Temporal Feature Weighting for Prototype-Based Action Recognition*

Thomas Mauthner, Peter M. Roth, and Horst Bischof

Institute for Computer Graphics and Vision
Graz University of Technology
{mauthner,pmroth,bischof}@icg.tugraz.at

Abstract. In action recognition recently prototype-based classification methods became popular. However, such methods, even showing competitive classification results, are often limited due to too simple and thus insufficient representations and require a long-term analysis. To compensate these problems we propose to use more sophisticated features and an efficient prototype-based representation allowing for a single-frame evaluation. In particular, we apply four feature cues in parallel (two for appearance and two for motion) and apply a hierarchical k-means tree, where the obtained leaf nodes represent the prototypes. In addition, to increase the classification power, we introduce a temporal weighting scheme for the different information cues. Thus, in contrast to existing methods, which typically use global weighting strategies (i.e., the same weights are applied for all data) the weights are estimated separately for a specific point in time. We demonstrate our approach on standard benchmark datasets showing excellent classification results. In particular, we give a detailed study on the applied features, the hierarchical tree representation, and the influence of temporal weighting as well as a competitive comparison to existing state-of-the-art methods.

1 Introduction

Recently, human action recognition has been of growing interest in Computer Vision, where typical applications include visual surveillance, human computer interaction, or monitoring systems for elderly people. Thus, a variety of approaches have been proposed introducing new features, representations, or classification methods. Since actions can be described as chronological sequences, special attention has been paid to how to incorporate temporal information. In general, this can be realized either by keeping a very strict spatial-temporal relation on the features (e.g., by spatio-temporal volumes [1, 2] or descriptors [3–5]) or on the representation level [6, 7]. For such approaches the classification is typically performed on single-frame basis and the analysis of longer sequences is based on a simply majority voting or on averaging over multiple frames. If

* This work was supported by the Austrian Science Fund (FWF) under the project MASA (P22299) and the Austrian Research Promotion Agency (FFG) under the project SECRET (821690) and the Austrian Security Research Programme KIRAS.

spatial-temporal consistency is totally ignored [8] only whole sequences can be analyzed. One effective way for directly describing temporal information, that showed great success in the past, is the usage of prototypes, e.g., [6–9].

In general, prototype-based learning methods can be described by a prototype space $\mathbf{X} = \{x_1, \ldots, x_n\}$, which is defined as a set of representative samples x_j describing the data (prototypes), and a distance function ρ [9]. In particular, for action recognition the data is split into a smaller set of reference templates referred to as prototypes [7], key-poses [8], or pose-primitives [6].

Weiland and Boyer [8] used foreground segmentations to create a set of silhouette exemplars, so called key-poses, using a forward selection process. The final action description is achieved by comparing the occurrence frequency of key-poses in a video. Although they presented excellent results, they completely neglected the temporal ordering of prototypes within a sequence and showed only recognition results on complete videos. Similarly, Elgammal et. al. [9] modeled an action as a sequence of silhouette prototypes using an HMM for incorporating temporal constrains and being more robust to small deviations. To incorporate temporal context in a prototype-based representation, Thurau and Hlavac [6] introduced n-grams models. They define sub-sequences of n frames and describe the transitions between prototypes by n-dimensional histograms. However, the required number of samples to fill the n-dimensional histograms is high and the temporal ordering is very strict. Furthermore, the representation of n-grams is getting difficult if $n > 3$. Experimentally, they showed state-of-the-art results on sequences with lengths of around 30 frames. Since shape information clearly gives only limited information on single-frame basis Lin et al. [7] recently proposed to create prototypes in a joined shape-motion space using binary foreground silhouettes for shape and flow features as introduced in [10]. The prototypes are trained and represented efficiently using hierarchical k-means, leading to real-time evaluation performance. Finally, the temporal information is incorporated using Dynamic Time Warping (DTW). Although DTW is a powerful method for aligning temporal sequences, as a drawback it only compares one sequence to another and cannot handle transitions between different actions. Again only results on sequence level are shown.

Even though showing competitive recognition results, existing prototype-based action recognition methods are limited due to a required long-term analysis (i.e., on a whole sequence) and mainly rely on accurate segmentations – at least for learning the shape prototypes (e.g., [6, 8, 7]). In praxis, however, perfect segmentations are often not available and short and fast actions such as in sports analysis should be recognized. Hence, the goal for human action recognition should be to robustly classify on a short sequence length. Schindler and van Gool [5] showed that if a more sophisticated representation is used human action recognition can also be performed from very short sequences (snippets). They use motion and appearance information in parallel, where both are processed in similar pipelines using scale and orientation filters. The thus obtained features are then learned by using a Support Vector Machine. Their approach showed

impressive results, reaching state-of-the-art results even though only short sequences of 5-7 frames are used.

Hence, the goal of this paper is to introduce an efficient action recognition approach working on short-frame level that takes advantage of prototype-based representations such as fast evaluation, multi-class capability, and sequential information gain. In particular, we propose to use four information cues in parallel, two for appearance and two for motion, and to independently train a hierarchical k-means tree [11] for each of these cues. To further increase the classification power, we introduce a temporal weighting scheme from temporal co-occurrences of prototypes. Hence, in contrast to existing methods that are using different cues (e.g., [5, 12]) we do not estimate global weights, which allows us to temporally adapt the importance of the used feature cues. Moreover, even using temporal context we can still run a frame-wise classification! The benefits of the proposed approach are demonstrated on standard benchmark datasets, where competitive results on short frame basis as well as when analyzing longer sequences are shown.

The reminder of the paper is organized as follows. First, in Section 2 we introduce our new action recognition approach consisting of an efficient prototype-based representation and a temporal feature weighting scheme. Next, in Section 3, we give a detailed analysis of our method and compare it to existing approaches on standard datasets. Finally, we summarize and conclude the paper in Section 4.

2 Temporal Action Learning

In the following, we introduce our new prototype-based action recognition approach which is illustrated in Fig. 1. To gain different kind of information, we apply four feature cues in parallel, two for appearance and two for motion (Section 2.1). For these cues we independently train hierarchical k-means trees [11], which provide several benefits such as very efficient frame-to-prototype matching and an inherent multi-class classification capability (Section 2.2). To incorporate temporal information, we further estimate temporal weights for the different feature cues. In particular, from temporal co-occurrences of prototypes we learn a temporal reliability measure providing an adaptive weight prior for the evaluation step (Section 2.3). In this way during evaluation at a specific point in time the most valuable representations get higher temporal weights increasing the overall classification power (Section 2.4).

2.1 Features

In contrast to existing prototype-based action recognition methods, which mainly use segmentation results to represent the data, we apply four more sophisticated feature cues in parallel, two describing the appearance and two describing the motion, respectively. In particular, for appearance these are the Histogram of

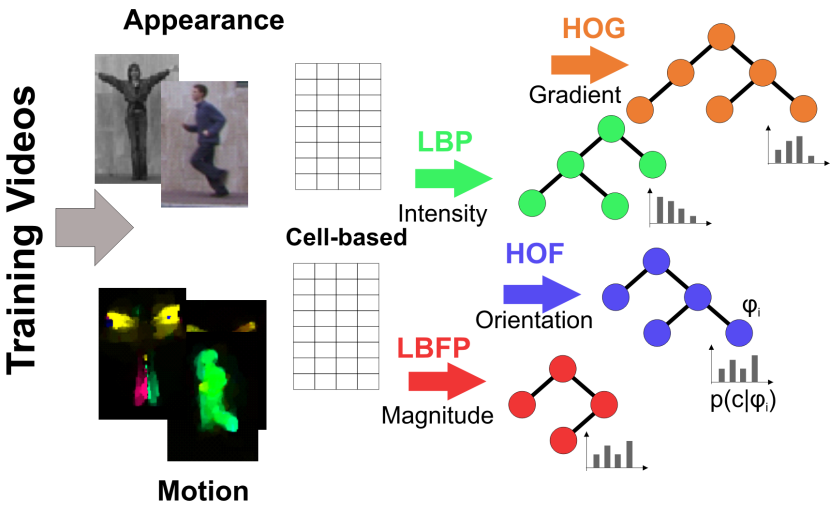


Fig. 1. Prototype-based action recognition: For each feature cue f a hierarchical k-means tree T^f is estimated, where the leaf nodes of T^f are treated as prototypes φ. In addition, to allow for a prototype-based classification, for each prototype φ the probabilities $p(c|\varphi)$ are estimated, where c is the corresponding action class.

Gradients (HOG) descriptor [13] and the Locally Binary Patterns (LBP) descriptor [14]. HOG estimates a robust local shape description using a histogram binning over the gradient orientation a local normalization whereas LBPs, originally introduced for texture description, are valuable due to invariance to monotonic gray level changes and robustness to noise. Thus, both have shown to be very valuable for human detection as well as for action recognition. To describe the motion information, we adapted both methods to describe a motion field obtained from an optical flow estimation: Histogram of Flow (HOF) and Locally Binary Flow Patterns (LBFP). In the following, we give the details on these descriptors given the image $\mathbf{I_t} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ at time t.

HOG As first step in the HOG calculation, we have to estimate the gradients $g_x(x, y)$ and $g_y(x, y)$. For each position (x, y) the image $\mathbf{I_t}$ is filtered by 1-dimensional masks $[-1, 0, 1]$ in x and y direction [13]. Then, we calculate the magnitude $m(x, y)$ and the signed orientation $\Theta_S(x, y)$:

$$m(x, y) = \sqrt{g_x(x, y)^2 + g_y(x, y)^2} \tag{1}$$

$$\Theta_S(x, y) = \tan^{-1}\left(g_y(x, y)/g_x(x, y)\right) \ . \tag{2}$$

To avoid problems due to intensity changes and to make the descriptor more robust, we transform the signed orientation Θ_S into an unsigned orientation:

$$\Theta_U(x,y) = \begin{cases} \Theta_S(x,y) + \pi & \theta_S(x,y) < 0 \\ \Theta_S(x,y) & \text{otherwise} \end{cases} . \tag{3}$$

To estimate the HOG descriptor, we divide the image $\mathbf{I_t}$ into non-overlapping cells of size 10×10. For each cell, the orientations Θ_U are quantized into 9 bins and weighted by their magnitude m. Groups of 2×2 cells are combined in overlapping blocks and the histogram of each cell is normalized using the L2-norm of the block. The final descriptor is built by concatenation of all normalized blocks. For speed issues we avoid the tri-linear interpolation.

HOF In contrast to HOGs, which are estimated from only one frame, for estimating HOFs the motion has to be estimated from two frames. In particular, to estimate the optical dense flow field, we apply the method proposed in [15], which is publicly available via *OFLib*[1]. In fact, the GPU-based implementation allows a real-time computation of the features. Given $\mathbf{I_t}, \mathbf{I_{t-1}}$, the optical flow describes the shift from frame $t-1$ to t with the disparity $\mathbf{D_t}$, where $d_x(x,y)$ and $d_y(x,y)$ denote the disparity components in x and y direction at location (x,y). We then compute the HOF descriptor similar as described above by applying Eqs. (2) and (3). However, the gradients $g_y(x,y)$ and $g_x(x,y)$ are replaced by the disparity components $d_y(x,y)$ and $d_x(x,y)$. Moreover, to capture different motion directions for same poses, we use the signed orientation Θ_S and quantize the orientation into 8 bins. The other parameters such as cell/block combination are the same as described above.

LBP An LBP pattern p is constructed by binarization of intensity differences between a center pixel and a number of n sampling points with radius r. The pattern p is assigned 1 if the intensity of a sampling point has a higher intensity than the center pixel and 0 otherwise. The final pattern is formed by the $0-1$ transitions of the sampling points in a given rotation order. To avoid ambiguities due to rotation and noise we restrict the number of allowed $0-1$ transitions to a maximum u, hence, defining uniform patterns $LBP^u_{n,r}$. For our final description we build $LBP^4_{8,1}$ pattern histograms for each cell and sum up the nonuniform patterns to one bin (see [14] for more details). To finally estimate the LBP descriptors, similar to [16], we keep the cell-based splitting of the HOGs and extract pattern histograms as described before for each cell.

LBFP Motivated by the relation between HOG and HOF we directly apply LBPs on optical flow as well. Integrating direction information into LBP descriptors is quite difficult since due to noise in the dense optical flow field the orientation information is sometimes misleading. However, LBPs are known to be robust against such clutter, also appearing in texture, and therefore they are a considerable choice for an additional complementary flow feature. In particular, we keep the cell structure of the appearance LBP and compute the LBFP histograms on the flow magnitude $m(x,y)$, which is computed using Eq. (1) from

[1] http://gpu4vision.icg.tugraz.at/

$d_y(x,y)$, $d_x(x,y)$. Although the description is slightly simpler compared to HOF, it is more robust in presence of noise. In general, the same parametrization as for LBP is used. Please note that LBFP are not related to Local Trinity Patterns [17], which are computed on video volumes.

2.2 Learning a Prototype-based Representation

Having the feature descriptions discussed in Section 2.1 with respect to a prototype-based representation two main issues have to be considered. First, how to select a representative set of prototypes. Second, if the number of prototypes is increasing, simple nearest neighbor matching gets infeasible and a more efficient method is required. In particular, we solve both problems by applying a hierarchical k-means clustering, which is also known as Vocabulary Tree [11].

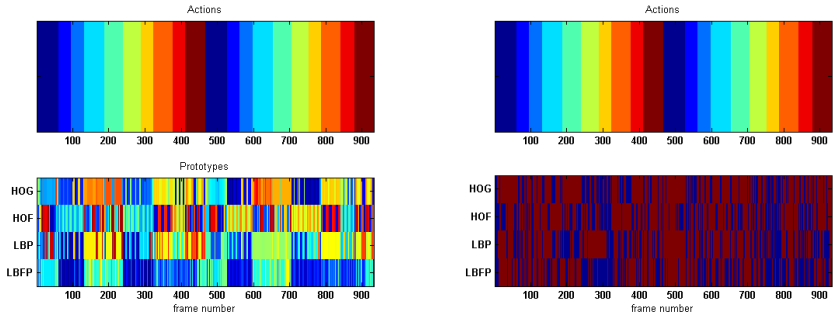
Given the training set S, we first perform a k-means clustering on all training samples s. According to the cluster indices, the data S is then split into subsets (branches of the vocabulary tree), and each subset is clustered again using k-means clustering. This process is repeated recursively until no samples are left in a branch of the tree or if the maximum depth is reached. The thus obtained leaf nodes of the tree are then treated as prototypes φ. Hence, only two parameters are required: the split number k and a maximum hierarchy depth L, allowing to generate a maximum number of k^L prototypes. During evaluation, a test sample is matched to a prototype by traversing down the tree, using depth-first-search, until it reaches a leaf node.

As illustrated in Fig. 1, we independently build a vocabulary tree T^f for each feature cue f. Thus, for each cue f we obtain prototypes φ_j^f, i.e., the leaf nodes of tree T^f, from which we can build the prototype sets $\Phi^f = \left\{\varphi_1^f, \ldots, \varphi_N^f\right\}$. To enable a multi-class classification (i.e., one class per action), we have to estimate the class probability distribution $p(c|\varphi)$ for all prototypes φ. Let $S_c \subset S$ be the set of all samples belonging to class c and $S_{\varphi,c} \subset S_c$ be the set of all samples belonging to class c matching the prototype φ. Then the probability that a sample s matching the prototype φ belongs to class c can be estimated as $p(c|\varphi) = \frac{|S_{\varphi,c}|}{|S_c|}$. If no samples from class c reached the prototype φ, i.e., $|S_{\varphi,c}| = 0$, the probability is set to $p(c|\varphi) = 0$.

Illustrative thus obtained classification results are shown in Fig. 2. The first row gives a color-coded view of different actions whereas in the second row (a) the corresponding prototypes and (b) the correct classifications are visualized. It can be seen that for correct classifications over time different prototypes are matched, leading to representative prototype sequences. This clearly shows that the variety in the data can be handled well by using our prototype-based description.

2.3 Learning Temporal Weights

However, from Fig. 2(b) it also can be recognized that the classification results for the single cues are very weak. Hence, the goal would be to combine these results to improve the classification results. The naive approach to fuse



(a) prototype sequences (b) classification

Fig. 2. Single cue prototype-based classification: (a) sequences (color-coded prototype numbers) of matched prototypes for each feature cue and (b) classification results, where red indicates a correct classification (second row). The actions (first row) are color-coded in the range $1-10$, respectively.

the results from different information cues would be to use majority voting or to estimate a mean over all decisions. Such approaches, however, totally neglect the information given by temporal constraints. Thus, in the following we introduce a more sophisticated information fusion strategy based on temporal weighting. The main idea is to exploit the knowledge which cues provided reliable results during training to assign temporal adaptive weights to the feature cues during evaluation.

Given the prototype sets Φ^f the key idea is to estimate the reliability of a feature cue for the prototype transitions $\varphi_i^f \rightarrow \varphi_j^f$. This is similar to prototype frequencies or transition statistics as used in [8, 6], which, however, require long sequences to get sufficient data to estimate the discriminative votes. Instead, we consider these transitions only in a short time frame introducing temporal bags, which is illustrated in Fig. 3(a).

A temporal bag[2] $b_{i,m}^t$ is defined as set of m prototypes φ_j, which followed the prototype φ_i at time t: $b_{i,m}^t = \left\{\varphi^{t+1}, \ldots, \varphi^{t+m}\right\}$. Once all bags $b_{i,m}^t$ were estimated (i.e., for each occurrence of φ_i) these are combined to a global bag $B_i = \left\{b_{i,m}^1, \ldots, b_{i,m}^T\right\}$, where T is the number of temporal bags $b_{i,m}^t$. Then from B_i we can estimate the temporal co-occurrences of φ_i and φ_j. In particular, we calculate a co-occurrence matrix $\mathbf{C}$, where $c_{i,j}$ integrates all cases within B_i where a prototype φ_i was followed by φ_j: $c_{i,j} = \sum_{t=1}^{T} |\varphi_j \in b_{i,m}^t|$. Having estimated the co-occurrence matrix $\mathbf{C}$, we now can compute a temporal reliability measure $w_{i,j}$. Let $n_{i,j}$ be the number of samples that were classified correctly by prototype $\varphi_j \in B_i$, then we set the reliability weight to $w_{i,j} = \frac{n_{i,j}}{c_{i,j}}$.

This is illustrated in Fig. 3(a). The bag B_i contains 7 instances of φ_h and 8 instances of φ_j. While prototype φ_j classified all 8 frames correctly, φ_h provided

[2] Since these calculations are performed for each cue f, in the following for reasons of readability we skip the superfix f in the notation.

the correct class for only two samples. Thus, yielding reliability weights of $w_{i,h} = 2/7$ and $w_{i,j} = 1$. If this procedure is repeated for all prototypes in all feature cues this finally yields to four co-occurrence matrices $\mathbf{C}^f$ and four reliability matrices $\mathbf{W}^f$, which can then be used during the test stage as illustrated in Fig. 3(b).

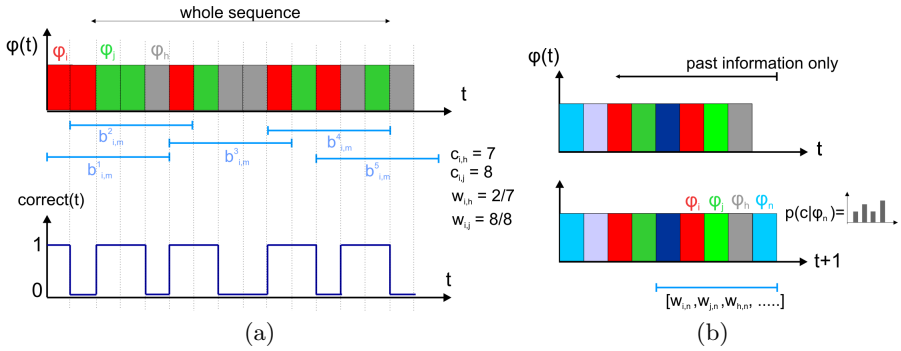


Fig. 3. Temporal weighting for feature cues: (a) during training the weights $w_{i,j}$ for temporal co-occurrences of prototypes φ_i and φ_j of a feature cue are estimated; (b) during evaluation these weights are used to temporally change the importance of that features cue.

2.4 Recognition Using Temporal Weights

Once we have estimated the hierarchical trees T^f and the prototype reliability matrices $\mathbf{W}^f$ as introduced in Sections 2.2 and 2.3, we can perform action recognition using the following classification problem:

$$p(c|t) = \sum_{f=1}^{4} w_t^f \, p\left(c|\varphi_t^f\right) \ , \tag{4}$$

where w_t^f is the weight of the feature cue f and φ_t^f the identified prototype for cue f at time t. The crucial step now is to estimate the weights w_t^f, which is illustrated in Fig. 3(b).

For that purpose, we use the information given by the past, i.e., the identified prototypes per cue, to estimate temporal weights. In particular, considering a temporal bag of size m we estimate the prototype transitions $\varphi_i^f \rightarrow \varphi_j^f$, where $i = t-m, \ldots, t-1$ and $j = t$. Based on these selections using the reliability matrices $\mathbf{W}^f$ we can estimate the m corresponding weights $w_{i,j}$. Finally, the weight w_t^t is estimated by averaging the m weights $w_{i,j}$ over the temporal bag.

This recognition process is demonstrated in Fig. 4, where the first row illustrates three actions, the second row the identified prototypes, and the last row

the corresponding weights. It clearly can be seen that the same action is characterized by different prototypes and also that the weights are changing over time.

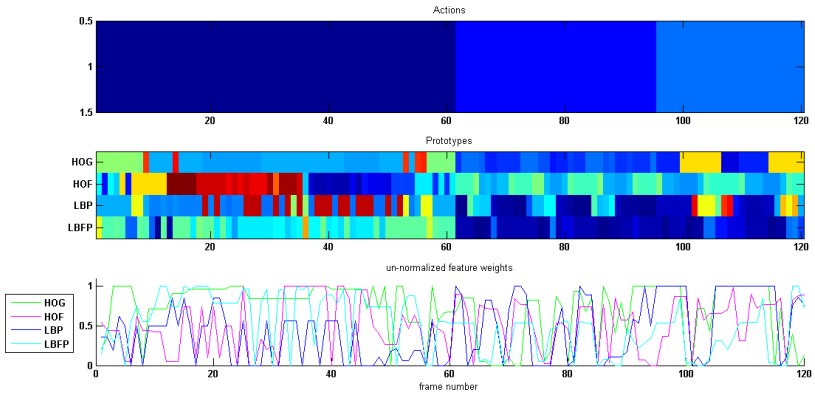


Fig. 4. On-line adapted feature weights obtained from our temporal reliability measure: color-coded actions (first row), matched prototypes of each feature cue (second row), and estimated weights (third row).

3 Experimental Results

In the following, we demonstrate our prototype-based action recognition approach, where we run several experiments on publicly available action recognition benchmark data sets, i.e., Weizmann and KTH. We first give a detailed analysis of prototype-based action recognition and show that temporal information can be useful to improve the classification results. Then, we give a detailed comparison to recently published state-of-the-art action recognition approaches. In both cases, the given results were obtained by a leave-one-out cross-evaluation [18, 19] (i.e., we used all but one individuals for training and evaluated the learned model for the missing one).

3.1 Benchmark Datasets

Weizmann Dataset The Weizmann human action dataset [1] is a publicly available dataset, that originally contains 81 low resolution videos (180×144) of nine subjects performing nine different actions: running, jumping in place, bending, waving with one hand, jumping jack, jumping sideways, jumping forward, walking, and waving with two hands. Subsequently, a tenth action, jumping on one leg, was added [20]. Illustrative examples for each of these actions are shown in Figure 5.

Fig. 5. Examples from the Weizmann human action data set.

KTH Dataset The KTH human action dataset, originally created by [3], consists of 600 videos (160×120), with 25 persons performing six human action in four different scenarios: outdoors (*s1*), outdoors with scale variation (*s2*), outdoors with different clothes (*s3*), and indoors (*s4*). Illustrative examples for each of these actions are shown in Figure 6.

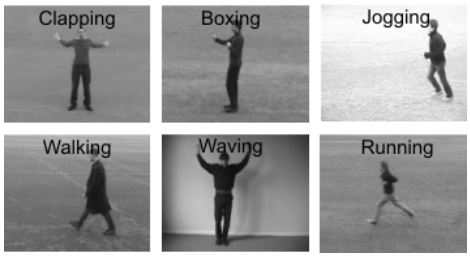
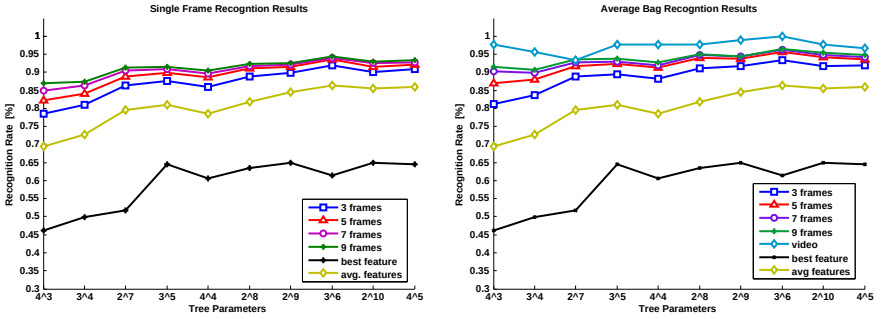


Fig. 6. Examples from the KTH action data set.

3.2 Analysis of Prototype-based Learning

First of all, we give a detailed analysis of our proposed prototype-based action recognition approach, where we analyze the influence of the parameters of the hierarchical k-means tree and the bag size for the temporal weighting. For that purpose, we run several experiments varying these parameters on the Weizmann data set. The corresponding results are given in Fig. 7. Three main trends can be recognized. First, increasing the temporal bag size, which was varied between 3 and 9 increases the classification accuracy. However, using a bag size greater than 5 has only little influence on the classification performance. Second, increasing the number of prototypes (using different tree parameters, i.e., split criteria and depth) increases the classification power. However, if the number of possible prototypes gets too large, i.e., too many leaf nodes are weakly populated, the classification power is decreased - the optimum number is around 2^8. Third, it

can be seen that using the proposed weighting scheme the single cue classification results as well as a naive combination can clearly be outperformed. In addition, Fig. 7(b) shows that averaging the single cue classification results over the temporal bags almost reaches the classification result if the whole sequences are analyzed.



(a) (b)

Fig. 7. Classification results on the Weizmann data set with different numbers of prototypes by varying parameters for hierarchical k-means tree and temporal bags: (a) single frame results and (b) bag averaged results.

Next, we compare different evaluation strategies in detail for the Weizmann as well as on the KTH data set: (a) single frame evaluation, (b) averaging the single frame results over temporal bags, (c) analyzing the whole sequence (using a majority voting). In addition, we show results (on single frame basis) without using the temporal weighting: (d) analyzing the best single feature cue and (e) a naive feature combination (majority voting). Based on the results in Fig. 7 for the remaining experiments we set the bag size to 5 and used a binary split criterion. The thus obtained results are summarized in Table 1 for the Weizmann data set and in Table 2 for the KTH data set.

	single frame	bag avg.	all video	best feature	comb. features	#proto.
2^8	92.4%	94.5%	97.8%	60.1%	84.6%	122
2^{12}	92.4%	94.2%	100.0%	70.2%	89.9%	614

Table 1. Overview of recognition results on the Weizmann-10 data set using 2-means clustering on a maximal depth of 8 and 12, respectively.

From Table 1 the benefits of the proposed method can clearly be seen. In fact, considering a tree-size of 2^8 the best single feature cue provides a classification result of approximative 60%. If the four cues are naively combined the overall classification result can improved to 85%. However, using the proposed temporal

weighting (using a bag size of 5) an improvement of the classification-rate of 7%, by further averaging over the bag even of 9% can be obtained. If the whole sequence is analyzed, we finally get a correct classification rate of 98%, which can further be improved to 100% if the tree depth is increased.

The same trend can be recognized for the KTH data set in Table 2, where we split the results for the four sub-sets. In particular, there is a significant improvement using the proposed method compared to the best single feature cue and the naive combination. However, as can be seen, the single frame classification results can be improved only a little by averaging over the whole bag. However, if the whole sequences are analyzed still a considerable improvement can be recognized.

	single frame	bag avg.	all video	best feature	comb. features	#proto.
s1	92.7%	92.7%	97.3%	69.4%	90.1%	113
s2	89.1%	90.6%	94.7%	59.0%	82.0%	118
s3	93.4%	94.5%	98.7%	71.0%	88.0%	116
s4	91.6%	91.7%	98.7%	63.0%	91.9%	109

Table 2. Overview of recognition results on KTH data set using 2-means clustering on a maximal depth of 8.

3.3 Comparison to State-of-the-Art

Next, we give a comparative study of our approach compared to state-of-the-art action recognition methods on the Weizmann and the KTH data set. Since different authors used different versions of the Weizmann data set, i.e., 9 vs. 10 actions, we split the Weizmann experiment into two parts. In particular, we compared our approach to Schindler & van Gool [5] and to Thurau & Hlaváč [6], which are the most similar methods to ours - also providing an analysis on short frame basis - and to recent methods reported the highest classification results. The thus obtained results are given in Tables 3–4. The best classification results when analyzing the whole sequence are set boldface, respectively.

From Table 3 it can be seen that we obtain competitive results on short frame basis as well as when analyzing the whole sequence. In fact, it can be seen that we obtain comparable results to Schindler & v. Gool and that we clearly can outperform the approach of Thurau & Hlaváč on short frame basis. Moreover, when analyzing the whole sequence for both data sets we obtain classification results of 100%. Finally, we carried out the same experiments on the KTH data set showing the results in Table 4. Again, it can be seen that we obtain competitive results on short frame basis as well as when analyzing the whole sequence, even (significantly) outperforming most state-of-the-art methods an all four data sets. In particular, also on this data set we outperform the approach of Schindler & van Gool on short frame basis and yield the best overall performance for the full sequence analysis!

method	rec.-rate	#frames
proposed	94.9%	1/5
	100.0%	all
Schindler [5]	93.5%	1/2
& v. Gool	96.6%	3/3
	99.6%	10/10
Blank et al. [21]	99.6%	all
Jhuang et al. [22]	98.8%	all

(a) Weizmann-09

method	rec.-rate	#frames
proposed	92.4%	1/5
	94.2%	5/5
	100.0%	all
Thurau	70.4%	1
& Hlaváč [6]	94.4%	30/30
Gorelick et al. [20]	98.0%	all
Lin et al. [7]	**100.0%**	all
Fathi & Mori [23]	**100.0%**	all

(b) Weizmann-10

Table 3. Recognition rates and number of frames used for different approaches reported for the *Weizmann* data set. The best results are shown in bold-face, respectively.

method	s1	s2	s3	s4	average	# frames
proposed	92.7%	89.1%	86.1%	91.3%	89.8%	1/5
	92.6%	90.6%	94.5%	91.7%	92.4%	5/5
	97.3%	**94.7%**	**98.7%**	**98.7%**	**97.4%**	all
Schindler & v. Gool [5]	90.9%	78.1%	88.5%	92.2%	87.4%	1/2
	93.0%	81.1%	92.1%	96.7%	90.2%	7/7
Lin et al. [7]	**98.8%**	94.0%	94.8%	95.5%	95.8%	all (NN)
	97.5%	86.2%	91.1%	90.3%	91.3%	all (proto.)
Yao and Zhu[24]	90.1%	84.5%	86.1%	91.3%	88.0%	all
Jhuang et al. [22]	96.0%	87.2%	91.7%	95.7%	92.7%	all

Table 4. Recognition rates and number of required frames for different approaches reported for the *KTH* data set. The best results are shown in bold-face, respectively.

4 Conclusion

In this paper, we addressed the problem of weighting different feature cues for action recognition. Existing approaches are typically limited due to a fixed weighting scheme, where most of the benefits get lost. In contrast, we propose a temporal weighting, where the weights for different features cues can change over time, depending on the current input data. In particular, we use a prototype-based representation, which has previously shown to provide excellent classification results. However, in contrast to existing methods using simple segmentation we apply more sophisticated features to represent the data. We compute HOG and LBP descriptors to represent the appearance and HOF and LBFP descriptors for representing motion information. For each of these feature cues we estimated a prototype-based representation by using a hierarchical k-means tree allowing for very efficient evaluation. These cues are evaluated using temporal weights showing an increasing performance, which was demonstrated on standard benchmark datasets, i.e., Weizmann and KTH.

References

1. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proc. ICCV. (2005)
2. Ke, Y., Sukthankar, R., Hebert1, M.: Event detection in crowded videos. In: Proc. ICCV. (2007)
3. Schueldt, C., Laptev, I., Caputo, B.: Recognizing human actions: A local svm approach. In: Proc. ICPR. (2004)
4. Dollar, P., Rabaud, V., Cottrell, G., Belongie, S.: Behavior recognition via sparse spatio-temporal features. In: Proc. PETS. (2005)
5. Schindler, K., van Gool, L.: Action snippets: How many frames does human action recognition require? In: Proc. CVPR. (2008)
6. Thurau, C., Hlaváč, V.: Pose primitive based human action recognition in videos or still images. In: Proc. CVPR. (2008)
7. Lin, Z., Jiang, Z., Davis, L.S.: Recognizing actions by shape-motion prototype trees. In: Proc. ICCV. (2009)
8. Weinland, D., Boyer, E.: Action recognition using exemplar-based embedding. In: Proc. CVPR. (2008)
9. Elgammal, A., Shet, V., Yacoob, Y., Davis, L.S.: Learning dynamics for exemplar-based gesture recognition. In: Proc. CVPR. (2003) 571–578
10. Efros, A.A., Berg, A.C., Mori, G., Malik, J.: Recognizing action at a distance. In: Proc. ICCV. (2003)
11. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proc. CVPR. (2006)
12. Ikizler, N., Cinbis, R., Duygulu, P.: Human action recognition with line and flow histograms. In: Proc. ICPR. (2008) 1–4
13. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: Proc. CVPR. (2005)
14. Ojala, T., Pietikinen, M., Harwood, D.: A comparative study of texture measures with classification based on featured distributions. Pattern Recognition **29** (1996) 51 – 59
15. Zach, C., Pock, T., Bischof, H.: A duality based approach for realtime TV-L1 optical flow. In: Proc. DAGM. (2007)
16. Wang, X., Han, T.X., Yan, S.: An hog-lbp human detector with partial occlusion handling. In: Proc. ICCV. (2009)
17. Yeffet, L., Wolf, L.: Local trinary patterns for human action recognition. In: Proc. ICCV. (2009)
18. Thurau, C., Hlaváč, V.: Pose primitive based human action recognition in videos or still images. In: Proc. CVPR. (2008)
19. Schindler, K., van Gool, L.: Action snippets: How many frames does human action recognition require? In: Proc. CVPR. (2008)
20. Gorelick, L., Blank, M., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. IEEE Trans. PAMI **29** (2007)
21. Blank, M., Gorelick, L., Shechtman, E., Irani, M., Basri, R.: Actions as space-time shapes. In: Proc. ICCV. (2005) 1395–1402
22. Jhuang, H., Serre, T., L.Wolf, Poggio, T.: A biologically inspired system for action recognition. In: Proc. ICCV. (2007)
23. Fathi, A., Mori, G.: Action recognition by learning mid-level motion features. In: Proc. CVPR. (2008)
24. Yao, B., Zhu, S.C.: Learning deformable action templates from cluttered videos. In: Proc. ICCV. (2009)