Matching Mixtures of Curves for Human Action Recognition

Michalis Vrigkas[1], Vasileios Karavasilis[1], Christophoros Nikou[1], and Ioannis A. Kakadiaris[2]

[1] *Department of Computer Science, University of Ioannina, Ioannina, Greece*
[2] *Computational Biomedicine Lab, Department of Computer Science, University of Houston, Houston, Texas, USA*
{*mvrigkas, vkaravas, cnikou*}*@cs.uoi.gr, ioannisk@uh.edu*

Abstract

A learning-based framework for action representation and recognition relying on the description of an action by time series of optical flow motion features is presented. In the learning step, the motion curves representing each action are clustered using Gaussian mixture modeling (GMM). In the recognition step, the optical flow curves of a probe sequence are also clustered using a GMM, then each probe sequence is projected onto the training space and the probe curves are matched to the learned curves using a non-metric similarity function based on the longest common subsequence, which is robust to noise and provides an intuitive notion of similarity between curves. Alignment between the mean curves is performed using canonical time warping. Finally, the probe sequence is categorized to the learned action with the maximum similarity using a nearest neighbor classification scheme. We also present a variant of the method where the length of the time series is reduced by dimensionality reduction in both training and test phases, in order to smooth out the outliers, which are common in these type of sequences. Experimental results on KTH, UCF Sports and UCF YouTube action databases demonstrate the effectiveness of the proposed method.

Keywords:
Human action recognition; Optical flow; Motion curves; Gaussian mixture modeling (GMM); Clustering; Dimensionality reduction; Longest common subsequence.

1. Introduction

Action recognition is a preponderant and difficult task in computer vision. Many applications, including video surveillance systems, human-computer interaction, and robotics for human behavior characterization, require a multiple activity recognition system. The goal of human activity recognition is to examine activities from video sequences or still images. Motivated by this fact. our human activity recognition system aims to correctly classify a video into its activity category.

In this paper, we address the problem of human action recognition by representing an action with a set of clustered motion curves. Motion curves are generated by optical flow features which are then clustered using a different Gaussian mixture [1] for each distinct action. The optical flow curves of a probe sequence are also clustered using a Gaussian mixture model (GMM) and they are matched to the learned curves using a similarity function [2] relying on the longest common subsequence (LCSS) between curves and the canonical time warping (CTW) [3]. Linear [1] and non linear [4] dimensionality reduction methods may also be employed in order to remove outliers from the motion curves and reduce their lengths. The motion curve of a new probe video is projected onto its own subspace by a projection matrix specified by that video, and then the action label of the closest projection is selected according to the learned feature vectors as the identity of the probe sequence. The LCSS is robust to noise and provides an intuitive notion of similarity between cureves. Since different actors perform the same action in different manners and at different speeds, an advantage of the LCSS similarity is that it can handle with motion curves of varied lengths. On the other hand, CTW, which is based on the dynamic time warping [5], allows the spatio-temporal alignment between two human motion sequences. A preliminary version of this work was presented in [6]. One of the main contributions of this paper is that the training sequences do not need to have the same length. When a new probe sequence comes, it is matched against all the training sequences using the LCSS similarity measure. This measure provides a similarity between motion curves without enforcing one-to-one matching. An optimal matching is performed using dynamic programming, which detects similar pairs of curve segments [2].

However, training an action recognition system with only the knowledge of the motion of the current subject it is on its own a challenging task. The main problem is how we can ensure the continuity of the curves along time as an action occurs uniformly or non-uniformly within a video sequence. Unlike other approaches [7, 8], which use snippets of motion trajectories, our approach uses the full length of motion curves by tracking the optical flow features. Another question concerns the optimal model that one should adopt for recognizing human actions with high accuracy. This is accomplished by a statistical measure based on the data likelihood. The different lengths of the video sequences and therefore the respective lengths of the motion curves is another problem that is addressed. The large variance between benchmark datasets shows how the algorithm may be generalized. All these problems are discussed here and proper solutions are proposed. To this end, we have conducted experiments on several datasets [9, 10, 11] that would help us to understand how human activity recognition works.

Concatenating of optical flow features along time allows us to collect time series that preserve their continuity along time. It is true that correspondence is missing. However, this is the main assumption in many works [12, 13, 14]. If data association were used the resulting feature curves would have short duration and would be incomplete, as the features disappear and reappear due to occlusion, illumination, viewpoint changes and noise. In that case, a combination of sparse approach of clustering curves with variant lengths and tracking approaches should be used [15, 16]. This is not the central idea in this paper, as the nature of the feature curves drastically changes.

In the rest of the paper, the related work is presented in Section 2, while the extraction of motion curves, the clustering and the curve matching are presented in Section 3. In Section 4, we report results on the KTH [9], the UCF Sports [10] and the UCF YouTube [11] action classification datasets. Finally, conclusions are drawn in Section 5.

2. Related Work

The problem of categorizing a human action remains a challenging task that has attracted much research effort in the recent years. The surveys in [17] and [18] provide a good overview of the numerous papers on action/activity recognition and analyze the semantics of human activity categorization. Several feature extraction methods for describing and recognizing human actions have been proposed [12, 9, 19, 20, 13]. A major family of methods relies on optical flow which has proven to be an important cue. Efros *et al.* [12] recognize human actions from low-resolution sports video sequences using the nearest neighbor classifier, where humans are represented by windows of height of 30 pixels. The approach of Fathi and Mori [13] is based on mid-level motion features, which are also constructed directly from optical flow features. Moreover, Wang and Mori [14] employed motion features as inputs to hidden conditional random fields and support vector machine (SVM) classifiers. Real time classification and prediction of future actions is proposed by Morris and Trivedi [21], where an activity vocabulary is learnt through a three step procedure. Other optical flow-based methods which gained popularity are presented in [22, 23, 24].

The classification of a video sequence using local features in a spatio-temporal environment has also been given much consideration. Schuldt *et al.* [9] represent local events in a video using space-time features, while an SVM classifier is used to recognize an action. Gorelick *et al.* [25] considered actions as 3D space time silhouettes of moving humans. They take advantage of the Poisson equation solution to efficiently describe an action by utilizing spectral clustering between sequences of features and applying nearest neighbor classification to characterize an action. Niebles *et al.* [20] addressed the problem of action recognition by creating a codebook of space-time interest points. A hierarchical approach was followed by Jhuang *et al.* [19], where an input video is analyzed into several feature descriptors depending on their complexity. The final classification is performed by a multi-class SVM classifier. Dollár *et al.* [26] proposed spatio-temporal features based on cuboid descriptors. An action descriptor of histograms of interest points, relying on [9] was presented in [27]. Random forests for action representation have also been attracting widespread interest for action recognition [28, 29]. Furthermore, the key issue of how many frames are required to recognize an action is addressed by Schindler and Van Gool [30].

The problem of identifying multiple persons simultaneously and perform action recognition is presented in [31]. The authors considered that a person has first been localized by performing background subtraction techniques. Based on the Histograms of Oriented Gaussians [32] they detect a human, whereas classification of actions are made by training a SVM classifier. Action recognition using depth cameras are introduced in [33] and a new feature called "local occupancy pattern" is also proposed. A novel multi-view activity recognition method is presented in [34]. Descriptors from different views are connected together forming a new augmented feature that contains the transition between the different views. A new type of feature called the "Hankelet" is presented in [35]. This type of feature, which is formed by short tracklets, along with a BoW approach is able to recognize actions under different viewpoints, without requiring any camera calibration. Zhou and Wang [36] have also proposed a new representation of local spatio-temporal cuboids for action recognition. Low level features are encoded and classified via a kernelized SVM classifier, whereas a classification score denotes the confidence that a cuboid belongs to an atomic action. The new feature act as complementary material to the low-level feature.

Earlier approaches are based on describing actions by using dense trajectories. The work of Wang *et al.* [37] is focused on tracking dense sample point from video sequences using optical flow. Le *et al.* [38] discover the action label in an unsupervised manner by learning features directly from video data. A high-level representation of video sequences, called Action Bank, is presented by Sadanand and Corso [39]. Each video is represented as a set of action descriptors which are put in correspondence. The final classification is performed by a SVM classifier. Yan and Luo [27] have also proposed a new action descriptor based on spatial temporal interest points (STIP) [40]. In order to avoid overfitting they have also proposed a novel classification technique by combining the Adaboost and sparse representation algorithms. Wu *et al.* [41] employed, a visual feature using Gaussian mixture models efficiently represents the spatio-temporal context distributions between the interest point at several space and time scales. An action is represented by a set of features extracted by the interest points over the video sequence. Finally, a vocabulary based approach has been proposed by Kovashka and Grauman [42]. The main idea was to find the neighboring features around the detected interest points quantize them and form a vocabulary. Raptis *et al.* [8] proposed a mid-level approach extracting that spatio-temporal features construct clusters of trajectories, which can be considered as candidates of an action, and a graphical model is utilized to control these clusters.

Human action recognition using temporal templates has also been proposed by Bobick and Davis [43]. An action was represented by a motion template composed of a binary motion energy image (MEI) and a motion history image (MHI). Recognition was accomplished by matching pairs of MEI and MHI. A variation of the MEI idea was proposed by Ahmad and Lee [44], where the silhouette energy image (SEI) was proposed. The authors have also introduced several variability models to describe an action, and action classification was carried out using a variety of classifiers. Moreover, the proposed model is sensitive to illumination and background changes. In the sense of template matching techniques Rodriguez *et al.* [10] introduced the Maximum Average Correlation Height (MACH) filter which is a method for capturing intra-class variability by synthesizing a single action MACH filter for a given action class.

The main disadvantage of using a global representation such as optical flow, is the sensitivity to noise and partial occlusions. Space-time approaches can hardly recognize actions when more than one person are present in the scene. However, background subtraction can help to overcome this problem. Trajectory-based methods face the problem of human body detection and tracking as these are still open issues. Complex activities are more difficult to be recognized when space-time feature based approaches are employed. Furthermore, viewpoint invariance is another problem that these approaches must handle.

In the context described above, the present work has several contributions. First, human motion is represented by a small set of curves which are the mean curves of the mixture components along with their covariance matrices. The complexity of the model is considered low, as it is determined by the Bayesian Information Criterion (BIC), but any other model selection technique may be applied. Second, the computational cost is lower since the use of dimensionality reduction allows the algorithm to cope with curves of smaller lengths. Finally, the use of the longest common subsequence index allows input curves of different lengths to be compared reliably.

3. Action Representation and Recognition

Our goal is to analyze and interpret different classes of actions to build a model for human activity categorization. Given a collection of figure-centric sequences, we represent motion templates using optical flow [45] at each frame. Assuming that a bounding box can be automatically obtained from the image data, we define a rectangle region of interest (ROI) around the human. A brief overview of our approach is depicted in Figure 1. In the training mode, we assume that the video sequences contain only one actor performing only one action per frame. However, in the recognition mode, we allow more than one action per video frame. The optical flow vectors as well as the motion descriptors [12] for each sequence are computed. These motion descriptors are collected together to construct motion curves, which are clustered using a mixture model to describe a unique action. Then, the motion curves are clustered and each action is modeled by a set of clustered motion curves. Action recognition is performed by matching the clusters of motion curves of the probe sequence and the clustered curves in each training sequence.

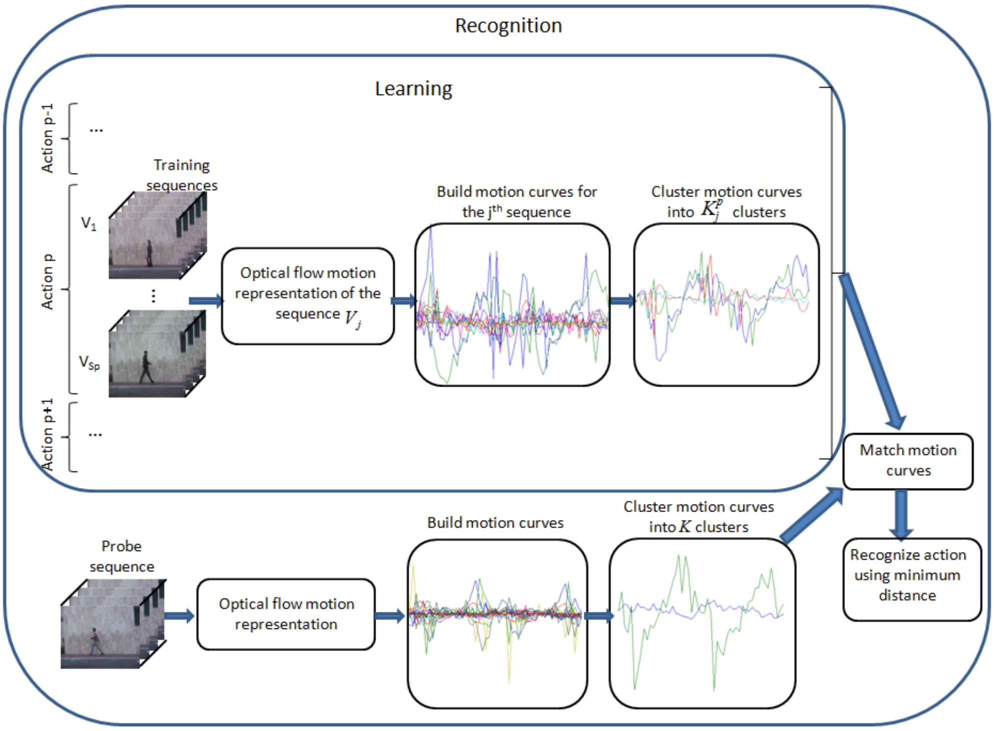

Recognition
Learning
Action p-1
...
Training sequences
V$_1$
Action p
⋮
V$_{Sp}$
Action p+1
...
Optical flow motion representation of the sequence V_j
Build motion curves for the jth sequence
Cluster motion curves into K_j^p clusters
Match motion curves
Probe sequence
Optical flow motion representation
Build motion curves
Cluster motion curves into K clusters
Recognize action using minimum distance

Figure 1: Overview of our approach.

3.1. Motion Representation

The proposed approach employs optical flow features [45]. These motion descriptors are commonly used in many recognition problems and they are shown to be quite reliable despite the existence of noisy features. Within a figure-centric scene, any human motion may be decomposed to the motion of different body parts (e.g., head and limbs). We can easily localize the motion by computing the optical flow vectors for the regions around the human torso.

Following the work of Efros *et al.* [12], we compute the motion descriptor for the ROI as a four-dimensional vector $\mathbf{F}_i = \left(F^+_{x_i}, F^-_{x_i}, F^+_{y_i}, F^-_{y_i}\right) \in \mathbb{R}^4$, where $i = 1, ..., N$, with N being the number of pixels in the ROI. Also, the matrix $\mathbf{F}$ refers to the blurred, motion compensated optical flow. We compute the optical flow $\mathbf{F}$, which has two components, the horizontal $\mathbf{F}_x$, and the vertical $\mathbf{F}_y$, at each pixel. It is worth noting that the horizontal and vertical components of the optical flow $\mathbf{F}_x$ and $\mathbf{F}_y$ are half-wave rectified into four non-negative channels F_x^+,F_x^-,F_y^+,F_y^-, so that $\mathbf{F}_x = F_x^+ - F_x^-$ and $\mathbf{F}_y = F_y^+ - F_y^-$. In the general case, optical flow is suffering from noisy measurements and analyzing data under these circumstances will lead to unstable results. To handle any motion artifacts due to camera movements, each half-wave motion compensated flow is blurred with a Gaussian kernel. In this way, the substantive motion information is preserved, while minor variations are discarded. Thus, any incorrectly computed flows are removed. Since all curves are considered normally distributed there is an intrinsic smoothing of the optical flow curves. Moreover, at a preprocessing step, we discard flows whose amplitude is over 20% of the standard deviation of the mean amplitude of all curves for each video.

3.2. Extraction of Motion Curves

A human action is represented by a set of primitive motion curves which are constructed directly from the optical flow motion descriptors. The main idea is to extract the salient features, which describe a relative motion from each frame and associate them with the corresponding feature in the next frame.

Consider T to be the number of image frames and C $=\{c_i(t)\}, t \in [0, T]$, is a set of motion curves for the set of pixels $i = 1, ..., N$ of the ROI. Each motion curve is described as a set of points corresponding to the optical flow vector extracted in the ROI. Specifically, we describe the motion at each pixel by the optical flow vector $\mathbf{F}_i = \left(F_{x_i}^+, F_{x_i}^-, F_{y_i}^+, F_{y_i}^-\right)$. A set of motion curves for a specific action is depicted in Figure 1. Given the set of motion descriptors for all frames, we construct the motion curves by following their optical flow components in consecutive frames. If there is no pixel displacement we consider a zero optical flow vector displacement for this pixel.

The set of motion curves describes completely the motion in the ROI. Once the motion curves are created, pixels and therefore curves that belong to the background are eliminated. We assume that the motion are normally distributed, thus, we keep flows whose values are inside 6 standard deviations of the amplitude distributions. In order to establish a correspondence between the motion curves and the actual motion, we perform clustering of the motion curves using a Gaussian mixture model. We estimate the characteristic motion which is represented by the mean trajectory of each cluster.

3.3. Motion Curves Clustering

A motion curve is considered to be a 2D time signal:

$$c_{ji}(t) = \left(F_{x_{ji}}(t), F_{y_{ji}}(t)\right), t \in [0, T], \tag{1}$$

where the index $i = 1, \dots, N$ represents the i^{th} pixel, for the j^{th} video sequence in the training set. To efficiently learn human action categories, each action is represented by a GMM by clustering the motion curves in every sequence of the training set. The p^{th} action ($p = 1, ..., A$), in the j^{th} video sequence ($j = 1, \dots, S_p$), is modeled by a set of K_j^p mean curves learned by a GMM. The likelihood of the i^{th} curve $c_{ji}^p(t)$ of the p^{th} action in the j^{th} video is given by:

$$p(c_{ji}^p; \pi_j^p, \mu_j^p, \Sigma_j^p) = \sum_{k=1}^{K_j^p} \pi_{jk}^p \mathcal{N}(c_{ji}^p(t); \mu_{jk}^p, \Sigma_{jk}^p), t \in [0, T], \tag{2}$$

where $\pi_j^p = \{\pi_{jk}^p\}_{k=1}^{K_j^p}$ are the mixing coefficients, $\mu_j^p = \{\mu_{jk}^p\}_{k=1}^{K_j^p}$ is the set of the mean curves and $\Sigma_j^p = \{\Sigma_{jk}^p\}_{k=1}^{K_j^p}$ is the set of covariance matrices. The covariance matrix in equation (2) is a diagonal $\Sigma_{jk}^p = diag(\sigma_{jk,1}^{2p}, \dots, \sigma_{jk,T}^{2p})$. Therefore, the log-likelihood of the p^{th} action in the j^{th} video can be written as:

$$L(c_j^p) = \prod_{i=1}^{N_j^p} \ln \sum_{k=1}^{K_j^p} \pi_{jk}^p \mathcal{N}(c_{ji}^p(t); \mu_{jk}^p, \Sigma_{jk}^p), t \in [0, T], \tag{3}$$

where N_j^p is the number of motion curves in the training set describing the p^{th} action in the j^{th} video.

The GMM is trained using the Expectation-Maximization (EM) algorithm [1], which provides a solution to the problem of estimating the model's parameters. The initialization of the EM algorithm is performed by the K-means algorithm. We have examined several configurations for the initialization of K-means and we decided to employ K-means with 50 different random initializations which were consistent and had no significant impact on the final classification. However, the number of mixture components should be determined. To select the number of the Gaussians K_j^p, for the j^{th} training video sequence, representing the p^{th} action, the Bayesian Information criterion (BIC) [1] is used:

$$BIC(c_j^p) = L(c_j^p(t)) - \frac{1}{2} M N_j^p, t \in [0, T], \tag{4}$$

where M is the number of parameters of the GMM to be inferred. Thus, when EM converges the cluster labels of the motion curves are obtained. This is schematically depicted in Figure 1, where a set of motion curves, representing a certain action (e.g., p), in a video sequence (e.g., labeled by j) is clustered by a GMM into $K_j^p = 2$ curves for action representation. Note that, a given action is generally represented by a varying number of mean curves as the BIC criterion may result in a different number of components in different sequences.

Apart from the BIC criterion, there are other techniques for determining the appropriateness of a model such as the Akaike Information Criterion (AIC) [1].

$$AIC(c_j^p) = L(c_j^p(t)) - M, t \in [0, T], \tag{5}$$

where M is the number of parameters of the GMM to be inferred. BIC is independent of the prior, it can measure the efficiency of the parameterized model in terms of predicting the data and it penalizes the complexity of the model, where complexity refers to the number of parameters in the model. It is also approximately equal to the minimum description length criterion [1] but with negative sign, it can be used to choose the number of clusters according to the intrinsic complexity present in a particular dataset and it is closely related to other penalized likelihood criteria such as the AIC. BIC tends to select highly parsimonious models, while AIC tends to include more parameters [46, 47]. Complexity measures such as BIC and AIC have the virtue of being easy to evaluate, but can also give misleading results.

3.4. Matching of Motion Curves

Once a new probe video is presented, where we must recognize the action depicted, the optical flow is computed, motion curves are created and clustered, and they are compared with the learned mean curves of the training set. Recall that human actions are not uniform sequences in time, since different individuals perform the same action in different manner and at different speeds. This means that motion curves have varied lengths. An optimal matching may be performed using dynamic programming which detects similar pairs of curve segments. The longest common subsequence (LCSS) [2] is robust to noise and provides a similarity measure between motion curves since not all points need to be matched.

Let $\mu(t)$, $t \in [0, T]$ and $\nu(\tau)$, $\tau \in [0, T']$ be two curves of different lengths.

Then, we define the affinity between the two curves as:

$$\alpha\left(\mu(t), \nu(\tau)\right) = \frac{LCSS\left(\mu(t), \nu(\tau)\right)}{\min(T, T^{'})}, \tag{6}$$

where the $LCSS\left(\mu(t), \nu(\tau)\right)$ (Eq. (7)) indicates the quality of the matching between the curves $\mu(t)$ and $\nu(\tau)$ and measures the number of the matching points between two curves of different lengths.

$$LCSS\left(\mu(t), \nu(\tau)\right) =$$
$$\begin{cases} 0, & \text{if } \ T = 0 \ \text{ or } \ T^{'} = 0, \\ 1 + LCSS\left(\mu(t)^{T_t - 1}, \nu(\tau)^{T^{'}_{\tau} - 1}\right), & \text{if } \ |\mu(t) - \nu(\tau)| < \varepsilon \ \text{ and } \ |T - T^{'}| < \delta \\ \max\left\{LCSS\left(\mu(t)^{T_t - 1}, \nu(\tau)^{T^{'}_{\tau}}\right), \ LCSS\left(\mu(t)^{T_t}, \nu(\tau)^{T^{'}_{\tau} - 1}\right)\right\}, & \text{otherwise} \end{cases} \tag{7}$$

Note that the LCSS is a modification of the edit distance [5] and its value is computed within a constant time window δ and a constant amplitude ε, that control the matching thresholds. The terms $\mu(t)^{T_t}$ and $\nu(\tau)^{T^{'}_{\tau}}$ denote the number of curve points up to time t and τ, accordingly. The idea is to match segments of curves by performing time stretching so that segments that lie close to each other (their temporal coordinates are within δ) can be matched if their amplitudes differ at most by ε (Fig. 2). A characteristic example of how two motion curves are matched is depicted in Figure 2.

When a probe video sequence is presented, its motion curves $z = \{z\}_{i=1}^{N}$ are clustered using GMMs of various numbers of components using the EM algorithm. The BIC criterion is employed to determine the optimal value of the number of Gaussians K, which represent the action in the probe sequence. Thus, we have a set of K mean curves ν_k, $k = 1, \ldots, K$ modeling the probe action, whose likelihood is given by:

$$L(z) = \prod_{i=1}^{N} \ln \sum_{k=1}^{K} \pi_k \mathcal{N}(z_i; \nu_k, \Sigma_k), \tag{8}$$

where Σ_k is the covariance matrix for the k^{th} component.

Recognition of the action present in the probe video sequence is performed by assigning the probe action to the action of the labeled sequence which is most similar. As both the probe sequence and the j^{th} labeled video

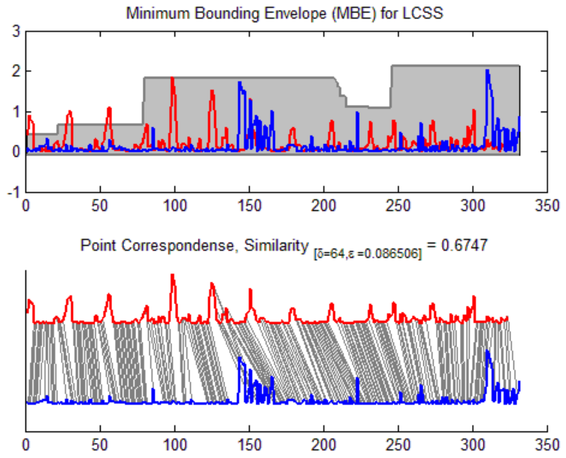


Figure 2: Depiction of the LCSS matching between two motions considering that they should be within $\delta = 64$ time steps in the horizontal axis and their amplitudes should differ at most by $\varepsilon = 0.086$.

sequence of the p^{th} action in the training set are represented by a number of mean curves $\nu = \{\nu_i\}_{i=1}^{K}$ and $\mu_j^p = \{\mu_{jk}^p\}_{k=1}^{K_j^p}$ respectively, the overall distance between them is computed by:

$$d(\mu_j^p, \nu) = \sum_{k=1}^{K_j^p} \sum_{\ell=1}^{K} \pi_{jk}^p \pi_\ell \left[1 - \alpha\left(\mu_{jk}^p(t), \nu_\ell(\tau)\right)\right], \tag{9}$$

where π_{jk}^p and π_ℓ are the GMM mixing proportions for the labeled and probe sequence, respectively, that is $\sum_k \pi_{jk}^p = 1$ and $\sum_\ell \pi_\ell = 1$. The probe sequence ν is categorized with respect to its minimum distance from an already learned action:

$$[j^*, p^*] = \arg\min_{j,p} d(\mu_j^p, \nu). \tag{10}$$

3.5. Canonical Time Warping

The canonical time warping (CTW) [3] solves the problem of spatio-temporal alignment of human motion between two time series. Based on dynamic time warping, the algorithm in [5] finds the temporal alignment of two subjects maximizing the spatial correlation between them. Given two time series $\mathcal{C}_1 = [c_1(0), \dots, c_1(T)]$ and $\mathcal{C}_2 = [c_2(0), \dots, c_2(T')]$ canonical time warping minimizes the following energy function:

$$J_{ctw}(\mathbf{W}_{\mathcal{C}_1}, \mathbf{W}_{\mathcal{C}_2}, \mathbf{V}_{\mathcal{C}_1}, \mathbf{V}_{\mathcal{C}_2}) = \|\mathbf{V}_{\mathcal{C}_1}^T \mathcal{C}_1 \mathbf{W}_{\mathcal{C}_1}^T - \mathbf{V}_{\mathcal{C}_2}^T \mathcal{C}_2 \mathbf{W}_{\mathcal{C}_2}^T\|_F^2, \tag{11}$$

where $\mathbf{W}_{\mathcal{C}_1}$ and $\mathbf{W}_{\mathcal{C}2}$ are binary selection matrices that need to be inferred to align $\mathcal{C}_1$ and $\mathcal{C}_2$, and $\mathbf{V}_{\mathcal{C}_1}$, $\mathbf{V}_{\mathcal{C}_2}$ parameterize the spatial warping by projecting sequences into the same coordinate system.

3.6. Dimensionality Reduction

Dimensionality reduction methods [4] may be employed in order to reduce the dimension of the motion curves and to enforce them to be of equal length. In the experiments, Principal Complements Analysis (PCA) [48] was chosen as a simple linear method but any other non-linear technique [4] could also be applied. When PCA is employed the time ordering is suppressed and curves are then transformed into feature vectors. In that case, the Bhattacharyya distance [5] is (among others) an appropriate matching measure.

Let v_1 and v_2, be two feature vectors following Gaussian distributions, with means μ_1 and μ_2 and covariance matrices Σ_1 and Σ_2, respectively. The Bhattacharyya distance has the form :

$$d_B(v_{1j}^p, v_2) = \frac{1}{8}(\mu_1 - \mu_2)^T \left(\frac{\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2}{2}\right)^{-1} (\mu_1 - \mu_2) + \frac{1}{2} \ln \left(\frac{\frac{\mathbf{\Sigma}_1 - \mathbf{\Sigma}_2}{2}}{2\sqrt{|\mathbf{\Sigma}_1||\mathbf{\Sigma}_2|}} \right). \tag{12}$$

To perform the match, one can project a probe video feature vector $v_2 = \{v_{2i}\}_{i=1}^K$ onto its own subspace by a projection matrix specified to that video and assume the label that lies closer than all the training feature vectors $v_{1j}^p = \{v_{1jk}^p\}_{k=1}^{K_j^p}$. For Gaussian mixture models, we define the Bhattacharyya distance as:

$$d_{GMM}(v_{1j}^p, v_2) = \sum_{k=1}^{K_j^p} \sum_{\ell=1}^{K} \pi_{jk}^p \pi_\ell d_B(v_{1jk}^p, v_{2\ell}), \tag{13}$$

where π_{jk}^p and π_ℓ are the GMM mixing proportions for the labeled and probe sequence, respectively. This is common in GMM modeling [49]. The probe feature vector v_2 is categorized with respect to the minimum distance from an already learned action:

$$[j^*, p^*] = \arg\min_{j,p} d_{GMM}(v_{1j}^p, v_2). \tag{14}$$

The overall approach for learning an action and categorizing a probe are summarized in Algorithm 1 and Algorithm 2, respectively. The steps inside the parenthesis indicate the extra steps when PCA is employed.

Algorithm 1 Action learning

```
Input:    Training video sequences
Output:   GMMs summarizing each action in each sequence

• For each action
    – For each video sequence representing the action
        ∗ Compute the optical flow at each pixel and generate half-wave
          rectified features [12].
        ∗ Construct the motion curves by concatenating the optical flow
          features.
        ∗ (Perform dimensionality reduction of the motion curves.)
        ∗ Cluster the motion curves by training GMMs with varying
          number of components and select the model maximizing the
          BIC criterion [1].
```

4. Experimental Results

In what follows, we refer to our mixtures of curves action recognition method by the acronym TMAR. We evaluated the proposed method on action recognition by conducting a set of experiments over publicly available datasets. First, we applied the algorithm to the KTH dataset [9]. This dataset consists of 2.391 sequences and contains six types of human actions such as walking, jogging, running, boxing, hand waving, and hand clapping. These actions are repeatedly performed by 25 different people in four different environments: outdoors (s1), outdoors with scale variation (s2), outdoors with different clothes (s3), and indoors (s4). The video sequences were acquired using a static camera and include a uniform background. The average length of the video sequences is four seconds, while they were downsampled to a spatial resolution of 160×120 pixels. Figure 3 depicts sample snapshots from the KTH dataset.

Algorithm 2 Action categorization

Input: A probe video sequence to be categorized and the GMMs summarizing the actions in the training sequences

Output: Action label

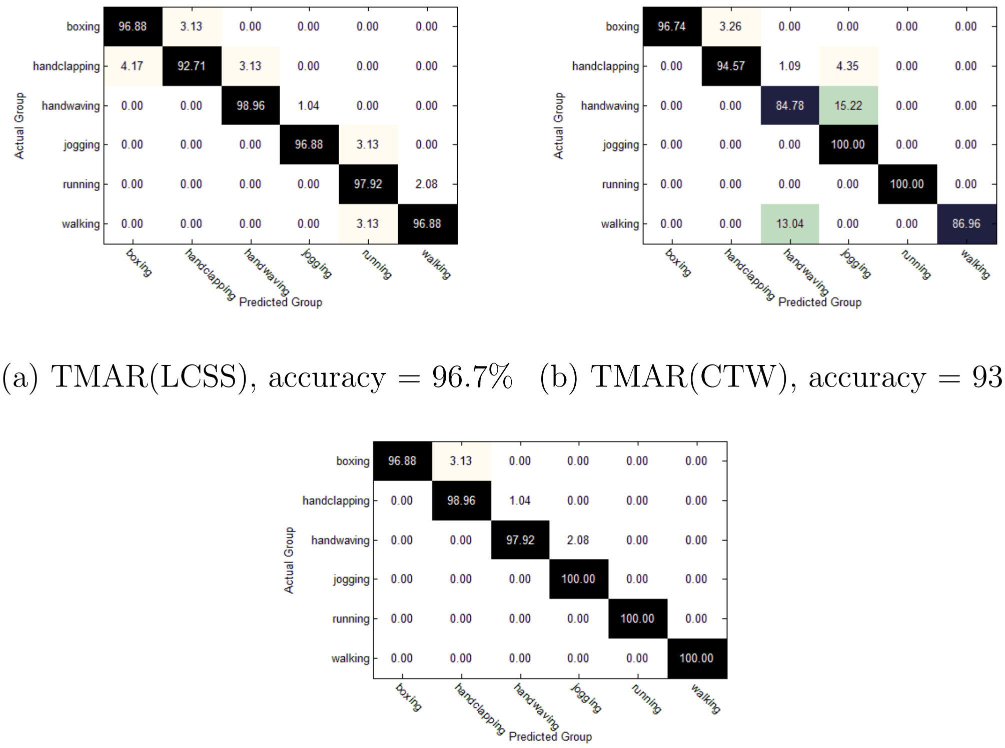
- Compute the optical flow at each pixel of the probe sequence and generate half-wave rectified features [12].
- Construct the motion curves by concatenating the optical flow features.
- (Project the motion curves onto their own subspace by a projection matrix.)
- Cluster the motion curves by training GMMs with varying number of components and select the model maximizing the BIC criterion [1].
- Compute the distances between the GMM of the probe sequence and each GMM of the learnt actions.
- Classify the probe sequence using a nearest neighbor classifier.

Figure 3: Sample frames from video sequences of the KTH dataset [9].

We tested the action recognition performance of the proposed method by using a leave-one-out cross validation approach. Accordingly, the model from the videos of 24 subjects was learned while the algorithm was tested on the remaining subjects and averaged the recognition results. The confusion matrices over the KTH dataset for this leave-one-out approach are shown in Figure 4. A recognition rate of 96.7% was achieved when only the BIC criterion was employed in conjunction with the LCSS metric, 93.8% when the CTW alignment is employed, and 98.3% using PCA. In addition, comparison of the proposed method with other state-of-the-art methods is reported in Table 1. The results of [9, 19, 13, 20, 22, 50, 14, 41, 38, 27, 39] are obtained from the original papers. Note that, the TMAR approach provides the more accurate recognition rates. Notice that all motion curves are reduced to a length that explains the 90% of the eigenvalue sum, which results in a reduced feature vector length of 50 instances with respect to the original 3.000 time instances.

In order to examine the behavior and the consistency of the method to the BIC criterion, we have also applied the algorithm without using BIC but having a predetermined number of Gaussian components for both the training and the test steps. Therefore, we fixed the number of Gaussians K_j^p to values varying from one to the square root of the maximum number of the motion curves and executed the algorithm. The TMAR(LCSS) approach attains high action classification accuracy as the BIC criterion determines the optimal value of Gaussians K_j^p for this dataset. Figure 5 depicts the accuracy rate for the TMAR(LCSS), TMAR(CTW) and TMAR(PCA) approaches with respect to the number of mixture components. As the number of curves representing each action is relatively small (30–60 curves per action), a large number of Gaussian components may lead to model overfitting. As the number of Gaussians is $K_j^p \geq 3$ for the TMAR(LCSS), $K_j^p \geq 5$ for the TMAR(CTW) and $K_j^p \geq 4$ for the TMAR(PCA) the accuracy rate drastically falls. This fact indicates the dependency of the recognition accuracy over the number of Gaussian components as an action is represented by few motion curves.

In order to provide a statistical evidence of the recognition accuracy we present some statistical indices (Tables 2 and 3). The p-value is the probability of obtaining a statistical test at least as extreme as the one that was actually observed, assuming that the null hypothesis is true. A small p-value ($p \leq 0.05$) indicates strong evidence against the null hypothesis, so we reject the null hypothesis. A large p-value ($p > 0.05$) indicates weak evidence against the null hypothesis, so we fail to reject the null hypothesis. Specifically, the null hypothesis was set to H_0: the recognition results of the state-of-the-art methods are better than the proposed and the alternative hypothesis is defined as H_α: the proposed method outperforms the state-of-the-art methods. In Table 2 and Table 3, statistical measurements for the



(a) TMAR(LCSS), accuracy = 96.7%

(b) TMAR(CTW), accuracy = 93.8%

(c) TMAR(PCA), accuracy = 98.3%

Figure 4: Confusion matrices of the classification results for the KTH dataset for (a) the proposed method denoted by TMAR(LCSS), (b) the the proposed method using the CTW alignment, denoted by TMAR(CTW), and (c) the proposed method using PCA, denoted by TMAR(PCA), for the estimation of the number of components using the BIC criterion.

KTH dataset are shown. TMAR(LCSS-BIC) and TMAR(PCA-BIC) reject the null hypothesis in the majority of the cases while, TMAR(CTW-BIC) rejects the null hypothesis in only two cases. Thus, the statistical significance meaning holds for TMAR(LCSS-BIC) and TMAR(PCA-BIC).

We have also applied our algorithm to the UCF Sports dataset [10]. This dataset consists of nine main actions such as diving, golf-swinging, kicking, lifting, horse riding, running, skating, swinging and walking. The dataset contains approximately 200 video sequences at a resolution of 720 × 480 pixels, which are captured in natural environment with a wide range of scenes

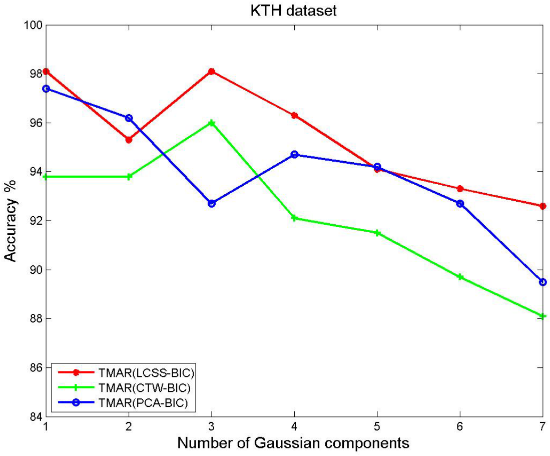


Figure 5: The recognition accuracy with respect to the number of Gaussian components for the KTH dataset.

Table 1: Recognition results over the KTH dataset.

Method	**Year**	**Accuracy (%)**
Schuldt *et al.* [9]	2004	71.7
Jhuang *et al.* [19]	2007	90.5
Fathi and Mori [13]	2008	90.5
Niebles *et al.* [20]	2008	83.3
Lin *et al.* [22]	2009	95.8
Seo and Milanfar [50]	2011	95.1
Wang *et al.* [14]	2011	94.2
Wu *et al.* [41]	2011	94.5
Le *et al.* [38]	2011	93.9
Yan and Luo [27]	2012	93.9
Sadanand and Corso [39]	2012	98.2
TMAR(LCSS-BIC)	2013	96.7
TMAR(CTW-BIC)	2013	93.8
TMAR(PCA-BIC)	2013	98.3

Table 2: P-values for measuring the statistical significance of the proposed methods for the KTH dataset.

Method	**TMAR(LCSS-BIC)**	**TMAR(CTW-BIC)**	**TMAR(PCA-BIC)**
Schuldt *et al.* [9]	4.6142×10^{-7}	0.0002	3.9887×10^{-8}
Jhuang *et al.* [19]	4.2002×10^{-4}	0.1368	1.3577×10^{-5}
Fathi and Mori [13]	4.2002×10^{-4}	0.1368	1.3577×10^{-5}
Niebles *et al.* [20]	1.0195×10^{-5}	0.0056	6.3654×10^{-7}
Lin *et al.* [22]	0.1847	0.7551	0.0015
Seo and Milanfar [50]	0.0660	0.6757	5.9672×10^{-4}
Wang *et al.* [14]	0.0181	0.5563	2.2285×10^{-4}
Wu *et al.* [41]	0.0274	0.5977	3.0348×10^{-4}
Le *et al.* [38]	0.0121	0.5141	1.6643×10^{-4}
Yan and Luo [27]	0.0121	0.5141	1.6643×10^{-4}
Sadanand and Corso [39]	0.9340	0.9189	0.9389

Table 3: Statistical measurements of the recognition results for each of the proposed approaches for the KTH dataset. Note that the values are expressed in percentages.

	mean	**median**	**std**	**min**	**max**
TMAR(LCSS-BIC)	96.65	96.80	2.11	92.70	98.90
TMAR(CTW-BIC)	93.80	95.60	6.58	84.70	100
TMAR(PCA-BIC)	98.90	99.45	1.41	96.60	100

and viewpoints. Figure 6 depicts some sample frames from the UCF Sports dataset.

Figure 6: Sample frames from video sequences of the UCF Sports dataset [10].

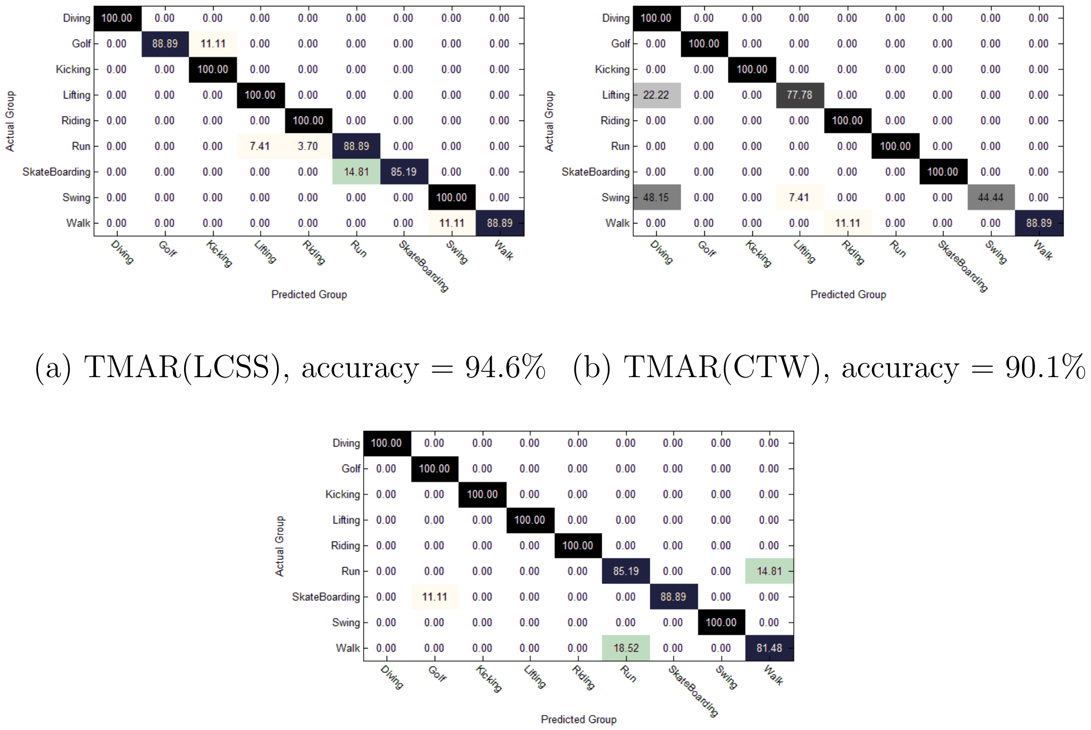
To test the proposed method on action recognition we also adopted the leave-one-out scheme. In Figure 7 are depicted the confusion matrices for the

Table 4: Recognition results over the UCF Sport dataset.

Method	**Year**	**Accuracy (%)**
Rodriguez *et al.* [10]	2008	69.2
Kovaska and Grauman [42]	2010	87.3
Wang *et al.* [14]	2011	88.2
Wu *et al.* [41]	2011	91.3
Le *et al.* [38]	2011	86.5
Yan and Luo [27]	2012	90.7
Sadanand and Corso [39]	2012	95.0
TMAR(LCSS-BIC)	2013	94.6
TMAR(CTW-BIC)	2013	90.1
TMAR(PCA-BIC)	2013	95.1

TMAR(LCSS), TMAR(CTW) and the TMAR(PCA) approaches. TMAR(LCSS) achieves 94.6% recognition accuracy with optimal number of components (BIC criterion) and 90.1% when the CTW alignment is employed. We also achieve the highest recognition accuracy of 95.1% when the proposed method uses PCA. In Figure 8, the dependency of the recognition accuracy with respect to the number of the Gaussian components is shown. Note that, for all three approaches as the number of components increases the recognition accuracy decreases, which may occur due to model overfitting. In the case where $K_j^p = 3$ all three approaches reach the highest peek of the graph. For $K_j^p \geq 4$ the recognition acuuracy begins to decrease. Table 4, shows the comparison between our TMAR approach, the baseline method using the BIC criterion in conjunction with ths LCSS metric and the CTW alignment, the proposed method with PCA and previous approaches on the UCF Sports dataset. The results of [10, 42, 14, 41, 38, 27, 39] are obtained from the original papers. As it can be observed, the TMAR(PCA) approach preforms better than all the other methods, while TMAR(LCSS) performs better for seven out of eight of the other methods. On the other hand, TMAR(CTW) has the less desirable performance as it outreaches four out of eight of the other methods on the same dataset.

Statistical evidence for the UCF Sports dataset is shown in Tables 5 and 6. TMAR(LCSS-BIC) and TMAR(PCA-BIC) appear to reject the null hypothesis for the majority of the cases, in contrary to the TMAR(CTW-



	Diving	Golf	Kicking	Lifting	Riding	Run	SkateBoarding	Swing	Walk
Diving	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golf	0.00	88.89	11.11	0.00	0.00	0.00	0.00	0.00	0.00
Kicking	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
Lifting	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
Riding	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
Run	0.00	0.00	0.00	7.41	3.70	88.89	0.00	0.00	0.00
SkateBoarding	0.00	0.00	0.00	0.00	0.00	14.81	85.19	0.00	0.00
Swing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
Walk	0.00	0.00	0.00	0.00	0.00	0.00	0.00	11.11	88.89

(a) TMAR(LCSS), accuracy = 94.6%

	Diving	Golf	Kicking	Lifting	Riding	Run	SkateBoarding	Swing	Walk
Diving	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golf	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kicking	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
Lifting	22.22	0.00	0.00	77.78	0.00	0.00	0.00	0.00	0.00
Riding	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
Run	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00
SkateBoarding	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00	0.00
Swing	48.15	0.00	0.00	7.41	0.00	0.00	0.00	44.44	0.00
Walk	0.00	0.00	0.00	0.00	11.11	0.00	0.00	0.00	88.89

(b) TMAR(CTW), accuracy = 90.1%

	Diving	Golf	Kicking	Lifting	Riding	Run	SkateBoarding	Swing	Walk
Diving	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Golf	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Kicking	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00	0.00
Lifting	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	0.00
Riding	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00
Run	0.00	0.00	0.00	0.00	0.00	85.19	0.00	0.00	14.81
SkateBoarding	0.00	11.11	0.00	0.00	0.00	0.00	88.89	0.00	0.00
Swing	0.00	0.00	0.00	0.00	0.00	0.00	0.00	100.00	0.00
Walk	0.00	0.00	0.00	0.00	0.00	18.52	0.00	0.00	81.48

(c) TMAR(PCA), accuracy = 95.1%

Figure 7: Confusion matrices of the classification results for the UCF Sports dataset for (a) the proposed method denoted by TMAR(LCSS), (b) the the proposed method using the CTW alignment, denoted by TMAR(CTW), and (c) the proposed method using PCA, denoted by TMAR(PCA), for the estimation of the number of components using the BIC criterion.

BIC) which reject the null hypothesis only for the Rodriguez *et al.* [10] method. TMAR(LCSS-BIC) and TMAR(PCA-BIC) seem to be statistical significant while TMAR(CTW-BIC) is not.

Finally, we have put our algorithm to test with the UCF YouTube dataset [11]. The UCF YouTube human action data set contains 11 action categories such as basketball shooting, biking, diving, golf swinging, horse riding, soccer juggling, swinging, tennis swinging, trampoline jumping, volleyball spiking, and walking with a dog. This data set includes actions with large variation in camera motion, object appearance and pose and scale. It also contains

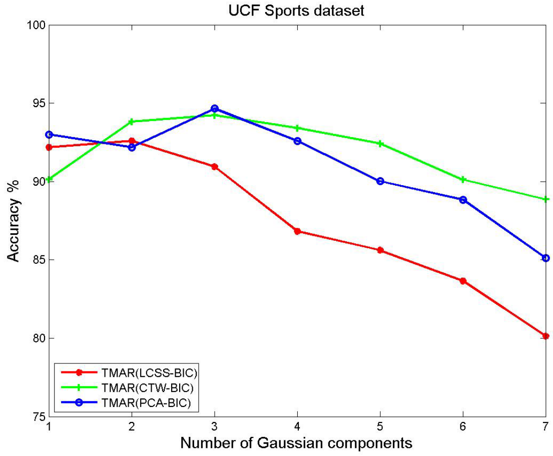


Figure 8: The recognition accuracy with respect to the number of Gaussian components for the UCF Sports dataset.

Table 5: P-values for measuring the statistical significance of the proposed methods for the UCF Sports dataset.

Method	**TMAR(LCSS-BIC)**	**TMAR(CTW-BIC)**	**TMAR(PCA-BIC)**
Rodriguez *et al.* [10]	1.2614×10^{-6}	0.0052	3.9515×10^{-6}
Kovaska and Grauman [42]	0.0048	0.3336	0.0083
Wang *et al.* [14]	0.0090	0.3848	0.0142
Wu *et al.* [41]	0.0822	0.5735	0.0913
Le *et al.* [38]	0.0028	0.2909	0.0052
Yan and Luo [27]	0.0541	0.5369	0.0644
Sadanand and Corso [39]	0.5691	0.7714	0.4950

Table 6: Statistical measurements of the recognition results for each of the proposed approaches for the UCF Sports dataset. Note that the values are expressed in percentages.

	mean	**median**	**std**	**min**	**max**
TMAR(LCSS-BIC)	94.61	100	6.49	85.10	100
TMAR(CTW-BIC)	90.10	100	18.81	44.40	100
TMAR(PCA-BIC)	95.03	100	7.68	81.40	100

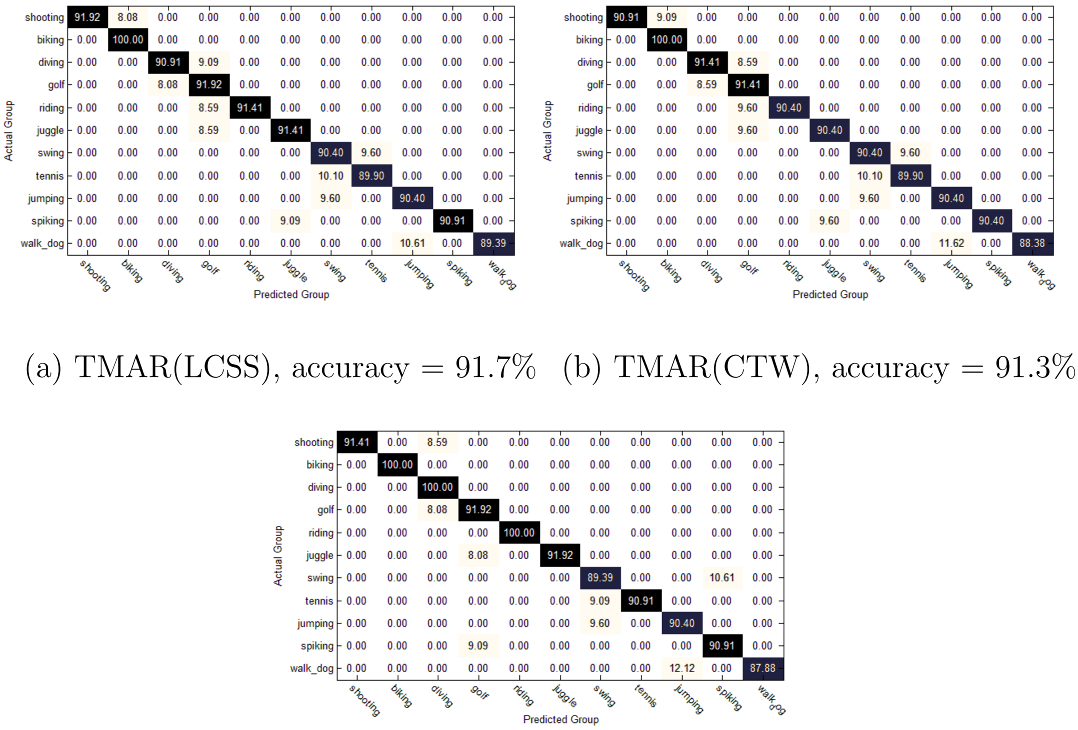
viewpoint and illumination changes, and spotty background. The video sequences are grouped into 25 groups of at least four actions each for each category, whereas the videos in the same group may share common characteristics such as similar background or actor. Representative frames of this data set are shown in Figure 9.

Figure 9: Sample frames from video sequences of the UCF YouTube action dataset [11].

In order to assess our method we have used the leave-one-out cross validation scheme. In Figure 10 the confusion matrices for the TMAR(LCSS), TMAR(CTW) and TMAR(PCA) approaches are shown. We achieve a recognition rate of 91.7% when the LCSS metric is employed and having estimated the Gaussian components using the BIC criterion. We also achieve 91.3% when the CTW alignment is employed and 93.2% when using PCA. In Table 7, comparisons with other state-of-the-art methods for this dataset are reported. The results of [11, 37, 38, 51] were copied from the original papers. As it can be seen, our algorithm achieves the highest recognition accuracy amongst all the others.

The performance of the proposed method with respect to the number of the Gaussian components is depicted in Figure 11. For TMAR(LCSS) the recognition accuracy begins to decrease for $K_j^p \geq 1$ and exhibits the worst performance than the other two approaches. The TMAR(CTW) approach decreases for $K_j^p \geq 2$ while TMAR(PCA) reaches its peak for $K_j^p = 4$ and then it begins to decrease. Note that, the best approach tends to be, attained by TMAR(PCA) which reaches a recognition accuracy of 91%.

Table 8 and Table 9 present the same indices for the UCF YouTube dataset. All three proposed methods reject the null hypothesis for all the cases. In this case, the recognition results of the proposed methods for the UCF YouTube dataset appear to be statistically significant.



(a) TMAR(LCSS), accuracy = 91.7%

(b) TMAR(CTW), accuracy = 91.3%

(c) TMAR(PCA), accuracy = 93.2%

Figure 10: Confusion matrices of the classification results for the UCF YouTube dataset for (a) the proposed method denoted by TMAR(LCSS), (b) the the proposed method using the CTW alignment, denoted by TMAR(CTW), and (c) the proposed method using PCA, denoted by TMAR(PCA), for the estimation of the number of components using the BIC criterion.

In the recognition step, in our implementation of the LCSS (7) the parameters δ and ϵ were optimized using 10-fold cross validation for all three datasets. These parameters need to be determined for each data set separately since each data set perform different types of actions. However, after we have determined the parameters no further action needs to be taken. To classify a new unknown sequence, we have already learned the parameters from the learning step and thus we are able to recognize the new action. For all the datasets, Table 10, Table 11 and 12 show the optimal values per action as they have resulted after the cross validation process. Note that,

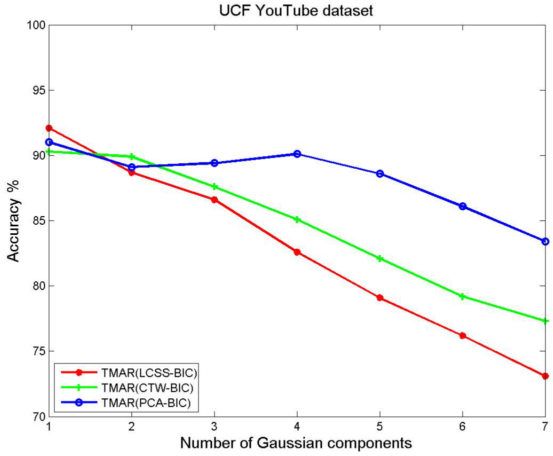


Figure 11: The recognition accuracy with respect to the number of Gaussian components for the UCF YouTube dataset.

Table 7: Recognition results over the UCF YouTube dataset.

Method	**Year**	**Accuracy (%)**
Liu *et al.* [11]	2009	71.2
Ikizler-Cinbis and Sclaroff [51]	2010	75.2
Le *et al.* [38]	2011	75.8
Wang *et al.* [37]	2011	84.2
TMAR(LCSS-BIC)	2013	91.7
TMAR(CTW-BIC)	2013	91.3
TMAR(PCA-BIC)	2013	93.2

the values in Table 10 for both δ and ε are consistently small. However, the handclapping and walking actions have larger values for ε parameter than the other actions, which may be due to the large vertical movement of the subject between consecutive frames. On the other hand, the actions in the UCF Sport dataset holds large movements from one frame to the other for both horizontal and vertical axes, which is the main reason why the actions show large variances between the values of δ and ε (Table 11). Finally, the actions in the UCF YouTube dataset have a uniform distributed representa-

Table 8: P-values for measuring the statistical significance of the proposed methods for the UCF YouTube dataset.

Method	**TMAR(LCSS-BIC)**	**TMAR(CTW-BIC)**	**TMAR(PCA-BIC)**
Liu *et al.* [11]	2.1189×10^{-10}	4.0791×10^{-10}	9.6806×10^{-9}
Ikizler-Cinbis and Sclaroff [51]	1.7831×10^{-9}	3.5804×10^{-9}	6.6558×10^{-8}
Le *et al.* [38]	2.5596×10^{-9}	5.1805×10^{-9}	9.1874×10^{-8}
Wang *et al.* [37]	3.0663×10^{-6}	7.5786×10^{-6}	3.4593×10^{-5}

Table 9: Statistical measurements of the recognition results for each of the proposed approaches for the UCF YouTube dataset. Note that the values are expressed in percentages.

	mean	**median**	**std**	**min**	**max**
TMAR(LCSS-BIC)	91.67	90.90	2.88	89.30	100
TMAR(CTW-BIC)	91.26	90.40	3.01	88.30	100
TMAR(PCA-BIC)	93.14	91.40	4.56	87.80	100

tion of the parameters δ and ε, since the parameter δ is determined as the 10% of the mean curves length for the most of the actions and the mean of the parameter ε is varies in the 15% of the standard deviation of the two curves to be compared.

Table 10: Parameters δ and ε for the KTH dataset estimated using cross validation.

Action	TMAR(LCSS)	
	$\delta(10^{-3})$	$\varepsilon(10^{-4})$
boxing	1	1
handclapping	10	100
handwaving	300	100000
jogging	5	3000
running	30	500
walking	1000	120000

The average percentage of matched curves for the TMAR(LCSS) and TMAR(CTW) approach in the case where the BIC criterion is employed

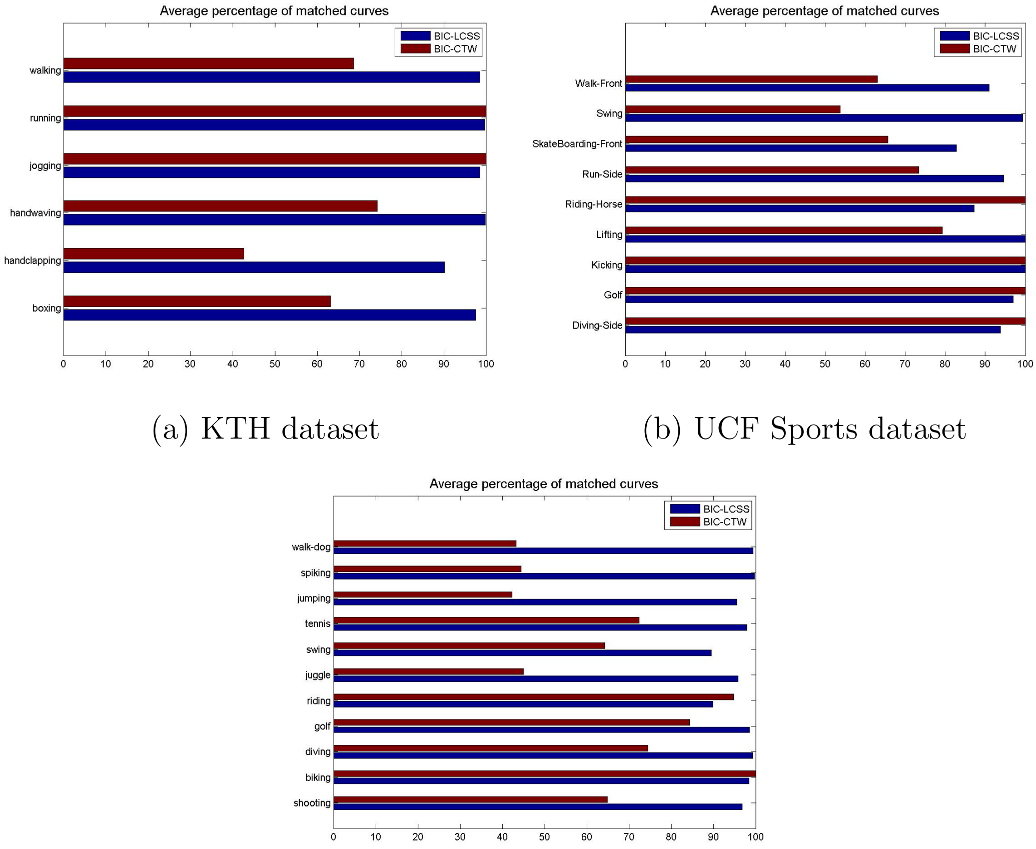
Table 11: Parameters δ and ε for the UCF Sports dataset estimated using cross validation.

Action	TMAR(LCSS)	
	δ	ε
diving	1	2.1
golf	2.01	6.1
kicking	10	15
lifting	11	10
riding	0.1	15
run	0.1	12
skateboarding	1.4	13
swing	0.6	20
walk	0.1	10

Table 12: Parameters δ and ε for the UCF Youtube dataset estimated using cross validation.

Action	TMAR(LCSS)	
	δ	ε
shooting	20	20
biking	10	10
diving	10	15
golf	20	10
riding	10	5
juggle	10	15
swing	10	5
tennis	10	10
jumping	10	5
spiking	10	30
walk dog	10	20

to determine the number of Gaussian components for all three datasets is depicted in Figure 12. As it can be observed, the TMAR(LCSS) method appears to match a larger part of curves for the same dataset than the TMAR(CTW) approach, which is the reason why TMAR(LCSS) performs better than TMAR(CTW).



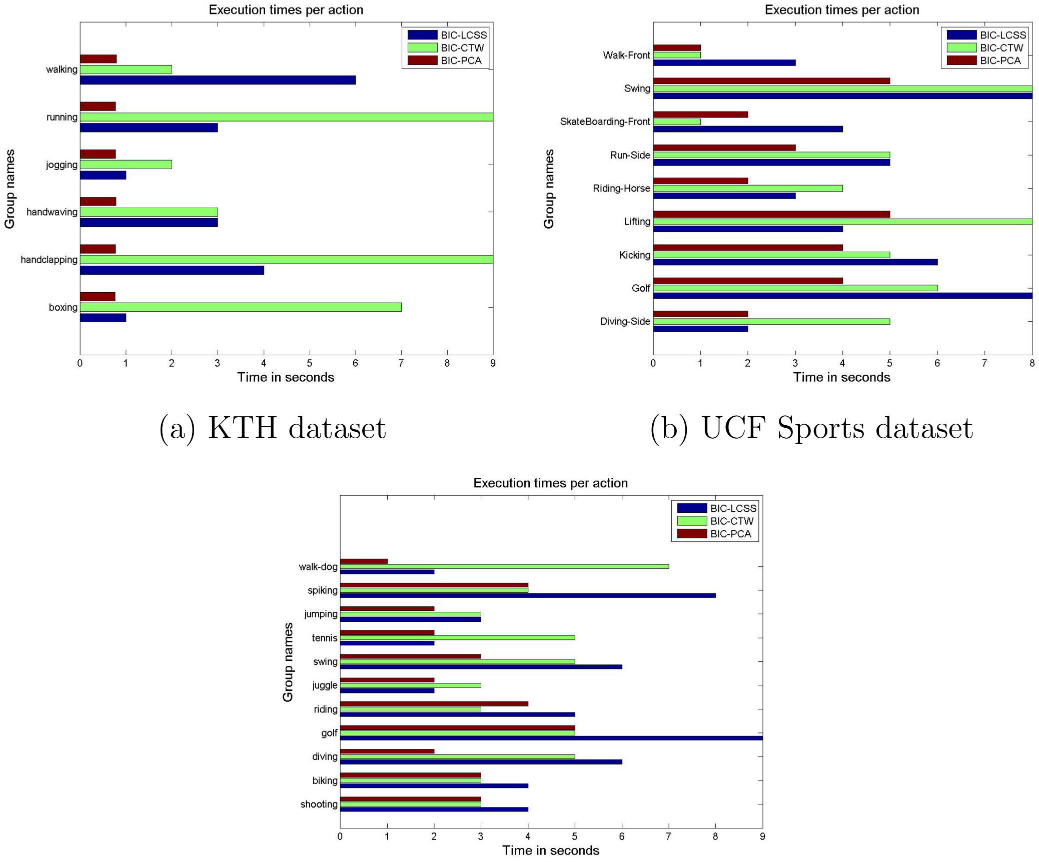
(a) KTH dataset

(b) UCF Sports dataset

(c) UCF YouTube dataset

Figure 12: Average percentage of matched curves for TMAR(LCSS) and TMAR(CTW), when the BIC criterion is used, for (a) KTH, (b) UCF Sports and (c) UCF YouTube datasets, respectively.

In Figure 13, the execution times using the BIC criterion are depicted in order to determine the number of the Gaussian components, for all three cases, when using the LCSS metric, the CTW alignment and PCA, for all three datasets. In the KTH dataset, TMAR(CTW) requires the highest execution time (needs nine seconds to recognize two out of six actions) and while TMAR(LCSS) takes less than six seconds for one action. Note that, the use of PCA speeds up the execution time for recognizing a single action in all datasets since feature vectors of smaller lengths are being used. However, in UCF Sport dataset TMAR(LCSS) and TMAR(CTW) both have the same upper bound of eight seconds to recognize an action. Finally, in UCF YouTube dataset, the average execution time to recognize an action ranges from two to nine seconds when TMAR(LCSS) approach is used. In the case where TMAR(PCA) is used the upper bound to recognize an action is five seconds in UCF Sports and UCF YouTube datasets, while in in KTH is less than a second. This makes the algorithm capable to adapt to any real video sequence and recognize an action really fast.



(a) KTH dataset

(b) UCF Sports dataset

(c) UCF YouTube dataset

Figure 13: Execution times per action in seconds for TMAR(LCSS), TMAR(CTW) and TMAR(PCA), when the BIC criterion is used, for (a) KTH, (b) UCF Sports and (c) UCF YouTube datasets, respectively.

5. Conclusion

In this paper, we presented an action recognition approach, where actions are represented by a set of motion curves generated by a probabilistic model. The performance of the extracted motion curves is interpreted by computing similarities between the motion curves, followed by a classification scheme. The large size of motion curves was reduced via PCA and after noise removal a reference database of feature vectors is obtained. Although a perfect recognition performance is accomplished with a fixed number of Gaussian mixtures, there are still some open issues in feature representation.

Our results classifying activities in three publicly available datasets show that the use of PCA has a significant impact on the performance of the recognition process, as its use leads to further improvement of the recognition accuracy, while it significantly speeds up the behavior of the proposed algorithm. The optimal model was determined by using the BIC criterion. Finally, our algorithm is free of any constraints in the curves lengths. Although the proposed method yielded encouraging results in standard action recognition datasets, it is requirement of a challenging task of performing motion detection, background subtraction, and action recognition in natural and cluttered environments.

References

[1] C. M. Bishop, Pattern Recognition and Machine Learning, Springer, 2006.

[2] M. Vlachos, D. Gunopoulos, G. Kollios, Discovering similar multidimensional trajectories, in: Proc. 18th International Conference on Data Engineering, San Jose, California, USA, 2002, pp. 673–682.

[3] F. Zhou, F. D. la Torre, Canonical time warping for alignment of human behavior, in: Advances in Neural Information Processing Systems Conference, 2009.

[4] J. B. Tenenbaum, V. D. Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, Science 290 (2000) 2319–2323.

[5] S. Theodoridis, K. Koutroumbas, Pattern Recognition, 4th Edition, Academic Press, 2008.

[6] M. Vrigkas, V. Karavasilis, C. Nikou, I. Kakadiaris, Action recogntion by matcing clustered trajectories of motion vectors, in: Proc. 8th International Conference on Computer Vision Theory and Application, Barcelona, Spain, 2013, pp. 112–117.

[7] P. Matikainen, M. Hebert, R. Sukthankar, Trajectons: Action recognition through the motion analysis of tracked features, in: Workshop on Video-Oriented Object and Event Classification, 2009, pp. 514–521.

[8] M. Raptis, I. Kokkinos, S. Soatto, Discovering discriminative action parts from mid-level video representations, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 1242–1249.

[9] C. Schuldt, I. Laptev, B. Caputo, Recognizing human actions: a local SVM approach, in: Proc. 17th International Conference on Pattern Recognition, Cambridge, UK, 2004, pp. 32–36.

[10] M. D. Rodriguez, J. Ahmed, M. Shah, Action MACH a spatio-temporal maximum average correlation height filter for action recognition., in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008, pp. 1–8.

[11] J. Liu, J. Luo, M. Shah, Recognizing realistic actions from videos "in the wild", Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2009) 1–8.

[12] A. A. Efros, A. C. Berg, G. Mori, J. Malik, Recognizing action at a distance, in: Proc. 9th IEEE International Conference on Computer Vision, Vol. 2, Nice, France, 2003, pp. 726–733.

[13] A. Fathi, G. Mori, Action recognition by learning mid-level motion features, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, 2008, pp. 1–8.

[14] Y. Wang, G. Mori, Hidden part models for human action recognition: probabilistic versus max margin, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (7) (2011) 1310–1323.

[15] V. Karavasilis, K. Blekas, C. Nikou, Motion segmentation by model-based clustering of incomplete trajectories, in: Proc. 2011 European

Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML-PKDD), 2011, pp. 146–161.

[16] V. Karavasilis, K. Blekas, C. Nikou, A novel framework for motion segmentation and tracking by clustering incomplete trajectories, Computer Vision and Image Understanding 116 (11) (2012) 1135–1148. doi:10.1016/j.cviu.2012.07.004.

[17] J. K. Aggarwal, M. S. Ryoo, Human activity analysis: a review, ACM Compututing Surveys 43 (3) (2011) 1–43.

[18] R. Poppe, A survey on vision-based human action recognition, Image and Vision Computing 28 (6) (2010) 976–990.

[19] H. Jhuang, T. Serre, L. Wolf, T. Poggio, A biologically inspired system for action recognition, in: Proc. IEEE International Conference on Computer Vision, Rio de Janeiro, Brazil, 2007, pp. 1–8.

[20] J. C. Niebles, H. Wang, L. Fei-Fei, Unsupervised learning of human action categories using spatial-temporal words, International Journal of Computer Vision 79 (3) (2008) 299–318.

[21] B. T. Morris, M. M. Trivedi, Trajectory learning for activity understanding: unsupervised, multilevel, and long-term adaptive approach, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (11) (2011) 2287–2301.

[22] Z. Lin, Z. Jiang, L. S. Davis, Recognizing actions by shape-motion prototype trees, in: Proc. IEEE International Conference on Computer Vision, Miami, Florida, USA, 2009, pp. 444–451.

[23] N. Dalal, B. Triggs, C. Schmid, Human detection using oriented histograms of flow and appearance, in: Proc. European Conference on Computer Vision, Graz, Austria, 2006, pp. 428–441.

[24] R. Chaudhry, A. Ravichandran, G. D. Hager, R. Vidal, Histograms of oriented optical flow and Binet-Cauchy kernels on nonlinear dynamical systems for the recognition of human actions, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Miami, Florida, USA, 2009, pp. 1932–1939.

[25] L. Gorelick, M. Blank, E. Shechtman, M. Irani, R. Basri, Actions as space-time shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 29 (12) (2007) 2247–2253.

[26] P. Dollár, V. Rabaud, G. Cottrell, S. Belongie, Behavior recognition via sparse spatio-temporal features, in: Proc. 14^{th} International Conference on Computer Communications and Networks, Beijing, China, 2005, pp. 65–72.

[27] X. Yan, Y. Luo, Recognizing human actions using a new descriptor based on spatial-temporal interest points and weighted-output classifier, Neurocomputing 87 (2012) 51–61.

[28] A. Yao, J. Gall, L. V. Gool, A Hough transform-based voting framework for action recognition, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 2010, pp. 2061–2068.

[29] K. Mikolajczyk, H. Uemura, Action recognition with motion-appearance vocabulary forest, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, 2008.

[30] K. Schindler, L. V. Gool, Action snippets: How many frames does human action recognition require?, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Anchorage, Alaska, USA, 2008, pp. 1–8.

[31] S. Khamis, V. I. Morariu, L. S. Davis, A flow model for joint action recognition and identity maintenance, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 1218–1225.

[32] N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2005, pp. 886–893.

[33] J. Wang, Z. Liu, Y. Wu, J. Yuan, Mining actionlet ensemble for action recognition with depth cameras, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 1290–1297.

[34] R. Li, T. Zickler, Discriminative virtual views for cross-view action recognition, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 2855–2862.

[35] B. Li, O. I. Camps, M. Sznaier, Cross-view activity recognition using hankelets, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 1362–1369.

[36] Q. Zhou, G. Wang, Atomic action features: A new feature for action recognition, in: Proc. European Conference on Computer Vision, Firenze, Italy, 2012, pp. 291–300.

[37] H. Wang, A. Kläser, C. Schmid, L. Cheng-Lin, Action recognition by dense trajectories, in: Proc. IEEE Conference on Computer Vision and Pattern Recognition, Colorado Springs, United States, 2011, pp. 3169–3176.

[38] Q. V. Le, W. Y. Zou, S. Y. Yeung, A. Y. Ng, Learning hierarchical invariant spatio-temporal features for action recognition with independent subspace analysis, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2011, pp. 3361–3368.

[39] S. Sadanand, J. J. Corso, Action bank: A high-level representation of activity in video, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Providence, Rhode Island, 2012, pp. 1234–1241.

[40] I. Laptev, On space-time interest points, International Journal of Computer Vision 64 (2-3) (2005) 107–123.

[41] X. Wu, D. Xu, L. Duan, J. Luo, Action recognition using context and appearance distribution features, in: Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, Washington, DC, USA, 2011, pp. 489–496.

[42] A. Kovashka, K. Grauman, Learning a hierarchy of discriminative space-time neighborhood features for human action recognition, in: Proc.

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, San Fransisco, CA, 2010, pp. 2046–2053.

[43] A. F. Bobick, J. W. Davis, The recognition of human movement using temporal templates, IEEE Transactions on Pattern Analysis and Machine Intelligence 23 (3) (2001) 257–267.

[44] M. Ahmad, S. W. Lee, Variable silhouette energy image representations for recognizing human actions, Image and Vision Computing 28 (5) (2010) 814–824.

[45] B. D. Lucas, T. Kanade, An iterative image registration technique with an application to stereo vision, in: Proc. 7$^{\text{th}}$ International Joint Conference on Artificial Intelligence, Nice, France, 1981, pp. 674–679.

[46] R. E. Kass, A. E. Raftery, Bayes factors, Journal of the American Statistical Association 90 (1995) 773–795.

[47] J. Kuha, AIC and BIC: Comparisons of assumptions and performance, Sociological Methods Research 33 (2) (2004) 188–229.

[48] I. T. Jolliffe, Principal Component Analysis, Springer Verlag, 1986.

[49] G. Sfikas, C. Constantinopoulos, A. Likas, N. P. Galatsanos, An analytic distance metric for gaussian mixture models with application in image retrieval, in: Proc of the 15$^{\text{th}}$ international conference on Artificial neural networks, Warsaw, Poland, 2005, pp. 835–840.

[50] H. J. Seo, P. Milanfar, Action recognition from one example, IEEE Transactions on Pattern Analysis and Machine Intelligence 33 (5) (2011) 867–882.

[51] N. Ikizler-Cinbis, S. Sclaroff, Object, scene and actions: combining multiple features for human action recognition, in: Proc 11$^{\text{th}}$ European conference on Computer vision: Part I, Heraklion, Crete, Greece, 2010, pp. 494–507.