Motion Based Foreground Detection and Poselet Motion Features for Action Recognition

Erwin Kraft[1] and Thomas Brox[2]

[1]Fraunhofer ITWM, Fraunhofer-Platz 1, Kaiserslautern, Germany
[2]University of Freiburg, Georges-Köhler-Allee 52, Freiburg, Germany
erwin.kraft@itwm.fhg.de, brox@cs.uni-freiburg.de

Abstract. For action recognition, the actor(s) and the tools they use as well as their motion are of central importance. In this paper, we propose separating foreground items of an action from the background on the basis of motion cues. As a consequence, separate descriptors can be defined for the foreground regions, while combined foreground-background descriptors still capture the context of an action. Also a low-dimensional global camera motion descriptor can be computed. Poselet activations in the foreground area indicate the actor and its pose. We propose tracking these poselets to obtain detailed motion features of the actor. Experiments on the Hollywood2 dataset show that foreground-background separation and the poselet motion features lead to consistently favorable results, both relative to the baseline and in comparison to the current state-of-the-art.

1 Introduction

All actions involve an actor and in most cases the actor must move to perform the action. Surprisingly, these facts have not been used much in the literature on action recognition. State-of-the-art works rather rely on global feature aggregation that do not make explicit use of the notion of an actor [2–6]. Exceptions are Ullah et al. [7], who run a person detector to find persons, and Prest et al. [8], who even try to detect action specific objects, such as cups and cigarettes. Wang et. al. [6] also use a person detector but it is only used to improve their video stabilization method.

In static action classification, the important role of the actor is more appreciated, which is reflected by the fact that in the Pascal VOC Action Classification challenge, the bounding box of the actor is already provided [9]. In the 2012 challenge, a task was offered, where only the coarse location of the actor is provided, but there was not any submission on this task.

In this paper, we advocate focusing on foreground items of an action, which includes especially the actor. In contrast to Ullah et al. [7] and Prest et al. [8], who suggested finding the actor and other persons in the video directly with a person detector, we propose to first detect the foreground items based on motion before running poselet detectors [12] to localize the (relevant) actor(s) in these foreground areas. While there has been much progress on person detectors in

Fig. 1. Foreground scores are computed from sparse point trajectories (left). The intensities of the red pixels reflect the soft foreground scores. From these we can compute dense saliency maps (right).

Fig. 2. Motion based foreground detection for some videos of the Hollywood2 dataset [1]. Foreground objects are marked with red pixels. Features computed specifically on the foreground help action classification.

recent years [10–12], person detection is still error prone. Here we exploit the fact that the actor usually has to move to perform an action, i.e., motion indicates the relevance of a person detection for the action. Motion cues enable separation of moving foreground objects from the background quite reliably, as demonstrated in [13]. Admittedly, there are cases, where items of an action and even the main actor itself are static. However, such failure cases are quite rare, and they are outnumbered by those cases where current person detectors fail. Hence, compared to [7], we achieve much better performance. We also compare to [14], where an eye tracking system was used to emphasize the part of the image that humans consider most important. Although this approach uses additional supervision, we obtain higher performance with the proposed motion based saliency.

In the foreground we collect poselet activations for two reasons. First, poselets indicate which parts of the foreground are indeed persons. As the examples in Fig. 2 and Fig. 7 show, the foreground in an action dataset consists mostly

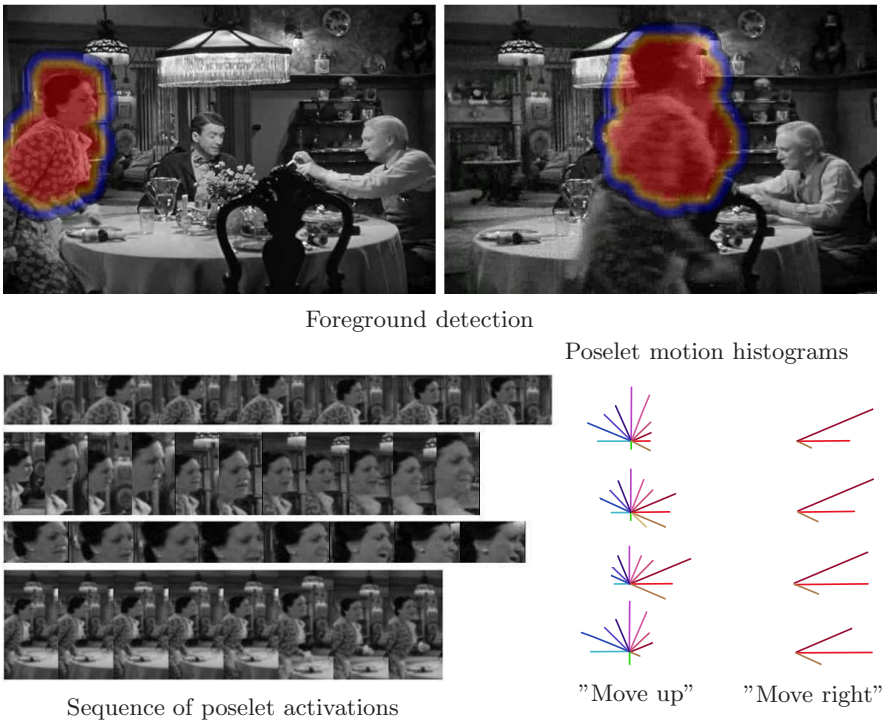

Fig. 3. Illustration of the proposed poselet motion features: we detect poselets [12] in the foreground and describe their motions over time. Thus, we encode pose information as well as the movement of specific body parts into our feature representation.

of persons, but it can contain also a car that enters the scene and stops before a person (the real actor) gets out. Second, poselets are not just person detectors but each poselet type indicates a specific body part and pose. Already static poselet activations help classify actions, as shown in [15, 16]. Here we propose motion histograms accumulated for each poselet along its trajectory, which makes a strong feature on how body parts move in a particular action; see the illustration in Fig. 3.

In general, having separate foreground features is advantageous since many actions are mostly independent of the scene background. For instance, the action "Answer Phone" may occur in multiple settings: indoors, while riding a car, or outdoors in a park. In all those cases the background holds only very little information about the action class, while foreground objects (persons holding a phone) are the main indicators.

On the other hand, there *is* correlation between the action and the background. Therefore, the strategy here is to have descriptors specific to the foreground *additionally* to joint descriptors that can capture the context. The same is true for motion compensation, as used in [4–6]. While it is advantageous to

A	Trajectory aligned features from Wang et al. [3]
B	A on motion compensated trajectories
C	A only on foreground
D	B only on foreground
E	Global camera motion histogram
F	Global foreground motion histogram
G	Poselet motion features [12]

Table 1. Overview of all features used in our action recognition approach.

define separate descriptors for motion compensated videos to be invariant to the camera motion, there is much correlation between certain actions and the camera motion. Rather than the ignorance of such correlation, we advocate the separation into correlated and non-correlated descriptors. For this reason, we also propose a very low-dimensional descriptor that explicitly represents the camera motion, which is a by-product of the foreground separation. Table 1 gives an overview of the overall feature combination. We show that this leads to state-of-the-art results on the challenging Hollywood2 dataset [1].

2 Motion Based Foreground Separation

In this section we aim for separating the actor and other moving objects that may be involved in the action from the background. We assume that for action recognition the most relevant parts of an image are those that show independent motion, i.e., the segmentation comes down to a special type of motion segmentation. The foreground separation allows us to define features that are independent of the background.

Previously, [5, 6] have used video stabilization by computing a global homography and subtracting it from the motion vectors. The main motivation for such motion compensation is the improved invariance of motion features to the camera motion. For this reason, we also use video stabilization when we compute motion features. However, motion compensation also allows for a rough foreground/background separation because, under good conditions, background trajectories become static. In [5, 6] a simple thresholding of the motion magnitude after stabilization was used to emphasize foreground objects. In Figure 4 we illustrate that this procedure oftentimes does not lead to a clean foreground separation.

The method we propose is based on clustering dense trajectories, which are computed using the LDOF-tracker from [17]. In contrast to [13], we are interested in exactly two clusters (foreground and background) and we model the background with a rigid 3D motion model. First, we compute a foreground score for each trajectory and then use a standard spatio-temporal minimum cut to obtain a segmentation.

Fig. 4. Background feature pruning based on video stabilization versus foreground saliency maps. **Top row:** Sample input sequence. **Middle row, left:** Optical flow. **Middle row, center:** Optical flow after motion compensation. **Middle row, right:** Foreground saliency map computed with our approach. **Bottom row, left:** Case without camera motion compensation keeps all feature points. **Middle:** The camera motion has been compensated and features on static trajectories have been pruned [5, 6]. Due to the remaining residual motion many features in the background are not pruned. **Right:** The proposed foreground saliency maps are based on a much more accurate motion model and include the long-term aspect of motion. The foreground separation is much cleaner.

2.1 Foreground Scores

Foreground scores are computed by comparing each trajectory to a background motion model that is estimated with a factorization approach [18, 19]. This motion model explores the assumption that under orthogonal projections the background motion can be described by three trajectories. A projection matrix P_τ is constructed from trajectories $\mathbf{w}_i = [\mathbf{x}_{1,i}^\mathrm{T}, ..., \mathbf{x}_{F,i}^\mathrm{T}] \in \mathbb{R}^{2F}$, where $\mathbf{x}_{t,i} = (x_{t,i}, y_{t,i})^\mathrm{T}$ are the spatial positions of the tracked points at time t, and F is the size of a temporal window centered at time τ [19]:

$$\mathrm{P}_\tau = \mathrm{W}_3(\mathrm{W}_3^\mathrm{T}\mathrm{W}_3)^{-1}\mathrm{W}_3^\mathrm{T}. \tag{1}$$

The matrix $\mathrm{W}_3 = [\mathbf{w}_i^\mathrm{T}\mathbf{w}_j^\mathrm{T}\mathbf{w}_k^\mathrm{T}]$ holds the three trajectories $\mathbf{w}_i$, $\mathbf{w}_j$ and $\mathbf{w}_k$, which describe the background motion hypothesis. The likelihood that a trajectory $\mathbf{w}$ is compatible with this hypothesis is measured by the projection distance:

$$f_\tau(\mathbf{w}|\mathrm{W}_3) = \|\mathrm{P}_\tau\mathbf{w} - \mathbf{w}\|_2 \tag{2}$$

The background motion is found using RANSAC. Motion hypotheses are scored by computing the projection errors for all trajectories and selecting the 25% quantile as a score.

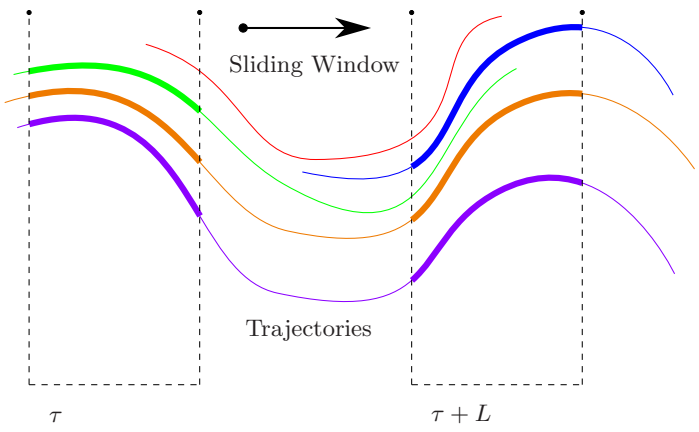

Fig. 5. Illustration of the sliding temporal window approach: only those trajectories that cover the temporal window completely are considered.

Once a background model has been estimated, the projected distance of a trajectory is converted into a foreground score:

$$s_\tau(\mathbf{w}) = \exp\left(-\frac{f_\tau(\mathbf{w}|\mathrm{W}_3)}{2\sigma_0^2}\right), \tag{3}$$

where σ_0 is the average distance over all trajectories. Small s_τ indicate trajectories that do not fit to the background model. Thus we use $1 - s_\tau$ as the foreground score and s_τ as the background score.

The above approach assumes that all trajectories have the same length. In practice this is not the case due to occlusion. Hence, we run the approach in sliding temporal windows of size $L = 6$. In each window we only consider those trajectories that fully cover that window; see Figure 5. Since most of the LDOF-trajectories are much longer than the window size, they typically receive multiple scores, one from each window they cover. We assign the maximum foreground score (and equivalently the minimum background score) as the overall score for a trajectory:

$$s(\mathbf{w}) = \min_\tau s_\tau(\mathbf{w}). \tag{4}$$

2.2 Binary Segmentation and Foreground Saliency Maps

We formulate the dense, binary segmentation as independent minimum cut problems on each frame. To ensure a labeling that is smooth in time, we build the graph for a spatio-temporal volume around the frame of interest (size 5 in time) and use a 26-neighborhood to connect the pixels in the volume. The unary term is only defined for the pixels covered by a trajectory. For pixels i covered by a trajectory $\mathbf{w}$ we have costs $\theta_i(0) = (1 - l(\mathbf{w})s(\mathbf{w}))$ and $\theta_i(1) = l(\mathbf{w})s(\mathbf{w})$,

Fig. 6. Dense, binary segmentations are obtained from the sparse foreground scores (red pixels in the left image) by solving a set of minimum cut problems with the foreground scores as unary cost. The segmented image is then turned into a saliency map by applying a Euclidean distance transform.

where $l(\mathbf{w})$ is the length of the trajectory normalized to a range between 0 and 1. Longer trajectories are assigned a larger score to emphasize objects that are visible for a longer time, since these are more likely to correspond to the actor in the shot. The energy on the binary labels X_i

$$E(X) = \sum_i \theta_i(X_i) - \kappa \sum_{i,j} \delta(X_i, X_j) \tag{5}$$

is minimized with the code from Kolmogorov [20]. δ is the 0-1 indicator function and we set $\kappa = 0.18$. This parameter was optimized using grid search on a small subset of the training set of the Hollywood2 [1] benchmark. Figure. 6 shows an example of the final segmentation of the actor.

Since features in the area directly around the actor are often beneficial for action recognition, we extend the foreground region to a saliency map by applying a Euclidean distance transform; see Figure 7.

3 Poselet Motion Features

It has been shown by Jhuang et al. [21] that high-level pose information improves action recognition performance significantly. Unfortunately, person detectors [12, 11] and pose estimators [22] are not yet reliable enough to provide fine-grained information about the position of limbs and body parts on challenging action datasets. For instance, person detectors have difficulties with poses that are uncommon in static images but appear more often in videos. We experimented with the person detector from Bourdev et al. [12] on a small set of action clips and found that the detection scores decrease strongly when the pose changes from standing to sitting. This problem can be approached by tracking detections over time, i.e., scores from easy frames are propagated to more difficult ones.

Rather than relying on the functioning of a person detector, we consider the statistics of the raw poselet activations in the foreground region. Restricting the activations to the foreground ignores persons not involved in the action (as they

Fig. 7. Soft saliency maps generated from our foreground segmentation on some sample videos from the Hollywood2 benchmark [1]. The main actors are well covered by our saliency maps and are clearly separated from the scene background.

do not move) but also false positive detections in the background; see Figure 8. In contrast to a full person detector, the poselets also localize certain body parts and their rough pose. It was demonstrated in [16] that the pose of a person can be encoded by the poselet activation vector. This may not be accurate, but still helps classify the action.

We are particularly interested in how the body parts described by the poselets move. For instance, it can be expected that the head and torso have to move upwards to perform the action "stand up" and downwards to perform the action "sit down". We extract the motion of a poselet from LDOF-trajectories [17]. For each poselet activation we consider all trajectories which are located inside the predicted bounding box. For all activations of a certain poselet we aggregate the motion vectors from an 8-frame time window in a motion histogram. A separate motion histogram is computed from stabilized and non-stabilized motion vectors. The histograms have 16 angular bins and two temporal bins. The latter allows consideration of whether a certain motion pattern appears more at the beginning or more at the end of a shot. Each motion vector votes with its magnitude for the respective bins using bilinear interpolation. The aggregation area of the temporal bins is deduced from the median length of the LDOF-trajectories. This ensures that a motion sequence will not be disrupted as it might be the case with evenly sized temporal bins. Since the video clips are in general very short, we found that a higher temporal resolution does not provide any further useful information. All motion histograms are normalized using the RootSIFT

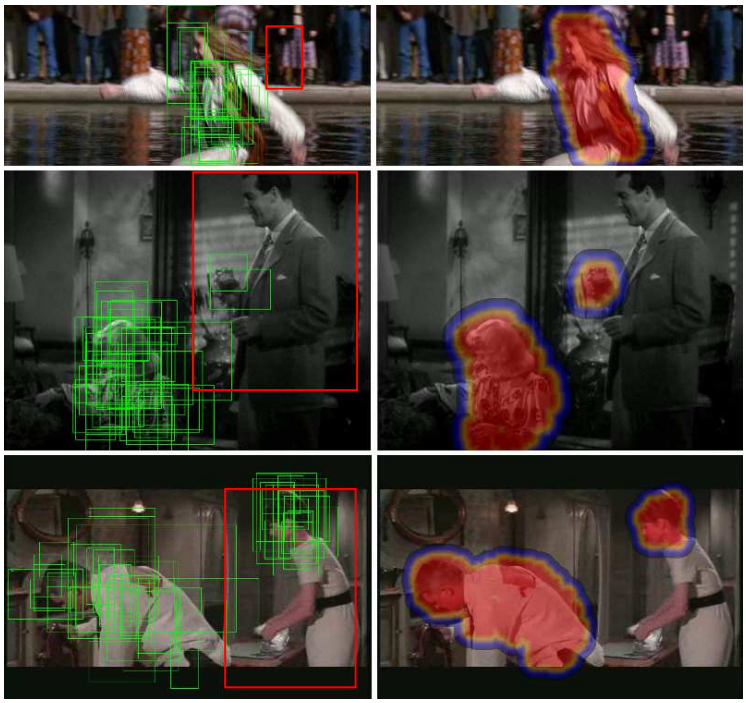

Fig. 8. Left: Poselet activations in three sample images. Green boxes show individual poselet activations [12] in the foreground region. Red boxes show the detected persons using [12]. **Right:** Foreground saliency maps. In all these examples, the main actor is missed by the person detector. Increasing the detection threshold would also lead to more false positives in the dataset. The motion based foreground saliency, on the other hand, indicates the main actor correctly in most cases. The individual poselet activations in the foreground indicate specific body parts and poses. We use the histogram over the motion of these poselets as features for classifying the action; see Fig. 9.

method [23]. Some poselet detections and the corresponding motion histograms are shown in Figure 9.

We consider 150 poselets. For each we have two motion histograms (one aggregated over the stabilized and one over the non-stabilized motion vectors) and two temporal bins. Poselets that are not active in the first and/or second part of a shot lead to a zero histogram. This means, the presence of a certain body part and pose is implicitly part of the overall feature vector.

4 Descriptors and Classification

With the foreground saliency maps from Section 2 and the poselet motion features from Section 3 we can define the descriptors as listed in Table 1. We build

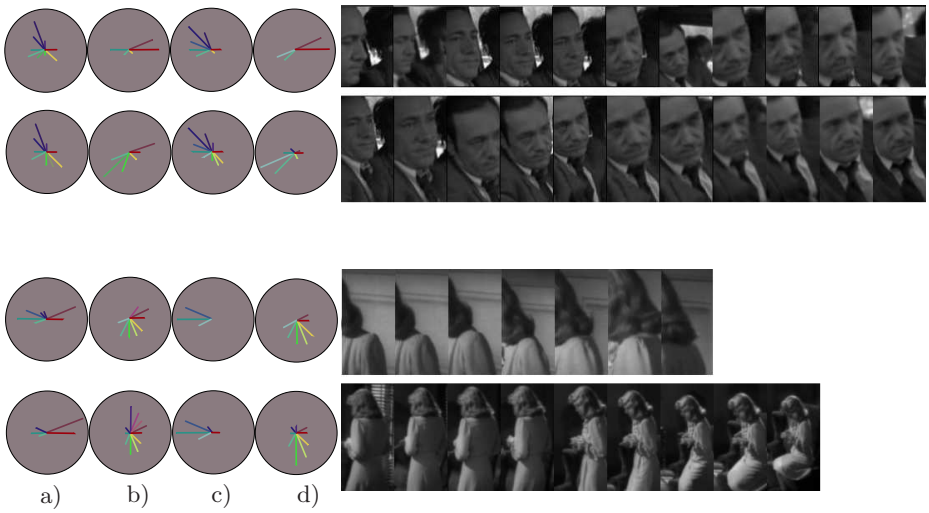

Fig. 9. Left: Poselet motion histograms for two video shots and two different poselets each. **a+b:** Histograms computed on non-stabilized motion vectors. **c+d:** Histograms computed on stabilized motion vectors. The poselet motion histograms have two temporal bins. a+c and b+d show the earlier and the later temporal bin, respectively. **Right:** Some of the activations of each of these poselets.

upon the local descriptors introduced by Wang et al. [3]: trajectory aligned HOG, HOF and MBH descriptors [10, 24], as well as normalized trajectory shape descriptors. The descriptors can be seen as histograms that capture local image and motion structure along a trajectory path of fixed length. They are aggregated individually using the VLAD representation [25].

Based on the same methodology, we create additional VLAD features by using motion compensated trajectories similar to [5], and by using the foreground saliency maps. In the latter case, feature aggregation takes into account the saliency scores as weights. Features located in the background have very small or zero weights, which means they have hardly any influence.

We add the poselet motion features as described in Section 3. Moreover, we suggest using two global histograms, which encode the camera motion and the foreground motion. The camera motion histogram is aggregated over non-stabilized trajectories in the background, while the foreground motion histogram is computed from camera motion compensated trajectories in the foreground region. The histograms capture the directions of the trajectory displacements in the same way as for the poselet motion features. Each displacement vector votes with its magnitude into one of the 16 angular bins of the histogram and there are two temporal bins that cover the beginning and the end of the video clip. The global motion feature vector has a total size of 64 dimensions. An example is shown in Figure 10.

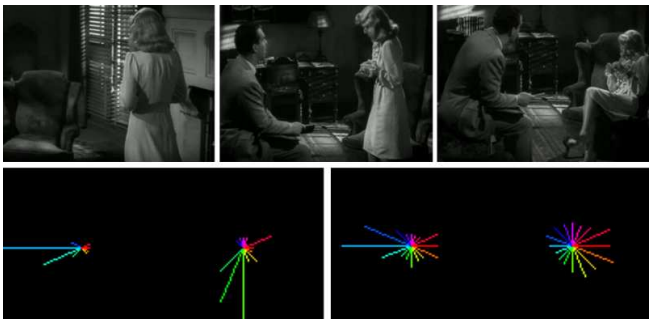

Fig. 10. Camera motion and foreground motion histograms at the beginning and the end of a video shot. Lines visualize the relative number of motion vectors pointing in a certain direction, as comprised in the histogram bins. Despite the correlation between camera motion and foreground motion, which is due to the camera following the actor, there are clear differences between foreground and background motion. This is exploited in our representation.

	Hollywood2 mAP [1]
Marzałek et al. [1]	33.9%
Gilbert et. al. [26]	50.9%
Ullah et al. [7]	55.7%
Wang et al. [3]	58.3%
Jiang et al. [4]	60.3%
Vig et al. [14]	61.9%
Jain et al. [5]	62.5%
Oneaţă et al. [27]	63.3%
Wang et al. [6]	64.3%
Our method	**67.8**%

Table 2. Comparison of our results to the currently best results reported in literature.

All features are combined in the kernel function of a multi-channel SVM:

$$K(v_i, v_j) = \sum_c h(v_i^c, v_j^c), \tag{6}$$

where v_i^c and v_j^c are the individual features with respect to the c-th channel and $h(v_i^c, v_j^c)$ computes the histogram intersection between v_i^c and v_j^c. The SVM-parameter $C = 0.001$ was selected based on cross-validation on the training set of the Hollywood2 benchmark [1].

5 Experiments

We evaluated our action recognition approach using the Hollywood2 benchmark [1]. This benchmark consists of 1707 video clips collected from 69 movies.

It is divided into a training set (823 clips) and a test set (884 clips). Both sets were sampled from different movies. We increased the size of the training set by mirroring the training video clips. There are 12 action classes: stand up, sit down, sit up, run, get out of car, hand shake, eat, drive, kiss, hug, fight and answer phone. Each action is evaluated by computing the average precision, and the mean average precision (mAP) is reported as overall result for the benchmark. The Hollywood2 benchmark can be considered as a very challenging benchmark. Many video clips contain multiple shots and show frequent and dynamic camera motion. Moreover, some of the actions have to be learned from a very small number of training samples. To show that the approach works on other datasets, we also evaluated it on the Olympic Sports dataset [28]. The performance is 77.5% (without poselets) and 85.5% (with poselet motion features). However, the Olympic Sports dataset [28] uses a very small test set (some actions are evaluated on only 3-4 video clips). This means that the overall score for Olympic Sports can be strongly affected based on just a few test examples. We therefore focus our detailed evaluation on the Hollywood2 benchmark where classification results depend on a significantly larger test set. A comprehensive overview of the plethora of different action recognition benchmarks can be found in [29].

Table 2 shows that our approach outperforms all other methods reported in literature so far. In particular, results are better than those of Ullah et al. [7], who used a person detector, and Vig et al. [14], who built upon saliency from an eye tracking system. A detailed analysis of the contribution of each set of features is presented in Table 3. It reveals that already the foreground saliency maps in conjunction with the global motion histograms compare favorably to the current state-of-the-art by [6]. It also shows that separating the foreground and background based on our saliency maps leads to much better results than using just stabilized features.

The new poselet motion features increase the results even further, especially the actions stand up, sit down, sit up, eat and kiss show a strong improvement in performance while the other actions do not benefit. We assume that this is because the performance of poselet detectors is mostly limited to certain body parts, such as the head or shoulder. Some actions, e.g. hand shake, would require strong poselets on hands and arms. Also, the vast majority of the 150 poselets represents body parts seen from the front. Table 4 shows the performance of each individual action with and without poselets. By using poselets only on action classes where they improved performance on the validation set, the overall test set performance can be improved further.

5.1 Computation Times

Computation times for a $576 * 304$ video with 144 frames on a single workstation (purchased in the year 2009) are as follows: $288s$ for optical flow computation and point tracking, $47s$ for the non-stabilized features by Wang et al. [3], $127s$ including stabilization. Our contributions saliency map computation and poselet motion features require $160s$ and $199s$, respectively. The most computationally

Hollywood2 Action Class	Motion-Stabilization B	Foreground-Separation with Saliency Maps (A-F)	Foreground-Separation with Saliency Maps + Poselet Motion (A-G)
Stand Up	77.3%	79.2%	**81.8**%
Sit Down	76.6%	78.9%	**80.3**%
Sit Up	38.1%	44.2%	**50.8**%
Run	80.8%	**85.3**%	84.9%
Get out of Car	68.1%	**68.8**%	64.3%
Hand Shake	43.6%	**48.4**%	46.9%
Eat	60.2%	66.2%	**69.1**%
Drive Car	96.3%	**97.2**%	97.0%
Kiss	61.6%	68.2%	**70.2**%
Hug	37.6%	**46.2**%	45.8%
Fight	**81.4**%	81.1%	79.5%
Answer Phone	30.5%	35.8%	**36.0**%
mAP	62.7%	66.6%	**67.2**%

Table 3. Action recognition results for the Hollywood2 benchmark [1]. We report the average precision (AP) for each action class and the mean average precision (mAP) as overall score. An overview and description of the individual features named A-G is given in Table 1.

expensive operations are: (1) LDOF computation, (2) running the poselets detector, (3) running the RANSAC part of the background subtraction, (4) video stabilization.

6 Conclusions

We have demonstrated that explicit separation of the video into foreground and background and computation of separate features has positive effects on action recognition performance. Since foreground detection with a person detector is still erroneous, we have proposed a bottom-up approach based on point trajectories. This is justified by the fact that most actions require the main actors (or the relevant parts of them) to move relative to the background. The foreground-background separation also allows us to define global, low-dimensional histograms for the camera motion and the foreground motion. To include more detailed evidence on the motion of the actor, we have proposed poselet motion features, which indicate how a certain body part in a certain pose usually moves in a certain action. These features strongly improve action recognition performance when actors are seen from a frontal view. We conclude that this is an artifact of the current poselet detectors and the poselet motion features will become even more useful as body part detectors will improve. Even now our method sets the state-of-the-art on the most challenging Hollywood2 dataset.

Hollywood2 Action Class	Validation Score (A-F)	Validation Score (A-G)	Use Poselet Motion	Score on Test Set
Stand Up	64.3%	**66.5**%	**Yes**	81.8%
Sit Down	61.7%	**68.5**%	**Yes**	80.3%
Sit Up	25.8%	**27.9**%	**Yes**	50.8%
Run	**93.3**%	85.6%	No	85.3%
Get out of Car	**30.6**%	21.8%	No	68.8%
Hand Shake	**43.3**%	27.3%	No	48.4%
Eat	39.5%	**47.3**%	**Yes**	69.1%
Drive Car	**89.1**%	88.3%	No	97.2%
Kiss	**67.1**%	62.7%	No	68.2%
Hug	**35.0**%	34.8%	No	46.2%
Fight	**74.7**%	70.8%	No	81.1%
Answer Phone	**28.1**%	23.8%	No	35.8%
mAP				**67.8**%

Table 4. Automatic selection of poselet motion features based on cross-validation: we split the Hollywood2 training set randomly into 7 subsets. For each feature set (A-F and A-G) we compute the mAP on the validation subsets and select the median as final score. An overview and description of the individual features named A-G is given in Table 1.

References

1. Marzałek, M., Laptev, I., Schmid, C.: Actions in Context. Computer Vision and Pattern Recognition (CVPR) (2009) 2929–2936
2. Laptev, I., Lindeberg, T.: Space-Time Interest Points. International Conference on Computer Vision (ICCV) (2003) 432–439
3. Wang, H., Kläser, A., Schmid, C., Liu, C.-L., Action Recognition by Dense Trajectories. Computer Vision and Pattern Recognition (CVPR) (2011) 3169–3176
4. Jiang, Y.-G., Dai, Q., Xue, X., Liu, W., Ngo, C.-W.: Trajectory-Based Modeling of Human Actions with Motion Reference Points. European Conference on Computer Vision (ECCV) (2012) 425–438
5. Jain, M., Jégou, H. Bouthemy, P.: Better Exploiting Motion for Better Action Recognition. Computer Vision and Pattern Recognition (CVPR) (2013) 2555–2562
6. Wang, H., Schmid, C.: Action Recognition with Improved Trajectories. International Conference on Computer Vision (ICCV) (2013) 3551–3558
7. Ullah, M. M., Parizi, S. N., Laptev, I.: Improving Bag-of-Features Action Recognition with Non-local Cues. British Machine Vision Conference (BMVC) (2010) 1–11
8. Prest, A., Schmid, C., Ferrari, V.: Weakly Supervised Learning of Interactions Between Humans and Objects. Pattern Analysis and Machine Intelligence (PAMI) (2012) 601–614
9. Pascal Visual Object Challenge. http://pascallin.ecs.soton.ac.uk/challenges/VOC/
10. Dalal, N., Triggs, B.: Histograms of Oriented Gradients for Human Detection. Computer Vision and Pattern Recognition (CVPR) (2005) 886–893
11. Felzenszwalb, P., Girshick, R., McAllester, D., Ramanan, D.: Object Detection with Discriminatively Trained Part Based Models. IEEE Transactions on Pattern Analysis and Machine Intelligence (2010) 1627–1645

12. Bourdev, L., Maji, S., Brox, T., Malik, J.: Detecting People Using Mutually Consistent Poselet Activations. European Conference on Computer Vision (ECCV) (2010) 168–181
13. Brox, T., Malik, J.: Object Segmentation by Long Term Analysis of Point Trajectories. European Conference on Computer Vision (ECCV) (2010) 228–295
14. Vig, E., Dorr, M., Cox, D.: Space-Variant Descriptor Sampling for Action Recognition Based on Saliency and Eye Movements. European Conference on Computer Vision (ECCV) (2012) 84–97
15. Yang, W., Wang, Y., Mori, G.: Recognizing Human Actions from Still Images with Latent Poses. Computer Vision and Pattern Recognition (CVPR) (2010) 2030–2037
16. Maji, S., Bourdev, L., Malik, J.: Action Recognition from a Distributed Representation of Pose and Appearance. Computer Vision and Pattern Recognition (CVPR) (2011) 3177–3184
17. Sundaram, N., Brox, T., Keutzer, K.: Dense Point Trajectories by GPU-accelerated Large Displacement Optical Flow. European Conference on Computer Vision (ECCV) (2010) 438–451
18. Tomasi, C., Kanade, T.: Shape and Motion From Image Streams Under Ortography: A Factorization Method. International Journal of Computer Vision (1992) 137–154
19. Sheikh, Y., Javed, O., Kanade, T.: Background Subtraction for Freely Moving Cameras. International Conference on Computer Vision (ICCV) (2009) 1219–1225
20. Kolmogorov, V., Zabih, R.: What Energy Functions Can be Minimized via Graph Cuts? IEEE Transactions on Pattern Analysis and Machine Intelligence (2004) 147–159
21. Jhuang, H., Gall, J., Zuffi, S., Schmid, C., Black, M. J.: Towards Understanding Action Recognition. International Conference on Computer Vision (ICCV) (2013) 3192–3199
22. Zuffi, S., Black, M. J.: From Pictorial Structures to Deformable Structures. Computer Vision and Pattern Recognition (CVPR) (2012) 3546–3553
23. Arandjelović, R., Zisserman, A.: Three Things Everyone Should Know to Improve Object Retrieval. Computer Vision and Pattern Recognition (CVPR) (2012) 2911–2918
24. Dalal, N., Triggs, B., Schmid, C.: Human Detection Using Oriented Histograms of Flow and Appearance. European Conference on Computer Vision (ECCV) (2006) 428–439
25. Jégou, H., Perronnin, F., Douze, M., Sánchez, J., Pérez, P., Schmid, C.: Aggregating Local Image Descriptors into Compact Codes. IEEE Transactions on Pattern Analysis and Machine Intelligence (2012) 1704–1716
26. Gilbert, A., Illingworth, J., Bowden, R.: Action Recognition using Mined Hierarchical Compound Features. IEEE Transactions on Pattern Analysis and Machine Intelligence (2011) 883–897
27. Oneaţă, D., Verbeek, J., Schmid, C.: Action and Event Recognition with Fisher Vectors on a Compact Feature Set. International Conference on Computer Vision (ICCV) (2013) 1817–1824
28. Niebles, J. C., Chen, C.-W., Fei-Fei, L.: Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. European Conference on Computer Vision (ECCV) (2010) 392–405
29. Hassner, T.: A Critical Review of Action Recognition Benchmarks. Computer Vision and Pattern Recognition Workshops (CVPRW) (2013) 245–250