Machine Vision and Applications (2014) 25:85-98
DOI 10.1007/s00138-013-0514-0

Selection of negative samples and two-stage combination of multiple features for action detection in thousands of videos

G.J. Burghouts, K. Schutte, H. Bouma, R.J.M. den Hollander

TNO, Intelligent Imaging, The Hague, The Netherlands

e-mail: gertjan.burghouts@tno.nl

Abstract

In this paper, a system is presented that can detect 48 human actions in realistic videos, ranging from simple actions such as 'walk' to complex actions such as 'exchange'. We propose a method that gives a major contribution in performance. The reason for this major improvement is related to a different approach on three themes: sample selection, two-stage classification, and the combination of multiple features. First, we show that the sampling can be improved by smart selection of the negatives. Second, we show that exploiting all 48 actions' posteriors by two-stage classification greatly improves its detection. Third, we show how low-level motion and high-level object features should be combined. These three yield a performance improvement of a factor 2.37 for human action detection in the visint.org test set of 1,294 realistic videos. In addition, we demonstrate that selective sampling and the two-stage setup improve on standard bag-of-feature methods on the UT-Interaction dataset, and our method outperforms state-of-the-art for the IXMAS dataset.

***Keywords**: human action detection, sparse representation, pose estimation, interactions between people, spatiotemporal features, STIP, tracking of humans, person detection, event recognition, random forest, support vector machines.*

1. Introduction

The amount of image and video data being recorded and stored is increasing daily, both on the internet (e.g. YouTube) and for surveillance applications. This yields practical problems when analysts want to extract information from the huge data collection. What is needed are tools for automatic analysis of the data. One of the key technological capabilities to aid search for the relevant data, is automated detection of specific events in videos. In many applications, the relevancy of events is determined by the actions that are being performed by the humans in the scene. A typical event is that one person approaches the other, walks up to the other person, and gives him something. These actions, 'walk', 'approach' and 'give', occur in a particular order, and are partially overlapping. This complicates the search for particular events. Therefore, in this paper, we address the challenge of detecting many actions, occurring in isolation, or simultaneously.

We consider the technological capability to analyse single- and multi-action events, where actions may involve a single actor ('jump'), multiple actors ('approach'), items ('give'), and interaction with the environment ('arrive'). An excellent overview of datasets related to action recognition was presented by Liu [36], however, the datasets discussed therein are not sufficient to test such a capability. Some datasets are single-actor, e.g. KTH [1] and Weizmann [2], and the performance on these datasets has almost been saturated [3] by methods with key components that serve as the basic pipeline in this paper (see Sec. 2). The MSR action dataset [37] contains only three actions but is very challenging because it requires spatio-temporal localization of the action, which is not a problem addressed by this paper. The UCF Sports [4], Hollywood2 [5], YouTube [6], Olympic sports [38] and the Columbia Consumer Video [46] datasets are more challenging (see e.g. [7]) as they involve interactions with other people and items and the recording conditions are hard. These actions in these datasets are all very context-dependent, which makes it a concept search rather than purely an action recognition task. The IXMAS [39] and UT-Interaction [20] datasets contain subtle actions with 12 action types [39] and two-person interactions with 6 action types [20], respectively. A very challenging dataset is the visint.org [8], where multiple actions may happen at the same time, or just before or after each other. This dataset includes 4,774 videos of a wide range of events that involve a large number (48) of human actions, in various forms including humans inside a car or on a motor bike or bicycle. Due to its realism and complexity, we select the visint.org dataset for the study in this paper on human action detection. In addition, we will include IXMAS [39] and UT-Interaction [20] datasets in our experiments, which allows a comparison to state-of-the-art action recognition methods. Together these two datasets represent a broad scope of possible actions.

The events in the visint.org dataset are realistic and each event is represented by multiple actions: on average 7 actions per movie clip. While each video scripts a single event, this event can be represented by several actions. For example, a transaction between persons can contain the actions “move”, “give” and “receive”. The actions vary from a single person (e.g. walk) to two or more persons (e.g. follow). Some of these actions are defined by the involvement of some object (e.g. give), or an interaction with the environment (e.g. leave). The most complex actions involve two persons and an object (e.g. exchange, throw-catch combination). Figure 1 depicts a few illustrations of this dataset. The dataset contains a test set of 1,294 realistic videos with highly varying recording conditions and on average 195 variations of each of the 48 actions. A complicating property of this dataset is that the actions are highly unbalanced: e.g. 1,947 positive learning samples for “move” to only 58 samples for “bury”, see Figure 2.

Fig. 1. Human actions in the visint.org dataset include persons, cars, interactions with other persons or cars, involvement of items like the balls (of which the ball in the upper-right video is practically not detectable), clutter in the background (such as the small car, at the upper-left video), and varying scenes and recording conditions. Green boxes depict the bounding boxes on which the EP features are based (Section 2.2), including e.g. pose information (as in the upper-right).

The improvement in human action detection performance as proposed in this paper follow from the observation that particular human actions are highly correlated, see Figure 3. One category of actions shows correlations due to similarity, e.g. chase and follow. Another category is the sequential relations, e.g. leave and exit. There is also a category of compound actions, e.g. to exchange something requires both to give and to receive something. Indeed, all of the actions from these examples, and many others (see Figure 3), are highly correlated in the human annotations of the visint.org dataset.

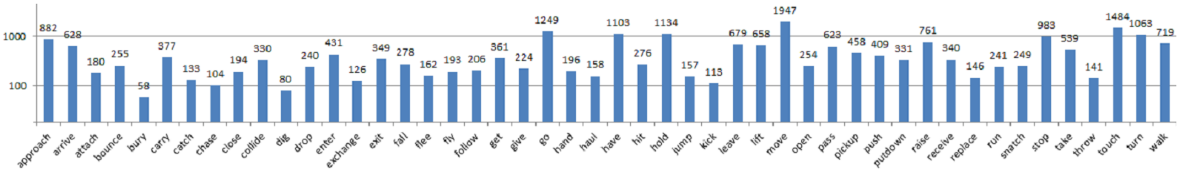

Fig. 2. The 48 human actions in the visint.org dataset and their (logarithmic) prevalence in the train set.

We propose a method for the human-action recognition application that gives a major contribution in the performance on this visint.org dataset with a gain factor of 2.37 relative to previously published results on this dataset [9]. The reason for this major improvement is the exploitation of the correlations between the 48 actions and it is related to a different approach on three themes:

- Better selection of negative training samples, by choosing those that are most similar to positives.
- Exploitation of the posterior probabilities of all action detectors in a two-stage SVM classifier.
- Combination of multiple features that are based on low level motion and high level object features.

The outline of this paper is as follows. Our method will be motivated and detailed in Section 2, where we also compare to related work. The experimental setup is defined in Section 3. The experiments and results on the visint.org dataset are shown in Section 4, and a comparison to state-of-the-art results on the IXMAS and UT-Interaction datasets is provided in Section 5. A discussion of the results is provided in Section 6, and Section 7 concludes the paper with our main findings.

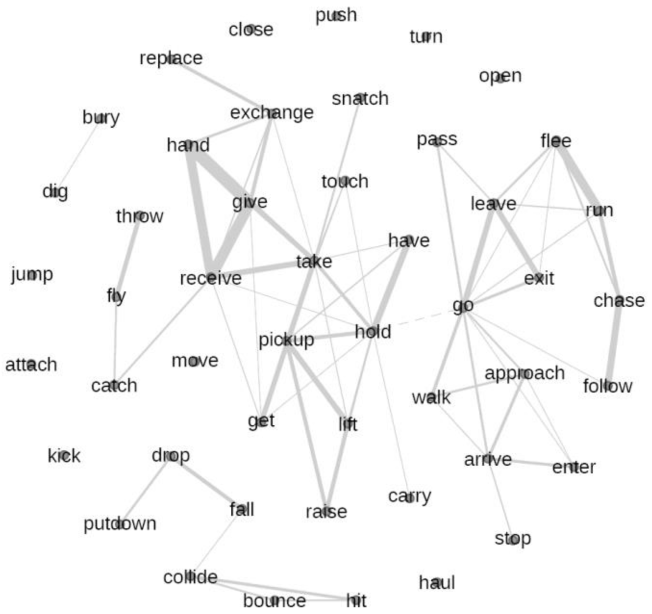

Fig. 3. Correlations of human action annotations in the visint.org dataset. Only correlations larger than 0.2 are shown. All large correlations are positive. The thickest line indicates a correlation of 0.7.

2. Method

2.1 Overview of the action recognition framework

The basic action recognition framework is similar to the bag-of-words model used in a recent performance evaluation [10]. The general scheme is that features are computed, quantized by a codebook, stored in histograms, and finally the action is detected by an action-specific classifier. More specifically, in [10], space-time interest points (STIP) features were used, quantized by a k-means codebook, and videos were classified by a SVM. Due to its simplicity and reasonable performance in various detector-feature configurations on the KTH dataset and also on the more challenging UCF Sports and Hollywood2 datasets [10], we select this framework as a baseline pipeline to which we can add our proposed improvements. We replace the k-means codebook by a random forest as a quantizer [11], because it worked better in our experiments (data not shown). Further we consider two types of features: the low-level STIP features and high-level event property (EP) features (Sec. 2.2), which we both compute on the data In our method and experimentation, we will compare and combine them.

The basic pipeline and the advanced pipeline that we use in our experiments are shown in Figure 4. Each of the components in the advanced pipeline will be described and compared to the basic pipeline in the following subsections: feature computation (Sec. 2.2); sampling for, and creation of, the random forest (Sec. 2.3); the classification, by a single classifier (one stage) and a two-stage classifier (Sec. 2.4); and finally combining features (Sec. 2.5).

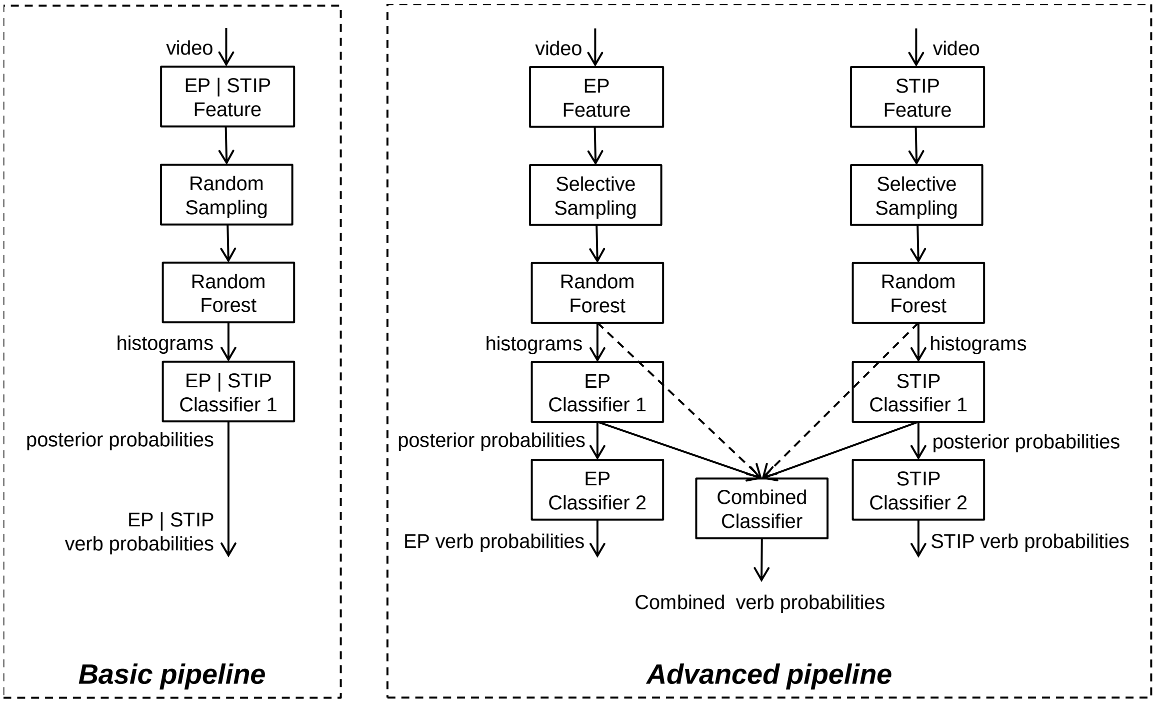

Fig. 4. Overview of the basic pipeline and the proposed advanced framework that is used in our experiments. In the basic pipeline either EP or STIP features are used, indicated by EP | STIP, where in the advanced pipeline the EP and STIP features are combined on feature or on classifier level.

In our ICPR '12 paper [12], we performed initial experiments by comparing a STIP-based two-stage classifier to its one-stage counterpart. In this paper, we extend this setup to include also a different video feature, i.e. the complementary EP feature. Thereby we generalize to the advanced pipeline shown in Figure 4, extending it further by exploring two schemes to combine the two features. In this paper, we show experimentally the impact of the several choices inside the pipelines. We demonstrate how an optimized pipeline design provides a major improvement for the detection of 48 human actions.

2.2 Feature computation

Many interesting features have been proposed for action recognition, which can be distinguished into two types of features: low-level motion features and high-level object features.

The low-level features aim at capturing properties of the actions such as motion. A well-known example is the motion template [13], which captures the motion of the full body. A drawback is that it depends on the silhouette, whose detection is known to be error-prone. Local motions have been studied extensively, e.g. the representation of actions in terms of a PCA-basis of optical flow [14]. The segmentation is still a critical element here, which has been a motivation to consider motion in several feature point configurations. That proved to be a promising direction due to its success for action recognition, see e.g. [15]. Several motion-based descriptors have been proposed since. The space-time interest point (STIP) detector and descriptor [16] is an example, capturing the local motions and shapes around detected Harris points in (x,y,t) space. It is very popular, because it has proven to be a very discriminative and robust feature for actions [17].

The high-level features are a different category and include, among others, detection of persons in the image and estimation of their poses. Poses are interesting because they relate very directly to human actions, as opposed to the low-level features such as STIP. Pose estimation has been done by matching exemplars [18] and searching the image for poses by evaluating deformable part models in spatial windows [19]. The window-based approach has also been successfully applied to the detection of persons, which potentially can be applied as a first step before analyzing the pose. The representation of a person's trajectory has enabled storyline generation [20]. In [21] the layout of the scene is exploited to interpret trajectories and actions. The Event Property (EP) features as proposed in [9] aim to combine the ideas that were proposed in previous work. Persons are detected by a dedicated detector [22] where other objects are detected by a generic moving object detector [23]. All entities are tracked with a standard search-and-update tracker, and described by 7 canonical poses that are classified from outputs of [19], and by additional event properties such as "two entities approach each other".

Our choice for the low-level feature is STIP due to its robustness, no need for segmentation, and its proven value for action recognition. Our choice for the high-level features is EP, because it combines several methods that each have proven their merit for representing meaningful intermediate-level aspects of human actions. In Sec. 2.5, we will show how these two can be combined.

We note that there are other more specific features that can also be used for action recognition, such as frequency domain based features [45] that are informative of (quasi) periodic motion patterns such as walking and digging. It is our goal to include features that are common to a wide range of human activities. Therefore, we choose not to include such more specific features as they are representative of a subset of activities only.

2.3 Sampling and the random forest

During the training phase, two components are learned for an action detector: the random forest to create bag-of-feature histograms [11] and a classifier to detect the presence of actions. For the classifiers considered this learning is no problem: after the random forest the amount of data is limited and it is capable of learning from unbalanced classes. However, for the learning of the random forest, both the size of the data and the unbalanced classes are an issue [24].

Sampling is needed to reduce the amount of data and to obtain a proper balance in the training set between positive and negative examples. On the visint.org dataset, on average we have 500 STIP and EP feature vectors per video, resulting in approximately 2M feature vectors. Considering all these feature vectors in intractable: in recent applications of random forests typically 500K feature vectors are used for learning, see e.g. [25]. The other issue is the unbalanced presence of actions in this dataset. An extreme example is "bury" being present in 2% of the samples only. In fact, only 1 action, "move" occurs in more than 50% of the clips. For all other actions, we are faced with the challenge of selecting proper negative samples for training.

It has been shown previously that significant improvements can be achieved by doing such sampling carefully [26]. In previous work, the rationale has been proposed to select negatives that are similar to positives, social tags, i.e. free textual strings, have been exploited in [27]. However, social tags are unbound as a user can input any textual string. That makes the search for good negatives hard and therefore an iterative scheme is applied to find a good subset of negatives. In our case, the tags are bound: the annotations are fixed, as 48 binary judgements have been made by the annotators whether each action is present in a video. This opens up the opportunity to exploit the other 47 tags on whether other actions are present/absent in order to select the negatives for the current action.

Each sample consist of one video clip, which typically has several action labels, and hundreds of feature vectors. For the positive class of a particular human action, the positive videos can be sampled randomly to obtain enough positive feature vectors. The most common approach to obtain the negatives is to do the same. However, the set of negative samples, is much more heterogeneous: these can consist of any combination of the other 47 human actions. Therefore, we propose a selective sampling scheme. For discriminative learning, the rationale is that good negative samples are similar to the positive samples. To select negatives that are similar to positives, we consider all remaining 47 binary judgements by the annotators whether each action is present. Due to the correlations between actions, we expect to be able to determine the similarity between videos based on correspondences between the remaining 47 annotations. Similarity is determined by comparing the 47 annotations based on the L^2 distance metric, where we normalize these vectors before comparison, such that they sum to one. The random forest quantizes the equally balanced positive and negative feature vectors into histograms (or visual words).

Furthermore, we investigate for each action which type of sampling (random or selective) works best. Selecting the best sampling strategy for each action allows further optimization of the performance.

2.4 Two-stage classification

Several classifiers have been used in previous work to estimate action probabilities, such as the Tag Propagation (TP) [9] or the Support Vector Machine (SVM), e.g. [10]. For classification we chose an SVM, which is currently the most popular classifier in bag-of-feature models due to its robustness against large feature vectors and sparse labels (see ‘Classifier’ in Figure 4). In preliminary experiments, the SVM proved indeed to be better than standard alternatives, e.g. K nearest neighbors, nearest mean, logistic linear, quadratic and Fisher classifiers (data not shown).

In this paper, we additionally investigate the merit of a second-stage classifier, where all 48 posterior probabilities of the first classifier bank are used as new features for a second stage. Both the first and the second stage consist of 48 single action classifiers. The rationale behind using a second stage is that actions are correlated (Figure 3), so they are informative of each other. For instance, the likelihood of walking is very predictive of moving. This is a source of information that we want to exploit, by a two-stage setup. In Figure 4 this is depicted by the ‘Classifier-2’ which is applied to ‘Classifier-1’ outputs. Note that the basic pipeline (Figure 4, left) does not exhibit this property.

In machine learning, this is commonly known as Sequential Stacked Learning (SSL), see e.g. [49]. In video concept detection, others have investigated methods to combine class-specific detectors in a second-stage classification. For instance, in the TRECVID challenge, the posterior probabilities of many concept detectors were taken as feature values and fed to a higher-level classifier [28]. Another technique for the second-stage analysis is a weighted linear combination of all posteriors, which yielded a better estimate of each of the classes in [29]. Typically, such techniques have been used to combine different feature modalities, like image and text features [30], and different concepts [28]. Very recently, for action detection, first-stage action detectors were used as an ‘action bank’ to represent activities in video and to be classified by a second-stage classifier [7]. In a first stage, action detectors were applied to the video frames, and in the second stage, the posteriors were max-pooled over volumes in the video and represented by a histogram. This advanced scheme involves sampling over viewpoints and scales, dividing the video into volumes, and accumulating the evidence from the first-stage detectors into a well-engineered histogram representation. This two-stage scheme proved to be very effective on multiple datasets, yet also computationally demanding. The authors indicate that on average 204 minutes are consumed to process a video of the UCF50 dataset, with a maximum of 34 hours [7]. Given that these are short video clips, we consider such a method currently computationally intractable for detection of action in thousands of realistic videos, which is the case that is investigated in this paper.

In this paper, we answer the question how much performance can be gained by exploiting the information in the correlated actions. To that end, we want to assess the immediate advantage of exploiting the information in the first-stage detectors. Hence we use their continuous posterior probabilities directly without complex and computationally demanding operations that are needed to make advanced representations such as the action bank [7], although we acknowledge the potential that can be gained by such advanced representations. The added value of our paper is that we assess the merit of combining multiple action detectors in a simple two-stage setup, where each detector has been composed of the standard state-of-the-art components STIP features, a random forest, and a SVM (see Sec. 2.1) that have proven to perform well (see e.g. [1,5,10,17]) and are very efficient (STIPs run at approximately 10 fps and the random forest quantization and SVM classification are real-time). Combining the first-stage detector outputs by a weighted linear combiner as used in [29] proved not to be effective for our purpose (data not shown). Rather we consider the simple and computationally efficient scheme where the posterior probabilities from stage one are used as features in stage 2.

2.5 Combining multiple features

In the last step of the pipeline, we improve the action detection by combining low-level (STIP) and high-level (EP) features. The most straightforward approach is to directly concatenate the EP and STIP histograms and feed them in a single classifier and create a detector for each action (dashed arrows in Figure 4). As an alternative, we also consider the concatenation of the posteriors of the first classifiers for both EP and STIP features and the creation of a final-stage combined detector for each action. Other combiner schemes are available, such as multiple kernel learning [31], but we will show that this simple two-stage combiner is effectively taking advantage of the complementarity both feature types.

3. Experimental Setup

3.1 Videos and Annotations

The visint.org dataset [8] includes 48 human actions in a train set of 3,480 videos of 10-30 seconds, and a test set of 1,294 similar video. This dataset is novel and contributed by the DARPA Mind’s Eye program. The annotation is as follows: for each of the 48 human actions, a human has assessed whether the action is present in each video or not (“Is action X present?”). Typically, multiple actions are reported for every video, on average the number of reported actions is seven.

3.2 Implementation details of the framework

For each action we create a random forest [32] with 10 trees and 32 leafs, based on 200K feature vectors, 100K from randomly selected positive videos, and 100K from either randomly selected or selective sampled negative videos. The random forest quantizes the features into histograms [11] and a SVM classifier with a χ^2 kernel [33] is trained that serves as a detector for each action. For the random forest we use Breiman and Cutler's implementation [32], with the *M*-parameter equal to the total number of features (162 for STIP and 78 for EP features). For the SVM we use the libSVM implementation [34], where the χ^2 kernel is normalized by the mean distance across the full training set [33], with the SVM's slack parameter default *C=1*. The weight of the positive class (i.e. the samples of a particular action) is set to *(#pos+#neg)/#pos* and the weight of the negative class (i.e. samples that do not contain the action) to *(#pos+#neg)/#neg*, where *#pos* and *#neg* are the amount of positive and negative class samples [35].

3.3 Performance Measure

Although detection of actions will simultaneously involve spatial localization in the video, the performance of action detection will not be measured in terms of localization accuracy in this paper. The reason for this is that not all datasets contain annotations of action location. The performance will be measured by the MCC measure,

$$MCC = \frac{TP \cdot TN - FP \cdot FN}{\sqrt{(TP+FP)(TP+FN)(TN+FP)(TN+FN)}}$$

with T=true, F=false, P=positive and N=negative. The MCC measure has the advantage of its independence of the sizes of the positive and negative classes. This is important for our evaluation purpose, as the human actions are highly unbalanced (see Figure 2). This evaluation metric was also used in the evaluation of the DARPA's Mind's Eye program and it allows comparison to previous results on this dataset [9,12]. In this paper we report the average and the standard deviation of the MCC per verbs (denoted as 0.25±0.13) where in [9] we reported the MCC calculated using TP,TN,FP and FN averaged over all verbs.

4. Experiments and results

The parts of the pipeline that we vary are: the sampling of feature vectors for the random forest (Sec. 4.1), the usage of two-stage classification (Sec. 4.2), the combination of STIP and EP features in a joint pipeline (Sec. 4.3), and analysis of the best approach for each action (Sec. 4.4). The other parts remain fixed during the experiments. Optimization of the parameters, training of the random forest and classifiers, and selection of the best sampling approach for each action has been performed on the training set. The results in these sections are based on applying the trained method to the test set. A combined overview of these results is given in Section 4.5.

4.1 Random and selective sampling

Table 1 shows the results of random vs. selective sampling, for both EP and STIP features separately and one-stage classification. For completeness, we also inserted the state-of-the-art results [9] on the visint.org dataset (experiment number 00).

We see that for EP-only results, the baseline [9] with tag propagation outperforms the one-stage SVM based results. We consider the SVM because it proved to work better than tag propagation for all other cases in this paper (data not shown). Table 1 shows that STIP features work much better than EP features. For EP features, the selective sampling gives an overall relative performance loss of -17% compared to random sampling. However, for 15 out of 48 actions the performance is improved. For STIP features, selective sampling of negatives yields an improvement for 21 out of 48 actions. Note however, that we intend to select the best sampling scheme per action. If we consider the best sampling scheme per action and compare to random sampling, EP performance increases with 29% and STIP performance with 19%. Compared to the baseline [9], STIP features with best sampling give an improvement of 73%.

Table 1. Different sampling strategies, compared to the baseline from [9] (ExpNr 00).

ExpNr	Feature	Classifier	Sampling	Classification	MCC	Improvement
00	EP	TP	Random	1 stage	0.11±0.10	-
01	EP	SVM	Random	1 stage	0.07±0.11	-36%
02	EP	SVM	Selective	1 stage	0.06±0.09	-45%
03	EP	SVM	Best	1 stage	0.09±0.11	-18%
04	STIP	SVM	Random	1 stage	0.16±0.15	+45%

05	STIP	SVM	Selective	1 stage	0.18±0.15	+64%
06	STIP	SVM	Best	1 stage	**0.19±0.15**	+73%

4.2 One- and two-stage classification

Table 2 shows the results of one- vs. two-stage classification, for both EP and STIP features separately and best sampling. For comparison, we again inserted the best sampling one-stage results of the previous subsection (experiment number 03 and 06). The two-stage classifier improves the results for both EP and STIP features for more than 20%. For EP features the detection is improved for 30 out of 48 actions, and for STIP features an improvement is achieved for 32 out of 48 actions. Compared to the baseline of [9], a gain of 109% is achieved.

Table 2. The one- vs. two-stage classifier for both EP and STIP features separately.

ExpNr	Feature	Sampling	Classification	Combining	MCC	Improvement
03	EP	Best	1 stage	-	0.09±0.11	-18%
06	STIP	Best	1 stage	-	0.19±0.15	+73%
07	EP	Best	2 stage	-	0.13±0.09	+18%
08	STIP	Best	2 stage	-	**0.23±0.13**	+109%

4.3 Combining multiple features

We consider two combination schemes and compare them in the experiments. One combination is to concatenate the EP and STIP histograms and feed them in a one-stage combining classifier and create a detector for each action. The other combination is to concatenate the EP and STIP posteriors and create a second-stage combining classifier for the detection of each.

Table 3 shows the results for the two schemes of combining EP and STIP features. Compared to concatenation of the two features' histograms, the detection improvement achieved by combining the posteriors of both advanced pipelines is almost doubled. Note that the one-stage combination of features are worse than the one-stage classifier with STIP only (experiment number 06). Relative to the performance of combining posteriors of STIP only (experiment number 08), the performance increase is 7% (from 0.23 to 0.25). The posterior combination improves detection accuracy for 42 out of 48 human actions, and the relative improvement of the second-stage combiner is 39% with respect to the concatenated histograms, and 127% with respect to [9].

Table 3. The results for the two schemes of combining EP and STIP features, which combine either histograms in the first stage or posteriors in the second stage.

ExpNr	Feature	Sampling	Combining	MCC	Improvement

09	EP+STIP	Best	Histograms	0.18±0.14	+64%
10	EP+STIP	Best	Posteriors	**0.25±0.13**	+127%

4.4 Analysis of best feature, sampling and combiner

Analysis of the best feature (EP and/or STIP), sampling (random or selective) and combiner (one or two stage) for each action separately leads to the conclusion that 73% would be based on the EP+STIP two-stage combiner. Table 4 summarizes how often a particular scheme of the best feature, sampling and combiner is optimal. Based on these results, we can also conclude that STIP features by themselves are optimal in 17% of the cases, and that the EP features by themselves are almost never the best (2% of the overall cases). Interestingly, the combination of STIP and EP features add discriminative power to the overall solution in 73% of the cases. A surprise is that for two-stage classification, the combination of STIP and EP features are always better than either one of them alone in a two-stage classifier. Yet, the added value of EP features is limited: 7% (from 0.23 to 0.25, see experiment numbers 08 and 10, and details in Sec. 4.3).

Table 4. The number of the 48 human actions for which a particular scheme of “feature, sampling, combiner” performs best. (*) Only available for best sampling per action.

Feature(s)	Random sampling	Selective sampling
STIP: one-stage classifier	4/48 = 8%	4/48 = 8%
EP: one-stage classifier	1/48 = 2%	0/48 = 0%
STIP + EP: histograms combiner	1/48 = 2%	3/48 = 6%
STIP + EP: posteriors combiner	35/48 = 73% *	

If we select for each action the best feature, sampling and combiner, it would improve the results from experiment number 10. Figure 5 displays the results for this optimal scheme. The detection accuracy is on average for all 48 human actions: MCC = 0.26±0.13, which is slightly better than the two-stage combiner. Compared to the STIP features with random sampling and one-stage classification, a gain of 63% is achieved, and compared to the baseline [9], a major gain of 137% (gain factor 2.37) is achieved.

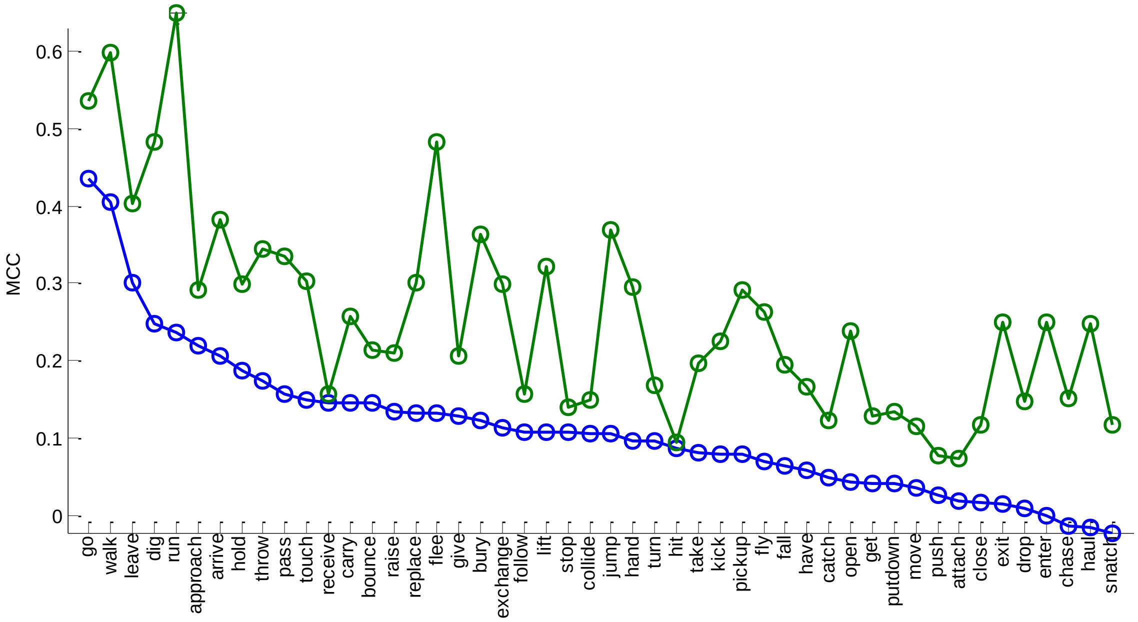

Fig. 5. The optimal scheme of "Best feature, sampling and combiner" per action (green, MCC=0.26±0.13), compared to the baseline on this dataset [9] (blue, MCC=0.11±0.10): the relative improvement overall is a gain factor of 2.37 compared to the baseline.

Examples of human actions that are detected well by our method (MCC > 0.4) are: Go, Walk, Leave, Dig, Run, Flee. These actions have limited intra-class variation and are well-defined in terms of their motion. Actions that perform reasonably well (MCC > 0.25) are e.g.: Approach, Arrive, Hold, Throw, Pass, Touch, Carry, Replace, etc. Here, some of the actions are quite complex, involving a person interacting with small items, such as to Hold and Carry something. We consider this result impressive. Examples of actions that do not perform well (MCC < 0.15) are: Stop, Collide, Hit, Catch, Get, Move, Push, Attach. These actions are hard due to the following reasons: subtle motion (Hit), very short duration (Catch), and badly annotated (Move) Overall, the results are promising, although we acknowledge that for an actual application the performance might need to be improved further.

Practical usage of automatic recognition for a specific action may require a minimum performance for that particular action. Table 5 summarizes the results by listing per MCC score the percentage of the 48 human actions for which such detection accuracy is achieved. Of all 48 human actions, 6% has a strong performance, MCC > 0.5, and almost half of the actions, 44% has a reasonable performance, MCC > 0.25.

Table 5. The cumulative histogram of the 48 human actions for which a level of MCC scores is achieved.

MCC	> 0.50	> 0.45	> 0.40	> 0.35	> 0.30	> 0.25	> 0.20	> 0.15	> 0.10	> 0.05
Actions	6%	10%	13%	19%	27%	44%	60%	75%	94%	100%

4.5 Combined overview of results

For all previous experiments (ExpNr 01 – 10) we display the performance by one precision-recall curve. A precision-recall curve provides an intuitive visualization of action detection performance for various thresholds of the detector [44]. Recall that the single MCC reported per experiment was computed by averaging the MCC per action. We follow a similar procedure for the precision-recall curve. For each action, we obtain a precision-recall curve. The single curves per experiment in Figure 6 are calculated by averaging the curves for all 48 actions. For readability of the figure, we have omitted the standard deviations from the points on the curve.

The ordering of curves up to the upper- right point, the ideal operating point is at (1,1), follows the ordering we found when comparing the average MCC across the 48 actions. Compared to ExpNr 04, a standard STIP-based action detector with random sampling and one-stage classification (i.e., a setup often applied by the action recognition research community), the combination of features with best of random/selective sampling and two-stage classification achieves approximately a higher recall of 0.10 – 0.15 absolute gain at the same precision.

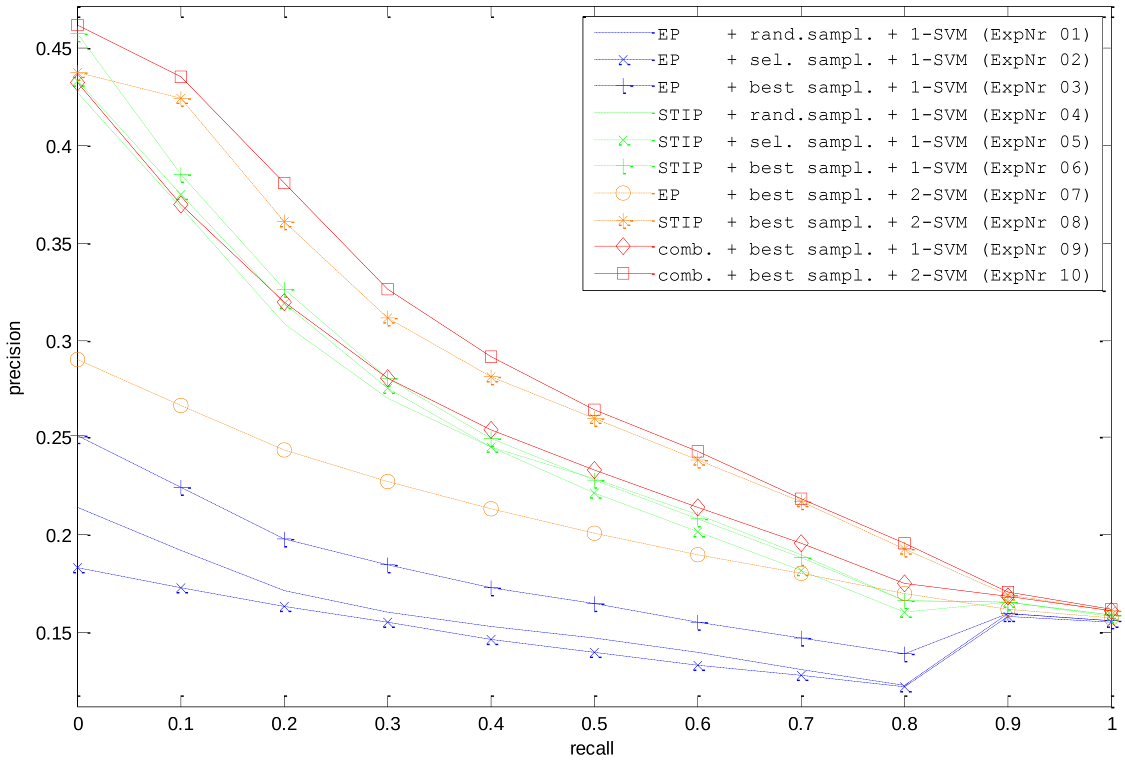

Fig. 6. Precision-recall curves corresponding to the visint.org experiments.

5. Comparison to the State-of-the-Art

In this section, we compare our approach to state-of-the-art methods on commonly used datasets. To the best of our knowledge, beside our earlier work [9,12] no results have been published on the visint.org dataset yet; it has been released very recently. Therefore, we will compare to the state-of-the-art on the single-person multi-viewpoint IXMAS dataset of subtle actions [39], and the UT-Interaction dataset [40] containing two-person interactions. For both datasets we will investigate the added value of the selective sampling for the random forest and the two-stage classification setup. We do this for the STIP feature only; due to their performance and also as we expect the EP features do not map straightforward to other data sets. STIP features are easily computed and Section 4 showed that they achieve near optimal results compared to the multi-feature setup.

5.1 Experimental setup

For the experiments on the IXMAS and UT-Interaction we follow an identical procedure. For each action we learn a random forest in two modes: random and selective. Random mode means that we learn the forest by labeling features from one action *A* and randomly selected features from all other actions. In selective mode we consider features from one action *A* and features from one other action *B* that is most similar. Contrary to visint.org with multiple labels per clip, we do not have metadata (like the correlations) to infer which action is most similar or related as IXMAS and UT-Interaction have only a single label per clip. Therefore we follow a slightly different approach for IXMAS and UT-Interaction. To infer which other action is most similar to *A*, we use a cross-validation setup to select the action *B* that yields the forest with most discriminative power in terms of classification accuracy. In this way, we can sample selectively and this will demonstrate to be very powerful.

For the first-stage action detectors, we obtain for each action a detector by training an SVM in an identical setup to Section 4. For the one-stage setup we classify the test sample as the single action that maximizes the a-posteriori probability. We also investigated the potential of the two-stage setup. Here we train a second-stage SVM on the posterior probabilities of the first-stage SVMs identical to Section 4. All random forests and SVMs are trained separately for each action in each dataset, and for each camera viewpoint in IXMAS. We report on the classification accuracy

$$accuracy = \frac{TP + TN}{TP + FP + TN + FN}$$

averaged over all actions, similar to results previously have been reported in literature.

5.2 IXMAS dataset

The IXMAS dataset [39] consists of 12 complete action classes with each action executed three times by 12 subjects and recorded by five cameras with the frame size of 390 × 291 pixels. These actions are: check watch, cross arms, scratch head, sit down, get up, turn around, walk, wave, punch, kick, point and pick up. The body position and orientation are freely decided by different subjects. The standard setup on this dataset is leave - one subject - out cross validation setting. We compare against the state-of-the-art result of [41], who achieved a 78.0% recognition accuracy across the five cameras using the Multiple-Kernel Learning with Augmented Features (AFMKL) method.

We tested the performance of each sampling strategy and the one-stage and two-stage classification setup. The results are provided in Table 6. The best setting is the selective sampling with one classification stage. The improved sampling yields a much more distinctive description of each action. The two-stage setup is no advantage here, we expect that this is due to the uncorrelated actions: waving is not informative of turning around, etc. The actions in IXMAS are much more dissimilar than in the visint.org dataset, and therefore they are not predictive of each other. The best performance that we obtain is 87.3%, thereby outperforming [41] by 9.3% (absolute), see Table 6.

Table 6. Mean accuracy on the IXMAS dataset [39] of the state-of-the-art (AFMKL [41]) method on this dataset and of our method for the random vs. selective sampling strategy and the one-stage and two-stage classification setup.

			Camera Viewpoint					
	Sampling	Stages	1	2	3	4	5	Average (%)
AFMKL [41]	-	-	81.9	80.1	77.1	77.6	73.4	78.0
Ours	Random	One	72.6	73.6	72.3	75.6	68.7	72.6
	Selective	One	85.3	88.4	87.5	87.3	88.0	**87.3**
	Random	Two	71.0	74.6	69.6	78.5	69.1	72.6
	Selective	Two	82.6	87.7	86.4	87.5	86.2	86.1

5.3 UT-Interaction dataset

We used the segmented version of the UT-Interaction dataset [40] containing videos of six human activities: hand-shaking, hugging, kicking, pointing, punching, and pushing. The UT-Interaction dataset is a public video dataset containing high-level human activities of multiple actors. We consider the #1 set of this dataset, following the setup in [42]. The #1 set contains a total of 60 videos of six types of human-human interactions. Each set is composed of 10 sequences, and each sequence contains one execution per activity. The videos involve camera jitter and/or background movements (e.g., trees). Several pedestrians are present in the videos as well, making the recognition harder. Following [42], we consider the leave-one-sequence-out cross validation, performing a 10-fold cross validation. That is, for each round, the videos in one sequence were selected for testing, and videos in the other sequences were used for the training. We compare against three well-performing methods, [40,42,43]. In [42] a Hough-voting scheme is proposed. In [43] a dynamical bag-of-words model is proposed, which together with the cuboid + SVM setup in [40] are very similar to our STIP single stage setup, being also bag-of-words methods. More advanced methods using local features have been proposed recently. For instance, in [48] the spatio-temporal layout has been taken into account. In [47] the local features do not vote individually, but feature voting using random projection trees (RPT) is performed. RPTs demonstrated to leverage the low-dimension manifold structure and they proved to be very discriminative for action recognition.
We tested the performance of each sampling strategy and the one-stage and two-stage classification setup. The results are provided in Table 7. The results show clearly that for various settings almost the same performance is achieved, all with accuracies similar to state-of-the-art. With the one-stage classification setup, the selective sampling does not

add discriminative power. We think that this is due to the distinctive appearance of the actions, which make the forests obtained by random sampling very discriminative already. The performance with random sampling and the one-stage classification setup is slightly better than other bag-of-words methods [40,43]. We hypothesize that the small improvement over those methods is due to usage of the random forest in our method, which has proven to be generally more distinctive than k-means as used in [40,43]. The best setting is the selective sampling with two-stage classification, resulting in a slight improvement over the state-of-the-art. Contrary to IXMAS, for UT-Interaction the two-stage setup provides a slight advantage here. The best performance that we obtain is 88.3%, thereby achieving a similar performance to [42]. Our result is not as good as the state-of-the-art on UT-Interaction, being [47,48]. These methods have successfully exploited additional properties of the data, resp. the low-dimensional manifold structure and the spatio-temporal layout. For the standard bag-of-features model, we have shown that an advantage can be obtained by the two-stage setup in combination with the selective sampling.

Table 7. Mean accuracy on the UT-Interaction dataset [40] of the state-of-the-art methods on this dataset and of our method.

Method	Sampling	Stages	Accuracy
Yu *et al.* [47]	-	-	93.3%
Burghouts *et al.* [48]	-	-	93.3%
Waltisberg *et al.* [42]	-	-	88.0%
Ryoo [43]	-	-	85.0%
Ryoo *et al.* [40]	-	-	83.3%
Ours	Random	One	86.7%
	Selective	One	86.7%
	Random	Two	86.7%
	Selective	Two	88.3%

6. Discussion

We have performed experiments on three very different datasets: visint.org, IXMAS and UT-Interaction. The visint.org dataset has the most samples, with the broadest set of actions, ranging from single-person actions both simple (e.g., jump) and complex (e.g., open an item), to multi-entity actions such as two persons exchanging an item. This is the only dataset which contains multiple actions per sample. The IXMAS is the second largest dataset and contains 12 single-person actions from multiple viewpoints, where the actions include very detailed motions (e.g., check watch). The UT-Interaction is the smallest dataset, and it contains 6 two-person interactions, with clearly visible motion patterns. This is the reason why the results on UT-Interaction are showing saturation: many different methods perform roughly the same (see Section 5) and that also holds for the two innovations that we propose in this paper, see Table 8 where we have summarized the relative contributions of the two innovations separately and combined. The accuracy on the IXMAS dataset has been significantly improved by our method (see Section 5) and the selective sampling is the reason for the performance gain, see Table 8. For the IXMAS dataset, the two-stage classification does not add value, while for the UT-Interaction dataset, the effect of two-stage classification is very limited. This is due to the fact that both datasets contain actions that are fairly distinct (for the IXMAS more so than for UT-Interaction), so the actions are not very informative of each other. Contrary, for the visint.org dataset, some actions are very similar and they correlate significantly. For the performance on the visint.org dataset we have shown that both innovations – selective sampling and two-stage classification – are adding discriminative power to the bag-of-words method.

Table 8. Our innovations and the relative improvements on the visint.org, IXMAS and UT-Interaction datasets.

Dataset	Stages	Random sampling	Selective sampling
visint.org	One	Baseline[(a)]	+19%[(b)]
	Two	+19%[(c)]	**+44%**[(b)]
IXMAS [39]	One	baseline	**+20%**
	Two	0%	+19%
UT-Interaction [40]	One	baseline	0%
	Two	0%	**+2%**

(a) For visint.org, this table includes the results using STIP features as they are most commonly used by the research community.

(b) For visint.org, this number reflects the best of random/selective sampling per action, because for that dataset the optimal setting was different for each action.

(c) This number is based on an experiment that was not reported in Section 4.

7. Conclusions

For human action detection, we have considered the visint.org dataset of 48 actions in 3,480 train and 1,294 test videos of 10-30 seconds each, ranging from simple actions such as walk to complex actions such as exchange. We have effectively merged high-level Event Property (EP) features and low-level space-time interest point (STIP) features, which capture complementary properties about human behavior. We have used a pipeline of visual processing leading to a sparse feature representation, a random forest to quantize the features into histograms, and an SVM classifier.

We have shown that our optimal advance pipeline results in a performance gain factor of 2.37 compared to the earlier baseline. The major improvement is related to a different approach on three themes: sample selection, two-stage classification, and the combination of multiple features. First, we have shown that the sampling of training data for the random forest can be improved by smart selection of the negatives. Second, we have shown that exploiting all 48 actions' posteriors by two-stage classification greatly improves its detection. Third, we have shown how the low-level motion STIP and high-level object EP features should be combined in a two-stage process.

Furthermore, we see that for EP-only results, the earlier baseline [9] outperforms the one-stage SVM based results. STIP features are clearly more discriminative than the EP features, but the posterior STIP and EP combination based configuration outperforms the STIP-only classifiers. Although the difference to the final two-stage combiner solution is quite small, as expected a per-action selection of the best classifier provides the best results.

Finally, we have compared our method to the existing state-of-the-art on the IXMAS and UT-Interaction datasets, and have shown that selective sampling for the random forest and two-stage classification improves the recognition results. For the UT-Interaction dataset, we show that selective sampling and the two-stage setup improve on standard bag-of-feature methods. On the IXMAS dataset, we outperform the state-of-the-art by 9.3% (was 78.0%, ours 87.3%).

Acknowledgements

This work is supported by DARPA (Mind’s Eye program). The content of the information does not necessarily reflect the position or the policy of the US Government, and no official endorsement should be inferred.

The authors acknowledge the CORTEX scientists for their significant contributions to the overall system: S.P. van den Broek, P. Hanckmann, J-W Marck, L. de Penning, J-M ten Hove, S. Landsmeer, C. van Leeuwen, A. Halma, M. Kruithof, S. Korzec, W. Ledegang and R. Wijn. Figure 3 has been contributed by E. Boertjes.

References

[1] C. Schuldt, I. Laptev, B. Caputo, Recognizing Human Actions: A Local SVM Approach, ICPR (2004)

[2] L. Gorelick, M. Blank, E. Shechtmanm, M. Irani, R. Basri, Actions as Space-Time Shapes, PAMI, 29 (12) (2007)

[3] T. Guha, R.K.Ward, Learning Sparse Representations for Human Action Recognition, PAMI, 34(8) (2012)

[4] S. Ali, M. Shah, Floor Fields for Tracking in High Density Crowd Scenes, ECCV (2008)

[5] M. Marszalek, I. Laptev, C. Schmid, Actions in Context, CVPR (2009)

[6] J. Liu, J. Luo, M. Shah, Recognizing Realistic Actions from Videos "in the Wild", CVPR (2009).

[7] S. Sadanand, J. J. Corso, Action bank: A high-level representation of activity in video. CVPR (2012)

[8] http://www.visint.org/datasets.html

[9] H. Bouma, P. Hanckmann, J-W. Marck, L. de Penning, R. den Hollander, J-M. ten Hove, S. P. van den Broek, K. Schutte, G. J. Burghouts, Automatic human action recognition in a scene from visual inputs, Proc. SPIE 8388 (2012)

[10] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, C. Schmid, Evaluation of local spatio-temporal features for action recognition, BMVC (2009)
[11] F. Moosmann, B. Triggs, F. Jurie, Randomized Clustering Forests for Building Fast and Discriminative Visual Vocabularies, NIPS (2006)
[12] G.J. Burghouts, K. Schutte, Correlations between 48 Human Actions Improve Their Detection, ICPR (2012)
[13] A. Bobick, J. Davis, The recognition of human movement using temporal templates, PAMI, 23(3):257–267 (2001)
[14] M. Black, Y. Yacoob, A. Jepson, D. Fleet, Learning Parameterized Models of Image Motion, CVPR (1997)
[15] A. A. Efros, A. C. Berg, G. Mori, J. Malik, Recognizing Action at a Distance, ICCV (2003)
[16] I. Laptev, On Space-Time Interest Points, IJCV, 64 (2/3) (2005)
[17] I. Laptev, M. Marszalek, C. Schmid, B. Rozenfeld, Learning Realistic Human Actions from Movies, CVPR (2008)
[18] G. Mori, S. Belongie, J. Malik, Efficient Shape Matching Using Shape Contexts, PAMI, 27(11):1832-1837 (2005)
[19] D. Ramanan, Learning to parse images of articulated bodies, NIPS (2006)
[20] A. Gupta, P. Srinivasan, J. Shi, L. S. Davis, Understanding Videos, Constructing Plots: Learning a Visually Grounded Storyline Model from Annotated Videos, CVPR (2009)
[21] N. Ikizler-Cinbis, S. Sclaroff, Object, Scene and Actions: Combining Multiple Features for Human Action Recognition, ECCV (2010)
[22] P. Felzenszwalb, R. Girshick, D. McAllester, D. Ramanan, Object detection with discriminatively trained part based models, PAMI, 32(9) (2010)
[23] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, CVPR (1999)
[24] N. Japkowicz, S. Stephen, The class imbalance problem: A systematic study, Intelligent Data Analysis, 6 (5) (2002)
[25]S J. R .R. Uijlings, A. W. M. Smeulders, R. J. H. Scha, The Visual Extent of an Object – Suppose We Know the Object Locations, IJCV (2012)
[26] A. K. Jain, R. P. W. Duin, J. Mao, Statistical Pattern Recognition: A Review, PAMI, 22 (1) (2000)
[27] X. Li, C. G. M. Snoek, M. Worring, A. W. M. Smeulders, Social Negative Bootstrapping for Visual Categorization, ICMR (2011)
[28] G. Iyengar, H. Nock, and C. Neti, Discriminative model fusion for semantic concept detection and annotation in video, ACM Multimedia (2003)
[29] Y. Aytar, O. B. Orhan, M. Shah, Improving Semantic Concept Detection and Retrieval using Contextual Estimates, ICME (2007)
[30] R. Yan. M. Naphade, Semi-supervised cross feature learning for semantic concept detection in videos, CVPR (2005)
[31] A. Vedaldi, V. Gulshan, M. Varma, A. Zisserman, Multiple Kernels for Object Detection, ICCV (2009)
[32] L. Breiman, Random forests, Machine Learning, 45 (1) (2001)
[33] J. Zhang, M. Marszalek, S. Lazebnik, C. Schmid, Local features and kernels for classification of texture and object categories: A comprehensive study, IJCV, 73 (2) (2007)
[34] C.-C. Chang and C.-J. Lin, LIBSVM: a library for support vector machines, software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)
[35] K. E. A. van de Sande, T. Gevers, C. G. M. Snoek, Evaluating Color Descriptors for Object and Scene Recognition, PAMI, 32 (9) (2010)
[36] H. Liu, R. Feris, M.T. Sun, Benchmarking human activity recognition, CVPR Tutorial, CVPR (2012)
[37] J. Yuan, Z. Liu and Y. Wu, Discriminative subvolume search for efficient action detection, CVPR (2009)
[38] J.C. Niebles, C. Chen and L. Fei-Fei, Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification. ECCV (2010)
[39] D. Weinland, E. Boyer, R. Ronfard, Action Recognition from Arbitrary Views using 3D Exemplars, ICCV (2007)
[40] M. S. Ryoo, C.-C. Chen, J. K. Aggarwal, A. Roy-Chowdhury, An Overview of Contest on Semantic Description of Human Activities, ICPR (2010)

[41] X. Wu, D. Xu, L. Duan, J. Luo, Action Recognition using Context and Appearance Distribution Features, CVPR (2011)
[42] D. Waltisberg, A. Yao, J. Gall, L. Van Gool, Variations of a Hough-voting action recognition system, ICPR (2010)
[43] M. S. Ryoo, Human Activity Prediction: Early Recognition of Ongoing Activities from Streaming Videos, ICCV (2011)
[44] J. Yuan, Z. Liu, Y. Wu, Discriminative Video Pattern Search for Efficient Action Detection, PAMI, 33 (9) (2011)
[45] G.J. Burghouts, J.M. Geusebroek, Quasi-periodic Spatio-Temporal Filtering, IEEE Trans. Image Proc., 15 (6), (2006)
[46] Y.G. Jiang, G. Ye, S. Chang, D. Ellis, A.C. Loui, Consumer video understanding: A benchmark database and an evaluation of human and machine performance, ACM Int. Conf. Multimedia Retrieval ICMR, (2011).
[47] G. Yu, J. Yuan, Z. Liu, Propagative Hough Voting for Human Activity Recognition, ECCV (2012).
[48] G.J. Burghouts, K. Schutte, Spatio-Temporal Layout of Human Actions for Improved Bag-of-Words Action Detection, Pattern Recognition Letters (2013).
[49] E. Puertas, S. Escalera, O. Pujol, Multi-class Multi-scale Stacked Sequential Learning, Multiple Classifier Systems, pp. 197-206, 2011.

G. J. Burghouts is a lead research scientist in visual pattern recognition at the Intelligent Imaging research group at TNO (the Netherlands). He studied artificial intelligence at the University of Twente (2002) and received his PhD degree from the University of Amsterdam in 2007 on the topic of visual recognition. He is the principal investigator of the Cortex project within the DARPA Mind's Eye program, about recognition of events and behaviors, resulting in a prominent achievement during the September 2011 evaluation trials. He won an award from the Netherlands Association ofEngineers for the 'best innovative project' in 2007. He has published papers in IEEE Transactions on Image Processing, Computer Vision and Image Understanding, International Journal of Computer Vision, IEEE Transactions on Systems, Man and Cybernetics, Machine Vision and Applications, and Pattern Recognition Letters.

K. Schutte received his PhD degree from the University of Twente, Enschede, the Netherlands, in 1994. He had a postdoctoral position with the Delft University of Technology's Pattern Recognition (now Quantitative Imaging) group. Since 1996, he has been employed by TNO, currently as lead research scientist of intelligent imaging. Within TNO he has actively led multiple projects in areas of signal and image processing and scene understanding. Recently, he has led many projects, for both international industries and governments, resulting in image processing products in active service. His research interests include behavior recognition, pattern recognition, sensor fusion, image analysis, and image restoration.

H. Bouma is a research scientist and project manager at TNO in the field of computer vision and pattern recognition. He received his MSc in Electrical Engineering from the University of Twente and his PhD in Biomedical Image Analysis from the TU Eindhoven, the Netherlands. Dr. Bouma participated in projects about ship recognition for the Dutch Ministry of Defence, in the Mind's Eye program of DARPA for human action and behaviour recognition, and he leaded several projects about the automatic tracking and behavior analysis in surveillance video for the Dutch Police.

R. J. M. den Hollander received his MSc and PhD degrees in Electrical Engineering from the Delft University of Technology in 2001 and 2007, respectively. Since 2006, he has been a research scientist at TNO, the Netherlands, where he works on the development of image processing algorithms for various vision tasks. His research interests include computer vision and pattern recognition.