimate weights $\mathbf{w}_n = [w_{n,0}, \ldots, w_{n,a}, \ldots]$ for each tree node. Weight $w_{n,a}$ indicates how discriminative node n is for category a. Weights are estimated by matching features extracted from all the training examples to the tree. We first estimate the probability to match node n by features from positive examples $p_{n,a} = \frac{F_{n,a}}{F_a}$, which is the ratio of the number of training features $F_{n,a}$ from a that matched to this node to the number of all features in this category F_a. We set $F_{n,a} = 1$ if no feature matched to node n. In addition to the positive training data we also use background category b which consists of image examples of scenes without our object-action categories. Background data is used to estimate $p_{n,b} = \frac{F_{n,b}}{F_b}$. The weight of the node is then given by $w_{n,a} = \log(\frac{p_{n,a}}{p_{n,b}})$. The top nodes usually have small weights as they match to many foreground and

background features. The weights tend to increase towards the bottom of the tree and the nodes become more discriminative for specific categories. In addition to the weight vector $\mathbf{w}_n$, each leaf node contains a list of parameter vectors $\mathbf{r}_{n,i} = [a_{n,i},\, d_{n,i},\, \sigma_{n,i},\, \beta_{n,i},\, \gamma_{n,i},\, \phi_{n,i},\, \mu_{n,i}]$ from the features that matched to this leaf node (see Fig. 4(b)). These parameters allow to hypothesize the positions and scales of object-actions during recognition. The nodes formed by features from static images also represent motion information. This information is transferred from motion features in action sequences that match to the static nodes.

4.2. Recognition

Given the trees the recognition starts by extracting features from the query sequence. To handle asymmetric object-actions we compute a symmetric copy of each query feature. This is done by swapping bins in the GLOH descriptor and inverting parameters $(x_q,\, \beta_q,\, \gamma_q,\, \phi_q)$ with respect to the vertical image axis. In this way with very little overhead we can handle various actions performed in different directions e.g. walking, even if the training set contains examples in one direction only. Next, the number of dimensions is reduced with the projection vectors estimated during training (cf. Sec. 3.1). The features are then matched to the trees. Each query feature f_q from frame t accumulates weights for different categories from all the nodes it matches to on its path to the leaf node (cf. Fig. 4(b)): $w_{a,t,f} = \sum_n k_{a,\phi} w_{n,a,f}$, where $k_{a,\phi}$ is the fraction of occurrence vectors $\mathbf{r}_{n,i}$ of class a for which the motion angles agree $|\phi_q - \phi_{n,i}| < T_\phi$. If f_q is static or motion angles are different the weight is labeled static. We keep only the weights accumulated by a query feature on its path to the nearest neighbor leaf node in each tree. Finally, we use 5 best paths from all trees. This matching strategy differs from the one in [20], where a single tree and a single path is used. From our observations, using a single path significantly reduces the number of good matches. Moreover, multiple paths allow to generate more votes for localization. Using many trees significantly improves the recognition performance compared to a single large tree which is demonstrated in Sec. 5.2.

Sequence classification. To classify the sequence we integrate the weights over motion features and frames: $w_a = \sum_t \sum_f w_{a,t,f}$. In contrast to [19] we do not use the static features here, otherwise they dominate the score and we cannot distinguish between similar categories e.g. running and jogging. The action is present if its accumulated weight w_a exceeds a fixed threshold. Thus, the classification is based on features in motion only.

Action localization. To find the location and size of the object performing an action we use the occurrence vectors $\mathbf{r}_{n,i}$ stored in the leaf nodes. A 3D voting space (x, y, σ) is created for each object-action category. Parameter vector $\mathbf{r}_q$ of query feature f_q that matches to leaf node n casts a vote in the 3D space for each parameter vector $\mathbf{r}_{n,i}$ stored in that node if the motion angles are similar $|\phi_q - \phi_{n,i}| < T_\phi$, otherwise the vote is labeled static. The weight $w_{a,t,f}$ is equally distributed among all the motion votes casted by this feature and the votes are stored in the corresponding bins in the voting space. The coordinates of the bins are computed with Eq. 1. Once all the features in motion cast their votes, the hypotheses are given by local 3D maxima in the voting space.

Refinement. Local maxima in the voting spaces are often drawn by only a few features in motion. We use static votes to improve the robustness of the recognition process. The voting space bin which corresponds to the local maximum and the neighboring ones are incremented by the weights of the static votes pointing to these bins. Thus, the motion based hypothesis is supported by the object appearance. This often changes the ranking of the hypotheses and improves the localization accuracy. If the action hypothesis is due to noise in the motion field, there are usually few additional static features that contribute to the score. The scores are thresholded to obtain the final list of object-actions with their positions and scales within the frame. In addition to that, from all the votes contributing to the local maximum we can compute a histogram of $\beta_q - \beta_{n,i}$ and estimate the global pose angle.

4.3. Efficient implementation

Many recognition systems work sequentially which requires large memory and high computational power. Alternatively GPU processors are deployed to speed up image processing. We adopt a different strategy to attain high efficiency of training and recognition. Our system is designed such that many operations can be done independently and simultaneously. This makes it possible to parallelize the training and recognition and run it with multiple processes on a single or many machines if available. The features are extracted and partitioned into subsets and all the trees are then trained in parallel. For example, 5000 frames of action sequences give approximately 3M features and result in 18 trees. It takes approximately 2h to train the system on eight P4 3GHz machines but running it in a sequential way takes 26h. It is also more efficient than the sequential way when run in parallel on a single machine. Estimating the training time without separating features into subsets was beyond our time constraints. Recognition takes 0.5s up to 10s per frame but it largely depends on the number of features extracted from the image.

5. Experiments

In this section the datasets and the evaluation criteria are discuss first. The results are then presented and compared to other methods.

action / test	hand **clap–** ping	hand **wav–** ing	**box–** ing	**jog–** ging	**walk–** ing	**run–** ning
#sequences	99	100	100	100	100	100
#frames annotated	2165	2570	2205	1354	2202	968
classification	.97	.96	.98	.88	.93	.87
state of the art	1.0[25]	.93[18]	1.0[18]	.75[25]	.90[2]	.88[18]
localization	.97	.96	.98	.79	.86	.78
static	.53	.71	.68	.57	.55	.59
motion	.83	.88	.84	.75	.81	.77
single tree	.79	.87	.85	.56	.66	.68
multiple trees	.85	.91	.89	.65	.76	.73
multi–KTH	.76	.81	.58	.51	.61	–

swim– ing	**row–** ing	**horse jump front**	**horse ride front**	**horse ride side**	**gym run front**	**gym salto front**	**gym salto side**	**sprint side**	**sprint front**	**sprint hurdle**	weig. **lift–** ing	weig. **pick–** ing	**cycl–** ing **front**	**cycl–** ing **side**	**box jump–** ing	**box punch–** ing
2	6	5	4	16	7	10	25	6	17	7	4	6	13	29	3	8
50	266	145	118	436	206	473	1139	139	410	181	111	155	311	725	76	174
.68	.91	.67	.71	.77	.81	.41	.47	.72	.76	.61	.81	.83	.53	.68	.73	.73
.61	.87	.53	.56	.61	.79	.32	.25	.58	.56	.47	.80	.81	.32	.65	.56	.53
.47	.82	.42	.44	.47	.63	.21	.19	.44	.43	.35	.78	.80	.22	48	.41	.31

17 Object–action categories from Olympic Games ,166 sequences, 5065 annotated frames

Figure 5. Action classification and detection results. (left) KTH action categories. (right) Olympic games categories.

5.1. Datasets

We use several datasets to train and evaluate the performance of our system. The KTH action sequences were introduced in [24] and frequently used in many action recognition papers [2, 18, 21, 25]. We present the results for this data and compare to the others methods. However, recognition performance for the KTH data has already saturated, we therefore acquire another sequence of actions included in the KTH set, but performed simultaneously with more complex background, occlusion and camera motion.

Olympic games are an abundant source of natural actions with high intra class similarities yet extremely challenging due to background clutter, large camera motion, motion blur and appearance variations. We select 10 different disciplines with various viewpoints and separate them in 17 action categories. The categories, the number of sequences and frames are summarized in Fig. 5. Image examples are displayed in Fig. 6. Each sequence contains more than one period of repetitive actions. We annotated every 5th frame of each sequence with bounding boxes using an interactive interface supported by color based tracking. In total, we annotated 11464 frames from 599 sequences of 6 KTH categories, 753 frames from multi-KTH sequence and 5065 frames from 166 sequences of 17 sport categories. In addition to the sequences we use images from Pascal set [3]: 1000 pedestrians, 200 horses and 200 bicycles, to capture large appearance variations. Finally, there are 1000 background images containing urban scenes and landscapes.

Performance measures. We evaluate the performance in a similar way to [2, 18, 21, 24, 25]. We refer to this test as 'classification'. In addition to that, we report results for detection of individual actions within frames, which we call 'localization'. The detection is correct if the intersection/union of the detected and the groundtruth bounding boxes is larger than 50% and the category label is correct. The detection results are presented with average precision, which is the area below precision-recall curve, as proposed in [3]. All the experiments are done with leave-one-out test, that is we train on all sequences except one in a given category, which is used for testing. Within that sequence, we perform recognition for every annotated frame and compare with the groundtruth. The results are averaged for all frames and all sequences of a given action. The temporal extent for integrating the votes is 5 frames (cf. Sec. 4.2).

5.2. KTH - Basic actions

Classification. We repeat the classification experiments from [2, 18, 25]. All action categories obtain high recognition score which favorably compares to the state-of-the art results, both are displayed in Fig. 5 (classification). It is interesting to observe that a system based on appearance with little motion information extracted from few frames can produce results comparable to a method that analyzes entire sequences but is based on very sparse features. There are small confusions between clapping and waving, as well as between walking, jogging and running that other approaches also suffer from. We also investigated the influence of the number of frames over which we integrate the votes. The results are high even if we use only a pair of frames as a query sequence. The score increases by up to 0.05 if we use more frames, which mainly helps in discriminating between similar categories e.g. waving - clapping, running - jogging. However, using more than 5 frames does not introduce significant improvements. Training on 16 sequences and testing on remaining 9, as done in [21, 24] produces similar results with a small drop of performance by 0.03 for running and jogging.

Localization. In addition to the classification we also present the results for recognition and localization in Fig. 5 (localization). Given that the KTH object-actions are on uniform background with exaggerated motion, the score for boxing, clapping, is as high as for the classification. A few missed detections for running, jogging and walking are due to incorrect scale estimation.

Multiple vs. single tree. In this experiment we demonstrate that multiple vocabulary trees are superior to a single large codebook. We reduce the number of training frames to 1000 by using very short sequences, to train a single tree for each feature type. For comparison we train from the same data a system with 5 vocabulary trees per type, thus 25 in total. We show in Fig. 5 (multiple trees) that the improvement is by up to 0.1 for walking. It is worth to mention that the

training and the recognition speed increases by a factor of 8. Multiple vocabulary trees allow to represent many variants of similar image patterns at different levels of details, thus the representation is richer and the probability that a query pattern is represented by some nodes is significantly higher than for a single tree or a flat codebook.

Motion vs. static features. In this test we investigate the impact of using the static features in addition to the motion ones. Some categories can be easily distinguished from a single frame without motion information due to very specific appearance of objects and background. In the context of the KTH data, static features are not discriminative enough and confusions between similar categories significantly increase if they are used for classification and initial localization (see Fig. 5 (static)). Location and scale estimation still works well as there is sufficient information in the appearance. Fig. 5 (motion) shows that the detection improves if based on features in motion, which corresponds to the classification and the initial localization in Sec 4.2. The results further improve if the refinement with static features follows the initial localization, which is shown in Fig. 5 (localization). Large number of static features help refine the hypotheses by increasing the score and improving estimation of pose parameters.

Multi-KTH. Fig. 6 (top row) shows frames from a sequence of 5 KTH actions performed simultaneously. We used the KTH data to train the system and the detection results are displayed in Fig. 5 (multi-KTH). Lower recognition rate than for KTH data is due to occlusion, camera motion, panning and zooming as well as differences between the background in training and testing sequences. This can be observed in the video sequence[2].

5.3. Sport actions

We demonstrated that our system can handle basic actions on static background or with camera motion. However, the real challenge is to recognize and localize real world actions filmed in uncontrolled environment.

Appearance-motion. Fig. 1 and Fig. 6 (row 2 to 4) show examples from 17 categories of sport actions. We perform the classification and localization tests as described in the previous section. Sport actions give more realistic estimates of recognition capabilities and the scores are significantly lower than for the KTH data. Some categories can be reliably recognized from static features only e.g. weight lifting or rowing. However, for the majority of object-actions motion information acts as focus-of-attention and allows to discard many features from the background. Note that it also excludes the context on which many image classification methods rely. We found that only 5% to 10% of query features are correctly matched with respect to both, appearance and motion. It is therefore essential to have a large number of features extracted from the query frames such that the initial voting is robust to noise. Similarly, static images used for training are essential for capturing large appearance variations. We observed an improvement of 0.09 for horse categories by using additional static data for training.

Motion vs. static features. This test corresponds to recognition based on the appearance only. It confirms the observations from the KTH data that static features tend to dominate in the classification and the performance for all categories is low (see Fig. 5 (static)). For example, for horse ride and jump, static features draw many false hypotheses due to significant clutter in these scenes. The motion constraint improves the results by up to 0.14. Unfortunately, features in motion can be significantly affected by motion blur which occurs in some categories e.g. gymnastics. Robustness to such effects is improved by using large number of features extracted with different detectors.

[2] Supplementary material

Conclusions

In this paper we proposed an approach for classification and localization of object-action categories. The system is capable of simultaneous recognition and localization of various object-actions within the same sequence. It works on data from uncontrolled environment with camera motion, background clutter and occlusion. The key idea here is the use of a large number of low dimensional local features represented in many vocabulary trees which capture joint appearance-motion information and allow for efficient recognition. We have conducted large scale experiments on unprecedented number of real action categories and demonstrated high capabilities of the system. We have also improved state-of-the art results on standard test data.

Possible improvements can be made in local motion estimation which is very noisy, in particular for small objects and for scenes with camera motion. Another direction to explore is tracking of individual features over longer time period to capture complex motion. This would also make the representation more discriminative and help in resolving ambiguities between similar actions e.g. jogging and running. Finally, the proposed system can be extended to recognize static as well as moving objects simultaneously using appearance and motion information when available.

Acknowledgment

This research was supported by EU VIDI-Video IST-2-045547 and UK EPSRC EP/F003420/1 grants. We would like to thank Richard Bowden, Falk Schubert and David Geronimo for their contribution to this work.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, 2005. 1

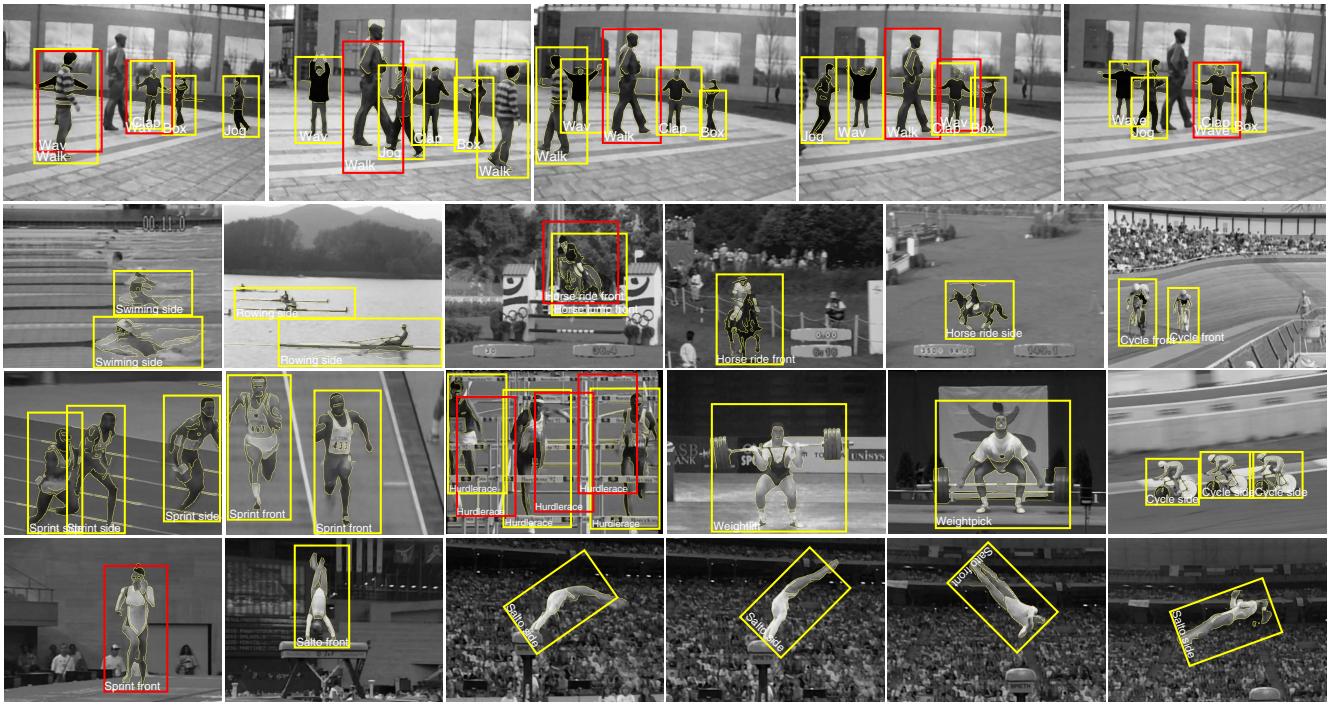

Figure 6. Examples of object-action categories with correct detections in yellow and false positives in red. (Top row) Frames from multi-KTH sequence. (Row 3) Pose angle was estimated only for this sequence and all recognition results are for fixed orientation. Some of the segment features that contributed to the score are displayed in yellow.

[2] P. Dollar, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *VS-PETS*, 2005. 1, 2, 3, 4, 6

[3] M. Everingham et. al. The PASCAL Visual Object Classes Challenge 2007. 1, 2, 6

[4] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, 2003. 1

[5] C. Fanti, L. Zelnik-Manor, and P. Perona. Hybrid models for human motion recognition. In *ICCV*, 2005. 1

[6] V. Ferrari, F. Jurie, and C. Schmid. Accurate object detection with deformable shape models learnt from images. In *CVPR*, 2007. 2

[7] G. Hua, M. Brown, and S. Winder. Discriminant Embedding for Local Image Descriptors. In *ICCV*, 2007. 2, 3

[8] Y. Ke, R. Sukthankar, and M. Hebert. Efficient visual event detection using volumetric features. In *ICCV*, 2005. 1

[9] I. Laptev. On space-time interest points. IJCV, 64:107123, 2005. 1

[10] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007. 1

[11] B. Leibe, A. Leonardis, B. Schiele Robust object detection with interleaved categorization and segmentation. IJCV, 2007. 2, 3, 4

[12] B. Leibe, K. Mikolajczyk, B. Schiele. Efficient clustering and matching for object class recognition. In *BMVC*, 2006. 3

[13] B.D. Lucas and T. Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision. In *IJCAI*, 1981. 3

[14] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *BMVC*, 2002. 2

[15] K. Mikolajczyk, B. Leibe, B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006. 2, 3

[16] K. Mikolajczyk and J. Matas. Improving Descriptors for Fast Tree Matching by Optimal Linear Projection. In *ICCV*, 2007. 2, 3

[17] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. PAMI, 27(10):1615–1630, 2005. 2, 3

[18] J.C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006. 1, 4, 6

[19] J.C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *CVPR*, 2007. 1, 2, 4, 5

[20] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 2, 5

[21] S. Nowozin, G. Bakir, K. Tsuda. Discriminative subsequence mining for action classification. In *ICCV*, 2007. 1, 6

[22] J. Philbin, O. Chum, M. Isard, J. Sivic and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2, 4

[23] D. Ramanan and D.A. Forsyth. Automatic annotation of everyday movements. In *NIPS*, 2004. 2

[24] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004. 1, 2, 3, 4, 6

[25] S. Wong and R. Cipolla. Extracting spatiotemporal interest points using global information. In *ICCV*, 2007. 1, 3, 6

[26] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, 2001. 1