Action Recognition by Learning Discriminative Key Poses

Shahzad Cheema[1] Abdalrahman Eweiwi[1] Christian Thurau[1, 2] Christian Bauckhage[1, 2]
[1]Bonn-Aachen Internatioal Center for IT, University of Bonn
[2]Fraunhofer IAIS, Sankt Augustin, Germany
{cheema, eweiwi}@bit.uni-bonn.de {christian.thurau, christian.bauckhage}@iais.fraunhofer.de

Abstract

This paper proposes a novel approach to pose-based human action recognition. Given a set of training images, we first extract a scale invariant contour-based pose feature from silhouettes. Then, we cluster the features in order to build a set of prototypical key poses. Based on their relative discriminative power for action recognition, we learn weights that favor distinctive key poses. Finally, classification of a novel action sequence is based on a simple and efficient weighted voting scheme that augments results with a confidence value which indicates recognition uncertainty. Our approach does not require temporal information and is applicable for action recognition from videos or still images. It is efficient and delivers real-time performance. In experimental evaluations for single-view action recognition and the multi-view MuHAVi data set, it shows high recognition accuracy.

1. Introduction

Human activity recognition from videos and still images is an important and active research area in computer vision. It serves a wide range of applications, *e.g.*, video surveillance, content based image retrieval, human-computer interaction, entertainment, etc. Often, human activities are categorized according to their duration or complexity. For example, there are *gestures* such as "smile" or "rotate head" or *(primitive) actions* such as "walk" or "turn back", as well as *(complex) activities* such as "cooking" or "playing cricket". In this paper, we focus on primitive actions that, when properly combined or sequenced (or put into context), could be used to explain more complex activities. In particular, we aim at recognizing actions that can be discriminated based on their pose.

Most existing work on action recognition relies on temporal cues. Many methods directly use motion features [9, 7] or spatio-temporal features [5, 4, 16, 10]. Other methods track local patches or interest points [12, 14] or use probabilistic models (*e.g.* n-grams or HMMs) to implicitly represent temporal contexts [19, 13, 20].

While motion information obviously plays an important role in action recognition, many human activities such as "running", "reading a book", "standing", or "playing football", can be recognized from only a single image or snapshot, assuming that the given pose is sufficiently distinctive or that enough context information is provided. Interestingly, only a few approaches have been introduced so far that work equally well for action recognition from videos and still images. A common idea of these approaches is to represent and classify human poses for each image or frame in a sequence [8, 21, 3, 19]. Common pose representations include silhouettes [21], line-pairs [3], histogram of oriented gradient (HoG) descriptors [18], or contour-HoG descriptors [8]. An action class is then usually represented as a histogram over a set of *key poses*, i.e. a representative pose of a complex action, or simply as concatenation of pose representations.

In this paper, we propose a novel non-temporal method for action recognition from videos and still images. In contrast to previous work, we apply a scale invariant contour feature for pose representation that can be efficiently computed from a silhouette image. For the representation of action classes, we make use of the idea of key poses. However, in addition to previous work, we rate the most discriminative key poses. For instance, key poses involved in a "turn back" action will include poses representing states of "standing", "walking", "turning-head", among others. Since key poses such as "standing" may be shared among different actions (*e.g.* "guard", "walk", etc.), we apply statistical learning to determine the relative importance of key poses. Additionally, the relative importance weights allow us to assign confidence values to classification results. As we do not use any temporal information, our model is suitable both for video and image based action recognition. By benchmarking on single- as well as on multi-view activity data sets, we demonstrate that our approach successfully deals with variations in view or distance.

The technical contribution of this work is twofold: (i) a novel combination of a contour-based pose representa-

tion and non-temporal key pose identification, (ii) a novel weighting scheme for rating the relative importance of key poses. Also, unlike various other approaches [21, 1], we do not require any subsampling, upsampling or trimming during training. We have no limitations with respect to the length of the considered video sequence, and the approach performs in real-time on any standard desktop computer or smartphone. This real time capability is of crucial importance in our intended application which aims at smartphone implementations of pose-based recognition of actions in *massive* image databases.

The reminder of the paper is organized as follows: Section 2 reviews related work. Section 3 provides details on the underlying contour-based feature. Details on the leading of discriminative key poses are given in Section 4. Section 5 reports details on our benchmark data, experiments, and results. Finally, Section 6 concludes the paper.

2. Related Work

The idea of using key poses for activity recognition has been applied frequently in previous work. Carlsson and Sullivan [6] used key-frame templates for action recognition. They recognized *forehand* and *backhand* tennis strokes in videos by computing an edge-based distance metric between candidate frames and manually chosen key-frames. Recently, Kilner et al. [11] used key poses to analyze 3D data in a multi-camera sports environment. However, their approach is not applicable if only one view is considered at a time. Thurau et al. [19] introduced a weighting scheme for histograms of key poses based on mutual information. In contrast to their work, we directly modify the weighting of each key pose and use a different weighting scheme. Weinland and Boyer [21] presents an exemplar-based embedding approach which does not use any motion information. Employing forward feature selection, they determine key-frames. The training data is then mapped to a distance space based on key frames. However, this is computationally demanding, especially when applied without subsampling on large, multi-view and multi-actor data sets. Our approach differs in two aspects. First, we use cluster centers as the representative key poses for each action class. Secondly, we model the inter-class variation by efficiently learning weights for key poses.

Our approach is most similar to recent work by Baysal et al. [3] that finds discriminative key poses using k-mediods and a ranking scheme over their potential score towards discriminating actions. We, on the other hand, propose the use of an intuitive, weighted voting scheme for classification. Also we propose to use a contour-based feature which is more informative and systematic than the manually marked *line-pair* edge segments considered in [3]. Our feature extraction is based on work by Dedeoglu et al. [8] who define a distance signal over object contours. For action representation, they use template histograms of key poses in a temporal context. We, on the other hand, avoid histograms or any other temporal model. This allows our action recognition approach to be applicable for both image- and video data sets.

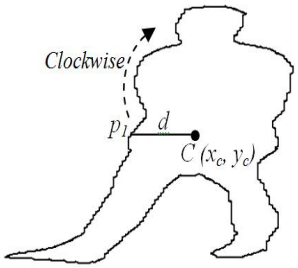


Figure 1. Contour-based feature extraction

3. Contour-based Pose Representation

Extraction of informative feature is crucial for success of human activity recognition. Binary silhouettes (or contours) are extensively employed to represent human actions [15, 13, 6, 20, 21]. Since the focus of this paper is on learning discriminative key poses, we assume silhouettes images to be available, which is indeed the case for many well-known benchmark data sets. Given a human silhouette, we extract its contour and transform it into a *distance space* as in [8]. Next, we describe details of this representation.

Let H be the binary silhouette image of an object. We determine its center of mass $C = (x_c, y_c)$ where

$$x_c = \frac{\sum_{i=1}^{n} x_i}{n}, y_c = \frac{\sum_{i=1}^{n} y_i}{n} \tag{1}$$

and n is the number of silhouette pixels.

Let $P = [p_1, p_2, ..., p_n]$ be the ordered set of contour points such that p_1 corresponds to the horizontal-left of C (see Fig. 1) and successive p_i are listed in a clockwise fashion. A distance vector $\mathbf{d} = [d_1, d_2, ..., d_n]$ is formed by calculating the Euclidean distance between p_i and C, i.e.

$$d_i = \|p_i - C\|, \qquad \forall i \in [1, 2, ..., n] \tag{2}$$

In order to provide a uniform representation for varying image sizes and shapes, $\mathbf{d}$ is scaled to a constant size s such that

$$\widehat{D[i]} = d\left\lceil \frac{i * n}{s} \right\rceil, \qquad \forall i \in [1, 2, ..., s] \tag{3}$$

where $\lceil \cdot \rceil$ is the ceiling function.

Finally, the scaled distance vector $\widehat{\mathbf{D}}$ is normalized to have unit sum:

$$\overline{D[i]} = \frac{\widehat{D[i]}}{\sum_1^s \widehat{D[i]}} \tag{4}$$

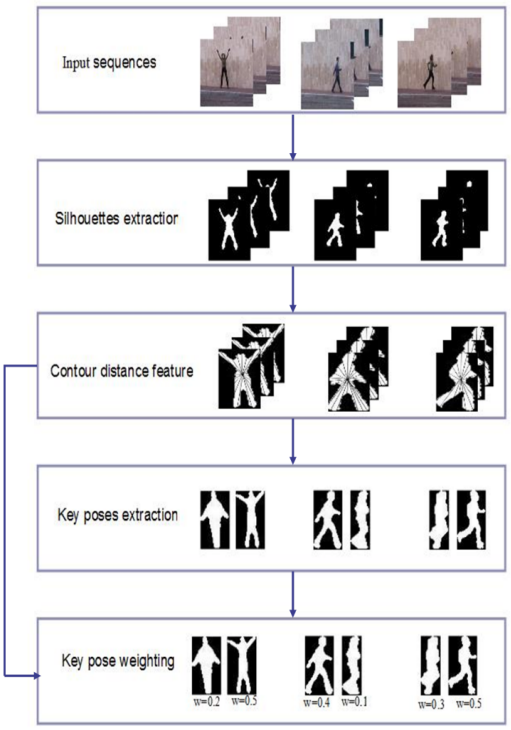


Figure 2. Overview of our approach

This contour-based feature is scale-invariant and can be efficiently extracted from silhouettes. Compared to the size of the original image, the size of the contour is much smaller. For example, in the MuHAVi data set, the resolution of the original silhouette images is 720×576 pixels whereas the contours of the silhouettes consist of only a few hundred pixels. The contour can be further scaled down if $s < n$. This implicit dimensionality reduction through transforming the silhouette to a distance signal ultimately enables efficient learning and classification.

4. Learning Discriminative Key Poses

Our approach to key pose extraction builds on [3] where key poses are learned over a space of line-pair segments. A significant feature of our approach is its ability to *adapt to* and to *exploit* the importance of key poses. Figure 2 summarizes the computational steps.

Given a set of labeled video sequences or still images, silhouettes can be extracted for all frames through background subtraction method which can be done reliably in many domains. For several benchmark data sets, including those considered in this paper, binary silhouettes are readily available which permits us to focus on the problem of action recognition. Given an extracted silhouette, each input frame is mapped to a normalized distance signal $\overline{D}$ of size s. By choosing s as a free parameter of the distance transform, the granularity of the resulting feature may be controlled.

Algorithm 1 Learning discriminative key poses

Input: Silhouettes for all input frames of all videos
Output: Key poses and their weights
Let k represents the number of clusters,
$A = \{a_1, a_2, ...a_r\}$ be the set of actions,
p_{ij} denotes $j - th$ key pose of action i and w_{ij} be the its weight

1: **for** all action $a \in A$ **do**
2: Cluster all frames into k groups using k-means
3: Take cluster centers as key poses thus ending up with $r \times k$ key poses
4: **end for**
5: **for** all actions $a \in A$ **do**
6: **for** all frames $f \in a$ **do**
7: Assign the key pose p_{ij} to f such that $\|f - p_{ij}\|$ is minimum
8: **end for**
9: **end for**
10: Let n_{ij} and $n_{ij}^{'}$ respectively denotes number of within-class assignments and number of out-of-class assignments to p_{ij}
11: $w_{ij} := \dfrac{n_{ij}}{n_{ij} + n_{ij}^{'}} \forall i, j$

In order to determine activity specific key poses from the available training data, we consider two successive steps. In the first stage, key poses are determined for each action by clustering all frames belonging to the corresponding class. In the second stage, weights are assigned to these key poses according to their ability to discriminate among different actions in the training data. Algorithm 1 summarizes the procedure.

Lines 1 – 4 corresponds to key pose extraction. We apply $k - means$ clustering with Euclidean distances to calculate key poses for each action. Since our model is strictly non-temporal, we do not create any histogram or ordering of key poses (KPs). Thus, key poses represent a *set* of different possible states of a primitive action. For example, key poses for the action "kick" may correspond to spatial states such as "standing", "arm adjustment", or "pulling the leg". Figure 3 shows an example of 8 key poses extracted from a video of the action KickRight in the MuHAVi data set. Notice that KP-2 and KP-6 through KP-8 do not seem to present distinctive states of the action. Instead, they look more related to actions such as "walk", "guard", or "punch". Yet they are automatically extracted since they apparently represent significant parts of the action sequence.

The issue of shared or ambiguous key poses is resolved by adopting a simple and intuitive mechanism of assigning rewards and penalties to key poses (Lines 5 – 11 of Al-

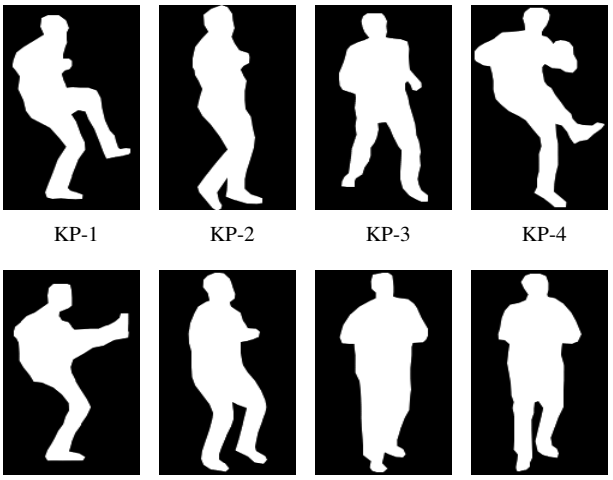


Figure 3. Different key poses for action KickRight in MuHAVi data set

gorithm 1). This procedure computes relative importance weights of key poses for different actions. By iterating over all actions, the closest key pose p_{ij} for each frame is determined. For each key pose p_{ij}, two values are n_{ij} and $n_{ij}^{'}$ are stored where the former denotes number of correct classifications to the key pose and latter denotes number of false assignments to the key pose. In this way those key poses which frequently matches to the frames from other classes, will have lower weights. On the other hand, key poses which appear only within one class will get higher weights.

From the perspective of key poses: if a key pose corresponds to frames from different action classes, it will have some false assignments which would decrease its weight. From the perspective of action classes: key poses which are common only *within* the action class and are discriminative with respect to other action classes will have higher weights. This mechanism also allows for automatically eliminating effects of overlapping actions. For instance, KickRight and GuardToKick in the MuHAVi data sets are two such actions for they share many common states such as "standing straight" or "standing in a punching position". Figure 4 shows the 8 top ranking key poses and their weights as determined by our approach. Notice that (a) the larger weights are assigned to more discriminative key poses (b) the poses corresponding to overlapping states(such as "standing in punching position" depicted by 7th key pose of KickRight and 2nd key pose of GuardToKick) have very different importance for the two actions. This indicates the ability of our approach to learn the relative importance of key poses.

In the application phase, in order to classify a given frame sequence, we first extract its contour feature. Then we determine the classes and the weights of the closest key poses for each query frame. Based on these weights, we apply a simple weighted voting scheme. Weights are accumulated for all related key poses and the label of the action class which has highest sum of weights is chosen. Notice that more discriminative poses dominate this process. In contrast to approaches such as [19, 3], this allows all query frames and all key poses to participate in the classification process. Due to a compact and non-temporal feature representation and a moderate number of key poses, we thus achieve real-time classification.

5. Experiments

To evaluate the effectiveness and robustness of our approach, we performed experiments on two well known data sets, namely the Weizmann collection [4] and the MuHAVi set [17]. In comparison to single-view Weizmann data, MuHAVi is a versatile multi-view action data set with more primitive action classes. All experiments presented in this section were carried out on a standard notebook computer using MATLAB 7. The Average processing rate was measured to be **56 frames per second**, indicating real-time applicability of the approach. In the following, we elaborate on the two data sets and our experimental results.

5.1. MuHAVi Data set

MuHAVi is a multi-camera and multi-action data set. It consists of videos of 17 activities performed multiple times by 14 actors. The action sequences are captured by 8 different CCTV cameras each with an angular difference of 45°. Silhouettes of 14 primitive actions performed by 2 actors (A1 and A4) captured from 2 views (45° and 90°) were manually annotated and made publicly available. This data set (also known as MuHAVi-MAS) provides 136 annotated silhouette sequences. In the following, we refer to this data as *MuHAVi-14*. The contained primitive actions can be further grouped into 8 action classes. For example, "WalkLeftToRight" and "WalkRightToLeft" may be merged into "Walk". We refer to this merged data set as *MuHAVi-8*.

In order to validate our approach w.r.t. the multi-view, multi-actor nature of the data set, we performed different experiments which are described next.

5.1.1 Leave-one-out Cross Validation

In this test, we iteratively trained the classifier on all instances except one and tested it on the left-out instance. Finally, the average accuracy was calculated over all 136 silhouettes. By using $k = 60$, we achieved an accuracy of up to 86.03% and 95.58% for MuHAVi-14 and MuHAVi-8, respectively. See Figures 5 and 6 for the resulting confusion matrices.

Notice that our approach is able to distinguish between actions involving similar poses in different temporal or-

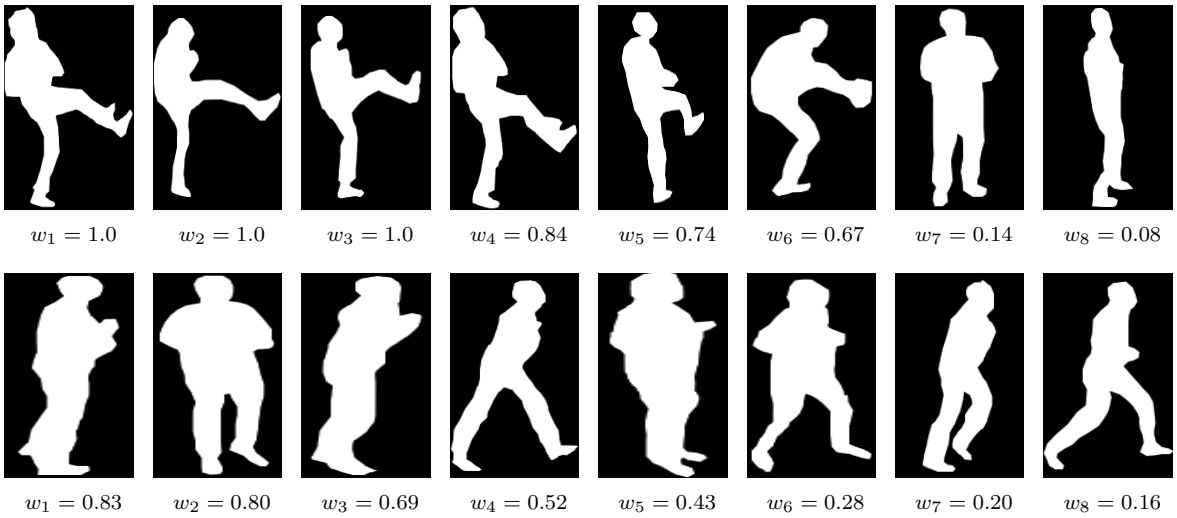


Figure 4. The 8 high-ranking key poses and their weights for each of the two *overlapping* actions ***KickRight*(first-row)** and ***GuardToKick*(second-row)** in the MuHAVi data set. The most distinctive and representative key-poses have higher weights. Key poses corresponding to overlapping states have different relative importance *e.g.* w_7 of KickRight and w_2 of GuardToKick.

	CollapseLeft	CollapseRight	GuardToKick	GuardToPunch	KickRight	PunchRight	RunLeftToRight	RunRightToLeft	StandupLeft	StandupRight	TurnBackLeft	TurnBackRight	WalkLeftToRight	WalkRightToLeft
CollapseLeft	7/8	1/8	0	0	0	0	0	0	0	0	0	0	0	0
CollapseRight	0	7/8	0	0	0	0	0	0	0	1/8	0	0	0	0
GuardToKick	0	0	12/16	3/16	0	0	0	0	0	0	0	1/16	0	0
GuardToPunch	1/16	0	4/16	10/16	1/16	0	0	0	0	0	0	0	0	0
KickRight	0	0	0	0	16/16	0	0	0	0	0	0	0	0	0
PunchRight	0	0	0	0	0	16/16	0	0	0	0	0	0	0	0
RunLeftToRight	0	0	0	0	0	0	7/8	0	0	0	0	0	1/8	0
RunRightToLeft	0	0	0	0	0	0	0	8/8	0	0	0	0	0	0
StandupLeft	0	0	0	0	0	0	0	0	1/4	3/4	0	0	0	0
StandupRight	0	0	0	0	0	0	0	0	0	8/8	0	0	0	0
TurnBackLeft	0	0	0	0	0	0	0	0	0	0	2/4	0	1/4	1/4
TurnBackRight	0	0	0	0	0	0	0	0	0	0	0	7/8	1/8	0
WalkLeftToRight	0	0	0	0	0	0	0	0	0	0	0	0	8/8	0
WalkRightToLeft	0	0	0	0	0	0	0	0	0	0	0	0	0	8/8

Figure 5. Confusion Matrix for MuHAVi-14

der. For instance, "Collapse" and "Standup" as well as actions involving many overlapping poses in the same order (*e.g.* "GuardToKick" and "KickRight") can be distinguished. Action-wise comparisons to the temporal baseline approach are listed in Tables 1 and 2.

5.1.2 Identical Training and Test Cameras, Novel Test Actor

In this experiment, we trained our classifier on all instances related to one actor and tested on the data of the other actor and calculated average classification rates. A comparison with the baseline is given in Tables 3 and 4. Note that the baseline evaluation in [17] was based on training on Actor-1 and testing on Actor-4. However, we alternatively considered both actors for training and testing.

Again, we observe a significant improvement in accuracy for both MuHAVi-14 and MuHAVi-8 collections. An increase of about 12% in accuracy, for MuHAVi-14, shows the relative robustness of our approach towards individual characteristics of actors. Althoug human silhouettes differ in test and train data, the novel combination of scale-invariant features with discriminative keyposes learning exhibits improved performance.

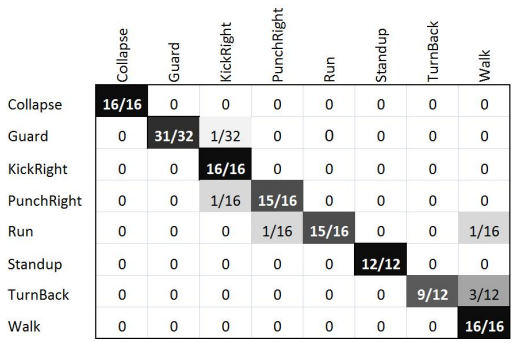


Figure 6. Confusion matrix for MuHAVi-8

	Accuracy (%)	
Action	**Baseline**[17]	**Our Approach**
CollapseLeft	50.0	87.5
CollapseRight	62.5	87.5
GuardToKick	81.2	75.0
GuardToPunch	62.5	62.5
KickRight	93.7	100.0
PunchRight	100.0	100.0
RunLeftToRight	87.5	87.5
RunRightToLeft	87.5	100.0
StandUpLeft	0.0	25.0
StandUpRight	100.0	100.0
TurnBackLeft	100.0	50.0
TurnBackRight	87.5	87.5
WalkLeftToRight	100.0	100.0
WalkRightToLeft	87.5	100.0
	82.35	**86.03**

Table 1. Action-wise comparison of our approach with the baseline on MuHAVi-14

	Accuracy (%)	
Action	**Baseline**[17]	**Our Approach**
Collapse	100.0	100.0
Guard	100.0	96.9
KickRight	93.7	100.0
PunchRight	100.0	93.7
Run	93.7	93.7
StandUp	100.0	100.0
TurnBack	91.7	75.0
Walk	100.0	100.0
	97.80	95.58

Table 2. Action-wise comparison of our approach with the baseline on MuHAVi-8

5.1.3 Identical Training and Test Actors, Novel Test Camera

This experiment aims to determine robustness of the algorithm towards changes in the view-point. Here, we trained our classifier on all instances captured by one camera and tested on data captured from the other camera. We alternatively considered both camera-views for training and testing. Our average results for the two cases are compared with the baseline in Tables 5 and 6.

These results reflect the challenging nature of this problem. In particular, we notice a low performance of our pose-based approach for the actions where the novel pose involves high self-occlusion (e.g. GuardToPunch and PunchRight). We expect that, if size and variety of training data were increased (*e.g.* by adding more actors or views to the training set), our simple yet effective approach will perform even better.

	Accuracy (%)	
Action	**Baseline**[17]	**Our Approach**
CollapseLeft	75.0	87.5
CollapseRight	50.0	87.5
GuardToKick	0.0	68.7
GuardToPunch	0.0	25.0
KickRight	87.5	81.3
PunchRight	100.0	68.7
RunLeftToRight	100.0	75.0
RunRightToLeft	75.0	100.0
StandUpLeft	0.0	50.0
StandUpRight	100.0	75.0
TurnBackLeft	50.0	75.0
TurnBackRight	50.0	75.0
WalkLeftToRight	100.0	100.0
WalkRightToLeft	100.0	75.0
	61.76	**73.53**

Table 3. Novel actor validation on MuHAVi-14

	Accuracy (%)	
Action	**Baseline**[17]	**Our Approach**
Collapse	75.0	100.0
Guard	50.5	75.0
KickRight	87.5	81.25
PunchRight	100.0	62.5
Run	87.5	75.0
StandUp	83.3	100.0
TurnBack	50.0	83.3
Walk	100.0	100.0
	76.47	**83.08**

Table 4. Novel actor validation on MuHAVi-8

5.2. Weizmann Data set

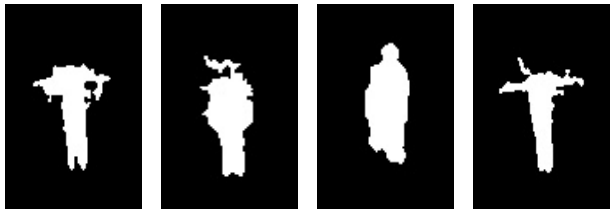
The Weizmann data [4] is a popular single-view action data set which contains video samples for 10 different actions performed by 9 actors. A common tradition is to consider only 9 actions by eliminating the samples of the action "skip". In this paper, we consider readily available silhouettes for the 9 actions in Weizmann data set. It is worth noting that many of these silhouettes are very noisy (*e.g.*

Action	Accuracy (%)	
	Baseline[17]	**Our Approach**
CollapseLeft	0.0	87.5
CollapseRight	50.0	37.5
GuardToKick	0.0	50.0
GuardToPunch	25.0	0.0
KickRight	87.5	100.0
PunchRight	75.0	62.5
RunLeftToRight	0.0	50.0
RunRightToLeft	0.0	50.0
StandUpLeft	50.5	25.0
StandUpRight	100.0	62.5
TurnBackLeft	100.0	25.0
TurnBackRight	75.0	62.5
WalkLeftToRight	0.0	0.0
WalkRightToLeft	75.0	62.0
	42.6	**50.0**

Table 5. Novel view validation on MuHAVi-14

Action	Accuracy (%)	
	Baseline[17]	**Our Approach**
Collapse	62.5	56.3
Guard	18.7	40.6
KickRight	87.5	87.5
PunchRight	75.0	56.2
Run	0.0	37.5
StandUp	83.3	91.6
TurnBack	83.3	83.3
Walk	37.5	37.5
	50.0	**57.4**

Table 6. Novel view validation on MuHAVi-8



jack pjump side wave2

Figure 7. Examples of some noisy silhouettes in Weizmann data set

see Figure 7). We use them *as they are* without any preprocessing.

For assessing the performance of our approach on this data set, we resorted to leave-one-out cross validation. We used 80 points on the contour and the best performance was achieved for only 20 key poses per action. The resulting confusion matrix is shown in Figure 8.

In Table 7, we compare our approach to other non-temporal approaches. Only Weinland and Boyer [21] achieve significantly higher accuracy than the proposed.

	bend	jack	jump	pjump	run	side	walk	wave1	wave2
bend	9/9	0	0	0	0	0	0	0	0
jack	0	9/9	0	0	0	0	0	0	0
jump	0	0	9/9	0	0	0	0	0	0
pjump	0	0	0	7/9	0	0	0	0	2/9
run	0	0	0	0	10/10	0	0	0	0
side	0	0	0	2/9	1/9	5/9	1/9	0	0
walk	0	0	0	0	1/10	0	9/10	0	0
wave1	0	0	0	0	0	0	0	9/9	0
wave2	0	0	0	0	0	0	0	0	9/9

Figure 8. Confusion matrix for Weizmann data set (excluding "skip")

Approach	Act.	Seq.	Acc.(%)
Thurau 2007 (unigram) [18]	10	90	86.6
Weinland and Boyer 2008 [21]	10	90	100
Baysal et al. 2010 [3]	9	81	92.6
This Paper	9	83	91.6

Table 7. Comparison of our approach with other non-temporal approaches on Weizmann data set

However, recall that, in contrast to their method, our approach does not require any subsampling of the data. Moreover their approach is based on forward selection of key poses which is computationally expensive for large and versatile action datasets. Thurau [18] reports accuracies of 86.6% and 94.4% by using non-temporal unigrams and 2-frame temporal bigrams, respectively. In terms of methodology, the work of Baysal et al. [3] is most close to approach. It appears that by using a contour-based pose features, we can achieve very close accuracy . Notice further that we achived this accuracy by using only 20 key poses per action compared to 47 key poses per action in their approach. Moreover our approach is very efficient for its feature extraction, dimensionality reduction, and similarity measure.

6. Conclusion

In this paper, we presented a novel, simple yet effective approach to pose based action recognition in videos and still images. We could show that employing a contour based pose representation and an efficient weighting scheme that favors distinctive key poses, a very high recognition accuracy can be achieved on standard benchmark data, even though the presented approach does not incorporate any temporal information or implicit modeling of the underlying sequence of key poses.

While we are confident that the addition of temporal cues might further increase the accuracy, the high recognition rates for a strictly pose based approach are an interesting result. Although our approach already outperforms a recent

baseline in more difficult *novel-actor* and *novel-view* scenarios, it may be further improved by enlarging the set of training data.

In our future work, we are mainly interested in application scenarios involving large image databases and handheld devices.An important aspect of future research will therefore be to estimate how many training samples are required for a sufficiently accurate estimation of key pose weights. From what we could observe so far, it appears that the weighting coefficients converge quickly but it remains to see if this is an artifact of the data sets considered here. Also, we are currently invoking clustering methods based on archetypal analysis [2] that are designed to yield more distinctive key poses during training.

References

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

[2] C. Bauckhage and C. Thurau. Making archetypal analysis practical. In *Proc. DAGM*, 2009.

[3] S. Baysal, M. C. Kurt, and P. Duygulu. Recognizing human actions using key poses. In *Proc. ICPR*, 2010.

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

[5] A. Bobick and J. Davis. The recognition of human movement using temporal templates. *IEEE Trans. PAMI*, 23(3):257–267, 2001.

[6] S. Carlsson and J. Sullivan. Action recognition by shape matching to key frames. In *IEEE computer society workshop on models versus exemplars in computer vision*, 2001.

[7] R. Cutler and L. Davis. Robust real-time periodic motion detection, analysis, and application. *IEEE Trans. PAMI*, 22(8):257–267, 2000.

[8] Y. Dedeoglu, B. U. Toereyin, U. Gueduekbay, and A. E. Cetin. Silhouette-based method for object classification and human action recognition. In *Proc. ECCV workshop on HCI*, 2006.

[9] A. Fathi and G. Mori. Action recognition by learning mid-level motion features. In *Proc. CVPR*, 2008.

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

[11] J. Kilner, J-Y.Guillemaut, and A. Hilton. 3d action matching with key-pose detection. In *Proc. ICCV workshop on search in 3D and video*, 2009.

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

[13] F. Martnez-Contreras, C. Orrite-Uruuela, J. H. J., H. Ragheb, and S. A. Velastin. Recognizing human actions using silhouette-based hmm. In *Proc. of AVSS*, 2009.

[14] J. C. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learing of human action categories using spatial-temporal words. In *Proc. BMVC*, 2006.

[15] F. Niu and M. Abdel-Mottaleb. Hmm-based segmentation and recognition of human activities from video sequences. In *Proc. ICME*, 2005.

[16] P. M. Roth, T. Mauthner, I. Khan, and H. Bischof. Efficient human action recognition by casecade linear classification. In *Proc. ICCV*, 2009.

[17] S. Sing, S. A. Velastin, and H. Ragheb. Muhavi: A multicamera human action video dataset for the evaluation of action recognition methods. In *Proc. AVSS*, 2010.

[18] C. Thurau. Behavior histograms for action recognition and human detection. In *ICCV workshop on human motion*, 2007.

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

[20] L. Wang and D. Suter. Recognizing human activities from silhouettes: motion subspace and factorial discriminative graphical model. In *Proc. CVPR*, 2007.

[21] D. Weinland and E. Boyer. Action recognition using exemplar-based embedding. In *Proc. CVPR*, pages 1–7, 2008.