Efficient Human Action Recognition by Cascaded Linear Classification

Peter M. Roth, Thomas Mauthner, Inayatullah Khan, and Horst Bischof
Institute for Computer Graphics and Vision
Graz University of Technology
{pmroth,mauthner,khan,bischof}@icg.tugraz.at

Abstract

We present a human action recognition system suitable for very short sequences. In particular, we estimate Histograms of Oriented Gradients (HOGs) for the current frame as well as the corresponding dense flow field estimated from two frames. The thus obtained descriptors are then efficiently represented by the coefficients of a Non-negative Matrix Factorization (NMF). To further speed up the overall process, we apply an efficient cascaded Linear Discriminant Analysis (CLDA) classifier. In the experimental results we show the benefits of the proposed approach on standard benchmark datasets as well as on more challenging and realistic videos. In addition, since other state-of-the-art methods apply weighting between different cues, we provide a detailed analysis of the importance of weighting for action recognition and show that weighting is not necessarily required for the given task.

1. Introduction

Human action recognition, in general, is of wide interest in the computer vision community. Typical applications include visual surveillance, sports analysis, or human computer interaction. Even the general topic is quite comprehensive (e.g., different time scales, single person or multiple persons, etc.) the main scientific interest is in single person action recognition. In particular, the task can be defined as recognizing basic actions such as walking, waving, etc. performed by a single person (also on cluttered and moving backgrounds).

The simplest way for action recognition is to apply a single-frame classification method. For instance, Mikolajczyk and Uemura [22] trained a vocabulary forest on feature points and their associated motion vectors to describe a specific action. However, most classification methods are based on the analysis of a temporal window around a specific frame. Bobick and Davis [4] used motion history images to describe an action by accumulating human silhouettes over time. Blank et al. [3] created 3-dimensional space-time shapes to describe actions. Weinland and Boyer [31] used a set of discriminative static key-pose exemplars without any spatial order. Thurau and Hlaváč [30] used pose-primitives based on HOGs and represented actions as histograms of such pose-primitives. Even though these approaches show that shape or silhouettes over time are well discriminating features for action recognition, the use of temporal windows or even of a whole sequence implies that actions are recognized with a specific delay.

Using additional spatio-temporal information such as optical flow is an obvious extension. Efros et al. [8] introduced a motion descriptor based on spatio-temporal optical flow measurements. An interest point detector in spatio-temporal domain based on the idea of Harris point detector was proposed by Laptev and Lindeberg [17]. They described the detected volumes with several methods such as histograms of gradients or optical flow as well as PCA projections. Dollár et al. [6] proposed an interest point detector searching in space-time volumes for regions with sudden or periodic changes. In addition, optical flow was used as a descriptor for the 3D region of interest. Niebles et al. [23] used a constellation model of bag-of-features containing spatial and spatio-temporal [6] interest points.

Recent results in the cognitive sciences have led to biologically inspired vision systems for action recognition. Jhuang et al. [14] proposed an approach using a hierarchy of spatio-temporal features with increasing complexity. Input data is processed by units sensitive to motion-directions and the responses are pooled locally and fed into a higher level. But only recognition results for whole sequences have been reported. The required computational effort is approximately 2 minutes for a sequence consisting of 50 frames.

A more sophisticated approach was proposed by Schindler and van Gool [27]. Additionally to motion they use appearance information, where both, appearance and motion, are processed in similar pipelines using scale and orientation filters. In both pipelines the filter responses are max-pooled and compared to templates. The final action classification is done by using multiple one-vs-all SVMs.

However, all of these methods can not be applied for practical applications. They either rely on complex models, which can not be evaluated in real-time, or they are based on long-time observations. Thus, if short actions should be recognized in real-time, such methods can not be applied. Since there is a need for robust human action recognition on a short frame level, e.g., in crowded scenes, rapid movements in sport, etc., the goal of this paper is to introduce an efficient action recognition system that allows for action recognition on short-frame basis.

For our action recognition system we apply two information cues in parallel: appearance and motion. Thus, even if the appearance is very similar, additionally analyzing the corresponding motion information can help to discriminate between two actions; and vice versa. In particular, given a frame at time t, we compute a dense optical-flow field from frame $t-1$ and frame t. Thus, only two frames are required to estimate a model at time t (i.e., even using two frames we have a single frame representation).

Similar to [30] and [27] we use HOG descriptors and NMF to estimate a robust and compact representation. In addition, due to a GPU-based flow estimation and an efficient data structure for HOGs our system is real-time capable. Moreover, in contrast to our previous work [21] instead of using an SVM-based classification we use a much simpler cascaded Linear Discriminant Analysis (CLDA) approach, which dramatically reduces the computational effort. However, in the experiments we show that even using this simplified classification approach state-of-the-art results can be obtained on typical publicly available datasets. Moreover, we demonstrate the approach on human action detection in crowded scenes, where an efficient and fast action recognition system is beneficial or even required.

Previous approaches using an appearance and a motion cue (e.g., [27, 13] applied a general weighting between the two cues. However, it is obvious that for different actions different weights might be meaningful. Thus, in addition, we give a detailed analysis of the importance of weighting based on Multi Kernel Learning (MKL) [25]. In fact, this analysis shows that increasing the size of the representation such that sufficient recognition results can be obtained decreases the importance of weighting. Hence, in our approach we do not apply any weighting while still getting state-of-the-art results!

The paper is organized as follows. First, in Section 2 we introduce and discuss our new efficient action recognition system. Next, in Section 3 we analyze the importance of using weights for action recognition using different cures and show why, in contrast to existing methods, we do not use any weighting. In Section 4 we show experimental results of the proposed method on two publicly available action recognition datasets. Finally, in Section 5 we give a summary and a conclusion of the work!

2. Action Recognition System

In this section, we introduce our action recognition system, which is illustrated in Figure 1. In particular, we combine appearance and motion information to enable a frame-wise action analysis. To represent the appearance, we use histograms of oriented gradients (HOGs) [5]. To estimate the motion information, a dense optical flow field is computed between consecutive frames, which is then also represented by HOGs.

Following the ideas presented in [1, 30] we apply NMF to reduce the dimensionality of the extracted histograms. Hence, the actions are described in every frame by NMF coefficient vectors for appearance and flow, respectively. Moreover, this provides a more suitable and more effective representation.

Finally, the frame-wise classification is performed using a cascaded LDA classifier. In particular, we learn two stages of LDA classifiers, which are applied consecutively. Most decisions can already be made in the first stage. Thus, only the ambiguous samples have to be forwarded to the second stage. In addition, the computationally very cheap linear classifier allows a very efficient and fast classification.

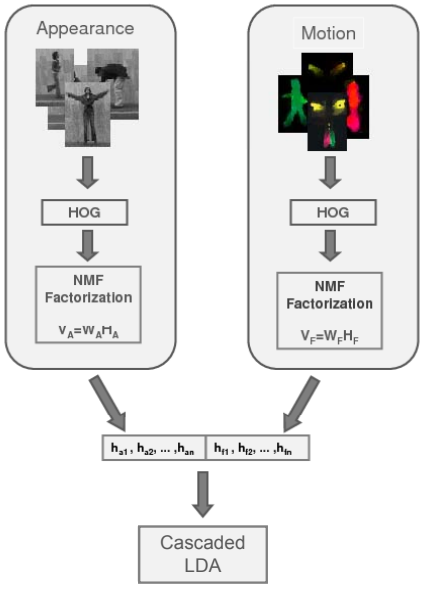


Figure 1. Overview of the proposed approach: Two representations are estimated in parallel, for appearance and flow. Both are described by HOGs and represented by NMF coefficients, which are combined to single feature vector. These vectors are then learned using a cascaded LDA classifier.

2.1. Appearance Features

Given an image $\mathbf{I_t} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$ at time step t. To compute the gradient components $g_x(x,y)$ and $g_y(x,y)$ for every position (x,y), the image is filtered by 1-dimensional masks $[-1,0,1]$ in x and y direction [5]. The magnitude $m(x,y)$ and the signed orientation $\Theta_S(x,y)$ are computed by

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

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

To make the orientation insensitive to the order of intensity changes, only unsigned orientations Θ_U are used for appearance:

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

To create the HOG descriptor, the patch is divided into non-overlapping 10×10 cells. For each cell, the orientations are quantized into 9 bins and weighted by their magnitude. Groups of 2×2 cells are combined in so called 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. The parameters for cell-size, block-size, and the number of bins may be different in literature.

2.2. Motion Features

In addition to appearance we use optical flow. Thus, for frame t the appearance features are computed from frame t, and the flow features are extracted from frames t and $t-1$. In particular, to estimate the optical dense flow field, we apply the method proposed in [32], which is publicly available: *OFLib*[1]. In fact, the GPU-based implementation allows a real-time computation of features.

Given $\mathbf{I_t}, \mathbf{I_{t-1}} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$, the optical flow describes the shift from frame $t-1$ to t with the disparity $\mathbf{D_t} \in \mathbb{R}^{\mathbf{m}\times\mathbf{n}}$, where $d_x(x,y)$ and $d_y(x,y)$ denote the disparity components in x and y direction at location (x,y). Similar to the appearance features, orientation and magnitude are computed and represented with HOG descriptors. In contrast to appearance, we use signed orientation Θ_S to capture different motion direction for same poses. The orientation is quantized into 8 bins only, while we keep the same cell/block combination as described above; but also other partitions might be applied.

2.3. Feature Representation

To represent the features extracted as described in Section 2.1 and Section 2.2 we use Non-negative Matrix Factorization [18]. In contrast to other sub-space methods, Non-negative Matrix Factorization (NMF) does not allow negative entries, neither in the basis nor in the encoding. This makes it highly suitable for sparse encoding. If the underlying data can be described by distinctive local information (such as the HOGs of appearance and flow) the representation is typically very sparse and can thus be described by NMF very well.

Formally, NMF can be described as follows. Given a non-negative matrix $\mathbf{V} \in \mathbb{R}^{m\times n}$ the goal of NMF is to find non-negative factors $\mathbf{W} \in \mathbb{R}^{m\times r}$ and $\mathbf{H} \in \mathbb{R}^{r\times n}$ that approximate the original data:

$$\mathbf{V} \approx \mathbf{WH} \ . \tag{4}$$

Since there is no closed-form solution, both matrices, $\mathbf{W}$ and $\mathbf{H}$, have to be estimated in an iterative way. Therefore, we consider the optimization problem

$$\begin{aligned} &\min ||\mathbf{V} - \mathbf{WH}||^2 \\ &\text{s.t. } \mathbf{W}, \mathbf{H} > 0 \ , \end{aligned} \tag{5}$$

where $||.||^2$ denotes the squared Euclidean distance. To solve the optimization problem Eq. (5) various numerical methods have be proposed (e.g.,[12, 11, 18]). In particular, in this work we apply the iterative multiplicative update rules proposed in [18].

2.4. Classification

For classification the NMF-coefficients obtained for appearance and motion are concatenated to a final feature vector. Typically, for that purpose multi-class SVM classifiers are applied, which implicitly depend on one-vs-all SVMs. Moreover, it was shown by [30, 1] that for action recognition tasks the NMF representations of the HOG coefficients are well linearly separable. Thus, even a simpler and thus computationally more efficient linear classifier might be applied for the classification. Thus, to overcome typical multi-class SVM problems (i.e., calibration problem) and to have an efficient classifier, instead of a multi-class SVM we use a cascaded LDA classifier.

2.4.1 Linear Discriminant Analysis

Linear Discriminant Analysis (LDA) is a linear inherently multi-class classification method. It was originally introduced by Fisher for two classes [9], but was later extended for multiple classes by Rao [26]. In particular, LDA computes a classification function

$$g(x) = \mathbf{W}^\top x \ , \tag{6}$$

where $\mathbf{W}$ is selected as the linear projection that maximizes the Fisher-criterion

$$\mathbf{W}_{opt} = \arg\max_{\mathbf{W}} \frac{\left|\mathbf{W}^\top \mathbf{S}_B \mathbf{W}\right|}{\left|\mathbf{W}^\top \mathbf{S}_W \mathbf{W}\right|} \ , \tag{7}$$

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

where $\mathbf{S}_W$ and $\mathbf{S}_B$ are the within-class and the between-class scatter matrices (see, e.g., [7]). The corresponding optimal solution for this optimization problem is given by the solution of the generalized eigenproblem

$$\mathbf{S}_B\mathbf{w} = \boldsymbol{\lambda}\mathbf{S}_W\mathbf{w} \tag{8}$$

or directly by computing the eigenvectors for $\mathbf{S}_W^{-1}\mathbf{S}_B$. Since the rank of $\mathbf{S}_w^{-1}\mathbf{S}_b$ is bounded by the rank of $\mathbf{S}_b$ there are $c-1$ non-zero eigenvalues resulting in a $(c-1)$-dimensional subspace $\mathbf{L} = \mathbf{W}^\top\mathbf{X} \in \mathbb{R}^{(c-1)\times n}$, which preserves the most discriminant information. For classification of a new sample $\mathbf{x} \in \mathbb{R}^m$ the class label $\omega \in \{1, \ldots, c\}$ is assigned according to the result of a nearest neighbor classification. For that purpose, the Euclidean distances d of the projected sample $g(\mathbf{x})$ and the class centers $\boldsymbol{\nu}_i = \mathbf{W}^\top\boldsymbol{\mu}_i$ in the LDA space are compared:

$$\omega = \underset{1\le i\le c}{\arg\min}\, d\left(g(\mathbf{x}), \boldsymbol{\nu}_i\right) \ . \tag{9}$$

2.4.2 Cascaded LDA Classifier

Loog et al. [19] showed that for more than two classes maximizing the Fisher criterion in Eq. (7) provides only a suboptimal solution! In particular, optimizing the Fisher criterion provides an optimal solution with respect to the Bayes error for two classes, but this can not be generalized for multiple classes. Nevertheless LDA can be applied for many practical multi-class problems. This was also confirmed by theoretical considerations by Martínez and Zhu [20]. However, they showed that increasing the number of classes decreases the separability.

Thus, to further improve the classification performance, we introduce a cascaded approach consisting of two stages. For the first stage a projection matrix $\mathbf{W}_c$ is estimated which contains all c classes. For a second stage, we estimate smaller pairwise projection matrices $\mathbf{W}_{ij}$ for the classes i and j. In the recognition stage an unknown sample $\mathbf{x}$ is projected onto $\mathbf{W}_c$ by Eq. (6) and is classified using Eq. (9). If the class label ω is well-defined (i.e., $\forall i \neq \omega : d\left(g(\mathbf{x}), \boldsymbol{\nu}_\omega\right) << d\left(g(\mathbf{x}), \boldsymbol{\nu}_i\right)$ the class label ω is assigned. Otherwise, the sample $\mathbf{x}$ is projected onto $\mathbf{W}_{ij}$, where i and j are the two best ranked classes, and the class label is finally assigned based on this local decision.

In this way ambiguous samples can be classified considerable better than using only the global space $\mathbf{W}_c$. However, as shown in the experiments, most of the samples can be already classified in the first stage. Hence the overall computational complexity is only slightly increased.

3. Influence of Weighted Classifiers

Previous action recognition methods applying different cues such as motion and appearance (e.g., [27, 13]) also explored the importance of weighting these different cues. The results showed that selecting the weights in the range $0.4 - 0.6$ provide the best results. Moreover, only a global weighting was considered. However, considering different actions different weights might be meaningful. For instance for an action like "running" motion is more important then for an action like "waving with one hand". Thus, in the following we give a detailed evaluation on the importance and the influence of weighting for action recognition. For that purpose, given specific actions, we apply Multi Kernel Learning (MKL) to estimate the optimal weights for the different information cues.

3.1. Multi Kernel Learning

Recently, Multiple Kernel Learning (MKL) [25, 16, 29] has become a quite popular method to combine data from multiple information sources. The main idea is to create a weighted linear combination of the kernels obtained from each information source. Moreover, in Rakotomamonjy et al. [25] it was shown that by using multiple kernels instead of one a more effective decision function can be obtained. In particular, the kernel $K(\mathbf{x}, \mathbf{x}')$ can be considered a convex combination of M basis kernels $K_j(\mathbf{x}, \mathbf{x}')$:

$$K(\mathbf{x}, \mathbf{x}') = \sum_{j=1}^{M} d_j K_j(\mathbf{x}, \mathbf{x}'), \tag{10}$$

where $d_j \geq 0$ are the weights of the kernels K_j and $\sum_{j=1}^{M} d_j = 1$. Thus, the decision function $g(x)$ of an SVM with multiple kernels can be represented as

$$\begin{aligned} g(x) &= \sum_{i=1}^{N} \alpha_i y_i K(\mathbf{x}_i, \mathbf{x}) - b \\ &= \sum_{i=1}^{N} \alpha_i y_i \sum_{j=1}^{M} d_j K_j(\mathbf{x}_i, \mathbf{x}') - b, \end{aligned} \tag{11}$$

where $\mathbf{x}_i$ are the training samples and $y_i \in \{-1, +1\}$ are the corresponding class labels. Hence, when training an MKL model the goal is to learn both, the coefficients α_i and the weights d_m, in parallel.

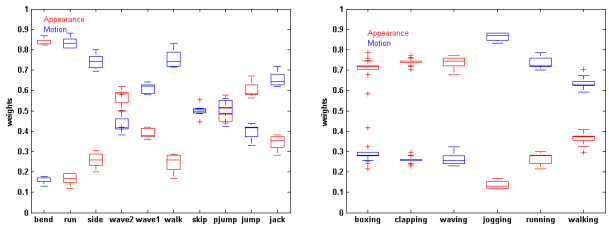
3.2. Weighted Results

In the following we analyze the importance of each information source for the classification task considering the MKL weights, which were estimated for linear kernels using the MKL method introduced in [25]. In our case, having only two cues, the convex combination of the basis kernels in Eq. (10) can be simplified to

$$K(\mathbf{x}, \mathbf{x}') = d_{mot} K_{mot}(\mathbf{x}, \mathbf{x}') + d_{app} K_{app}(\mathbf{x}, \mathbf{x}') \ , \tag{12}$$

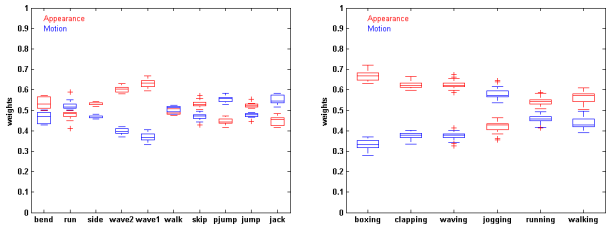
where the subscripts mot and app indicate the motion and the appearance components, respectively. The thus obtained

results obtained for the standard benchmark datasets described in Section 4.1 for 10 and 100 NMF modes are illustrated in Figure 2 and Figure 3, respectively.



(a) (b)

Figure 2. MKL weights using linear kernels for the Weizmann (a) and the KTH (b) dataset for an NMF representation of 10 modes.



(a) (b)

Figure 3. MKL weights using linear kernels for the Weizmann (a) and the KTH (b) dataset for an NMF representation of 100 modes.

These results clearly show that the different cues have different importance for different data; especially, if the representation size is quite small (see Figure 2). However, they also show that increasing the representation size (such that sufficient classification results can be obtained) the importance of weights is decreasing (see Figure 3). If the representation size is further increased, all weights are reaching approx. 0.5. Moreover, as can be seen from Figure 7 and Figure 8, for those actions where the classifications "fail" the weights are very similar and can thus not help to increase the classification power! Thus, for our approach, we do *not use any weighting* at all!

4. Experiments

To show the benefits of the proposed approach, we split the experiments into two main parts. First, we evaluated it on publicly available benchmark datasets (i.e., *Weizmann* [3] and *KTH* [28] human action datasets), which are described in Section 4.1. We show that even using a recognition system based on short frame basis (i.e., we require only two frames) competitive results can be obtained. In addition, to show the computational efficiency, we give an run-time analysis for our approach. Second, to demonstrate the generality and the robustness of the approach, we applied it for more realistic scenarios (i.e., [15]).

4.1. Benchmark Datasets

4.1.1 Weizmann Data Set

The Weizmann human action dataset [3] is a publicly available[2] dataset, that originally contained 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 [10]. Illustrative examples for each of these actions are shown in Figure 4.

Figure 4. Examples from the Weizmann human action dataset.

4.1.2 KTH Data Set

The KTH human action dataset[3], originally created by [28], 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 5.

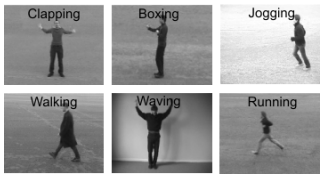


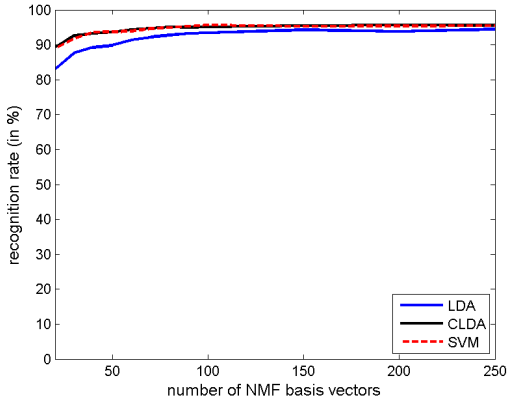
Figure 5. Examples from the KTH action recognition dataset.

4.2. Efficient Linear Classifier

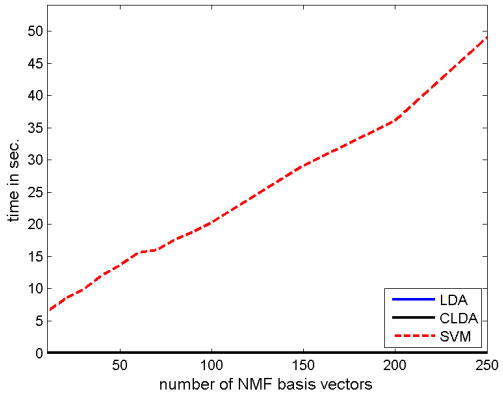
First of all, we want to demonstrate the benefits of using the cascaded LDA classifier (CLDA). For that purpose, we run action recognition experiments for the *Weizmann* database using different NMF representations and analyzed the classification performance as well as the run-time.

From Figure 6 the advantages of using CLDA instead of SVM for classification clearly can be seen. Figure 6(a) shows that the LDA classifier provides worse classification results compared to the SVM classifier. However, if the number of NMF basis vectors is increased LDA reaches a similar performance as SVM. In contrast, CLDA provides

[2] http://www.wisdom.weizmann.ac.il/ vision/SpaceTimeActions.html
[3] http://www.nada.kth.se/cvap/actions



(a)



(b)

Figure 6. CLDA vs. SVM for action classification: (a) classification results and (b) run-time analysis.

"exactly the same" results as SVM; if the the number of basis vectors is further increased (i.e., greater than 150) CLDA even slightly outperforms SVM! However, as can be seen from Figure 6(b), there are huge differences concerning the runtime for the classification. In fact, for a number of 250 modes, where all three methods yield a comparable classification performance, the LDA classifications are much more efficient (i.e., on a standard single core PC the classification time for the whole Weizmann dataset can be reduced from $50s$ to $0.04s$: speed up factor of approx. 1250)!

4.3. Classification results

Next, we demonstrate the proposed method for the *Weizmann* and the *KTH* benchmark datasets. Similar to, e.g., [30, 27] all experiments were carried out using a leave-one-out strategy (i.e., we used all individuals except one for training and evaluated the learned model for the missing one). In particular, the results for both datasets presented here were obtained using an NMF representation of 250 modes. The final obtained results compared to other state-of-the-art methods for the *Weizmann* dataset as well as the corresponding confusion matrix for the 10-actions database are given in Table 1 and in Figure 7, respectively.

method	rec.-rate	# frames
proposed *	97.0%	2
proposed	94.2%	2
Mauthner et al. [21]	91.3%	2
	94.3%	6
Thurau & Hlaváč [30]	70.4%	1
	94.4%	all
Niebles et al. [23]	55.0%	1
	72.8%	all
Schindler & v. Gool * [27]	93.5%	2
	96.6%	3
	99.6%	10
Blank et al. [3]	99.6%	all
Jhuang et al. [14]	98.9%	all
Ali et al. [2]	89.7%	all

Table 1. Recognition rates and number of required frames for different approaches reported for the *Weizmann* database. The results marked by * are obtained on the older 9-action data set.

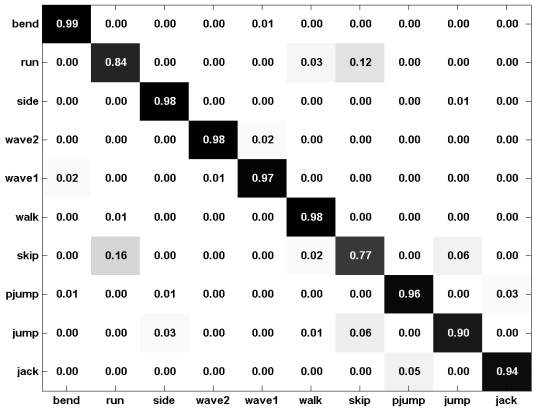


Figure 7. Confusion matrix for the *Weizmann* dataset.

From the results presented in Table 1 it can be seen that the proposed method yields competitive results and that it outperforms other approaches working on short frame basis[4]. In particular, using only 2 frames for the 9-actions and the 10-actions databases recognition results of 97% and 94% are obtained, respectively. Moreover, from the confusion matrix in Figure 7 it can be seen that we get excellent results but that the overall performance is slightly "degraded" since the actions "run" and "skip" are confused very often. This results from the fact that for both actions

[4]The improvement compared to our previous work in [21] is also the result of an improved NMF representation.

the appearance as well as the motion information are very similar. However, when looking at the optimal weights in Figure 3, it is clear that even using more sophisticated weights can not help to solve this problem.

For the KTH dataset we run the experiments for *s1*, *s3*, and *s4* subsets. Since the *s2* subset is mainly dealing with different scales, which was not considered in this work, no experiments were run for that dataset. The finally classification results for the other datasets are summarized in Table 2. The corresponding confusion matrix for dataset *s1* for our method is given in Figure 8.

dataset	*s1*	*s3*	*s4*
rec.-rate	88.1%	84.1%	88.4%

Table 2. Recognition rates for three different subsets of the *KTH* database.

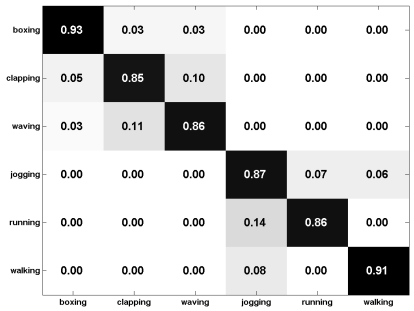


Figure 8. Confusion matrix for the *KTH-s1* dataset.

The overall performance compared to other state-of-the-art methods are given in Table 3. Please note that these results, in general, can not directly be compared, since all authors apply different data preparation (cropping of videos, data set selection, scaling, noise suppression, etc.)! In particular, we run the experiments on the full sequences, did no preprocessing, and used only a single scale. In fact, averaged over the three used datasets we get a recognition rate of 87%! Again from the confusion matrix in Figure 8 it can be seen that only very similar actions are mixed up.

method	rec.-rate	# frames
proposed	86.9%	2
Niebles et al. [24]	81.5%	all
Schindler & v. Gool [27]	88.0% 90.9% 92.7%	2 7 all
Dollár et al. [6]	81.2%	all
Jhuang et al. [14]	91.7%	all
Schüld et al. [28]	71.7%	all

Table 3. Recognition rates and number of required frames for different approaches reported for the *KTH* database.

4.4. Human Action Detection

Finally, we tested our proposed method for human action detection. We therefore choose the challenging dataset collected by [15], which consists of approximately 20 minutes of video material (160×120), acquired using hand-held cameras. It contains labeled humans performing events of interest, e.g., pick-up and handwave as well as many pedestrians in the background.

We used the Weizmann dataset for training and grouped the actions into three action sets for bending, walking/running, and waving/jumping-jack. All detection results are again achieved on a two frame basis. To handle scale variations, a HOG pyramid with six different scales has been used. Please note that we neither applied a global camera motion compensation nor have we trained an additional background class. This leads to some false detection of bending actions on textured background areas. Figure 9 shows eight detection examples for the three action sets and their corresponding optical flow fields. Red points depicts walking/running, green points bending and blue points waving/jumping detections[5].

Figure 9. Examples for action detection. First and third column shows detection results of bending (green), walking (red) and waving (blue) action. Corresponding optical flow fields are shown in column two and four.

5. Conclusion

We presented an efficient action recognition system based on a single-frame representation combining appearance-based and motion-based (optical flow) description of the data. Since in the evaluation stage only two consecutive frames are required (for estimating the flow), the method can also be applied for very short sequences. In

[5]Further results are given in the supplementary material.

particular, we propose to use HOG descriptors for both, appearance and motion. The thus obtained feature vectors are represented by NMF coefficients and are concatenated to learn action models using a cascaded LDA classifier. Since we apply a GPU-based implementation for optical flow, an efficient estimation of the HOGs, and a lightweight but very effective classifier, the method is highly applicable for tasks where quick and short actions have to be analyzed. The experiments show that even using this short-time analysis competitive results can be obtained on standard benchmark datasets. However, from these results it can also be concluded that similar to [27] by using longer sequences (which was not the intention this paper) even better results can be obtained. In addition, we analyzed the importance of weighting for action recognition and showed that competitive results can be obtained without weighting.

Acknowledgment

This work was supported by the FFG project AUTOVISTA (813395) under the FIT-IT programme, by the Austrian Joint Research Project Cognitive Vision under projects S9103-N04 and S9104-N04, by the Austrian Science Fund (FWF P18600), and by the Higher Education Commission of Pakistan under Overseas scholarships (Phase-II-Batch-I).

References

[1] A. Agarwal and B. Triggs. A local basis representation for estimating human pose from cluttered images. In *Proc. ACCV*, 2006.

[2] S. Ali, A. Basharat, and M. Shah. Chaotic invariants for human action recognition. In *Proc. ICCV*, 2007.

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

[4] A. F. Bobick and J. W. Davis. The representation and recognition of action using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[5] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proc. CVPR*, 2005.

[6] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie. Behavior recognition via sparse spatio-temporal features. In *Proc. IEEE Workshop on PETS*, 2005.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. John Wiley & Sons, 2000.

[8] A. A. Efros, A. C. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *Proc. ECCV*, 2003.

[9] R. A. Fisher. The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, 7:179–188, 1936.

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

[11] M. Heiler and C. Schnörr. Learning non-negative sparse image codes by convex programming. In *Proc. ICCV*, 2005.

[12] P. O. Hoyer. Non-negative matrix factorization with sparseness constraints. *J. of Machine Learning Research*, 5:1457–1469, 2004.

[13] N. Ikizler, R. Cinbis, and P. Duygulu. Human action recognition with line and flow histograms. In *Proc. ICPR*, 2008.

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

[15] Y. Ke, R. Sukthankar, and M. Hebert. Event detection in crowded videos. In *Proc. ICCV*, 2007.

[16] G. R. G. Lanckriet, T. d. Bie, N. Cristianini, M. I. Jordan, and W. S. Noble. A statistical framework for genomic data fusion. *Bioinformatics*, 20(16):2626–2635, 2004.

[17] I. Laptev and T. Lindeberg. Local descriptors for spatio-temporal recognition. In *Proc. ICCV*, 2003.

[18] D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.

[19] M. Loog, R. P. W. Duin, and R. Haeb-Umbach. Multiclass linear dimension reduction by weighted pairwise fisher criteria. *IEEE Trans. PAMI*, 23(7):762 – 766, 2001.

[20] A. M. Martínez and M. Zhu. Where are linear feature extraction methods applicable? *IEEE Trans. PAMI*, 27(12):1934 – 1944, 2005.

[21] T. Mauthner, P. M. Roth, and H. Bischof. Instant action recognition. In *Proc. SCIA*, 2009.

[22] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *Proc. CVPR*, 2008.

[23] J. C. Niebles and L. Fei-Fei. A hierarchical model of shape and appearance for human action classification. In *Proc. CVPR*, 2007.

[24] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning human action categories using spatio-temporal words. In *Proc. BMVC*, 2006.

[25] A. Rakotomamonjy, F. R. Bach, S. Canu, and Y. Grandvalet. SimpleMKL. *J. of Machine Learning Research*, 9:2491–2521, 2008.

[26] C. R. Rao. The utilization of multiple measurements in problems of biological classification. *Journal of the Royal Statistical Society – Series B*, 10(2):159–203, 1948.

[27] K. Schindler and L. van Gool. Action snippets: How many frames does human action recognition require? In *Proc. CVPR*, 2008.

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

[29] S. Sonnenburg, B. S. Bernhard, P. Bennett, and E. Parrado-Hernández. Large scale multiple kernel learning. *J. of Machine Learning Research*, 7, 2006.

[30] C. Thurau and V. Hlaváč. Pose primitive based human action recognition in videos or still images. In *Proc. CVPR*, 2008.

[31] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. CVPR*, 2008.

[32] C. Zach, T. Pock, and H. Bischof. A duality based approach for realtime TV-L1 optical flow. In *Proc. DAGM*, 2007.