Action Detection with Actom Sequence Models

Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid

Action Detection with Actom Sequence Models

Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid

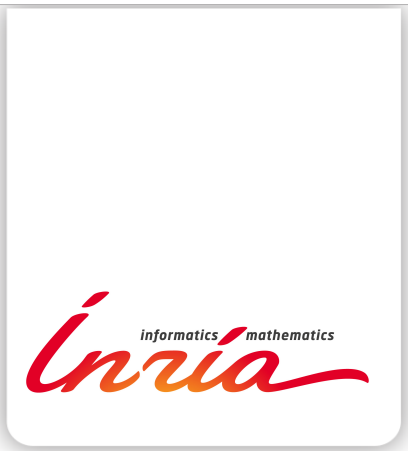
RESEARCH REPORT

N° 7930

April 2012

Project-Teams LEAR and MSR-INRIA

ISSN 0249-6399 ISRN INRIA/RR--7930--FR+ENG



Action Detection with Actom Sequence Models

Adrien Gaidon*†, Zaid Harchaoui†, Cordelia Schmid†

Project-Teams LEAR and MSR-INRIA

Research Report n° 7930 — April 2012 — 29 pages

Abstract: We address the problem of detecting actions, such as drinking or opening a door, in hours of challenging video data. We propose a model based on a sequence of atomic action units, termed "actoms", that are semantically meaningful and characteristic for the action. Our Actom Sequence Model (ASM) represents the temporal structure of actions as a sequence of histograms of actom-anchored visual features. Our representation, which can be seen as a temporally structured extension of the bag-of-features, is flexible, sparse, and discriminative. Training requires the annotation of actoms for action examples. At test time, actoms are detected automatically based on a non-parametric model of the distribution of actoms, which also acts as a prior on an action's temporal structure. We present experimental results on two recent benchmarks for temporal action detection: "Coffee and Cigarettes" and the "DLSBP" dataset. We also adapt our approach to a classification by detection set-up and demonstrate its applicability on the challenging "Hollywood 2" dataset. We show that our ASM method outperforms the current state of the art in temporal action detection, as well as baselines that detect actions with a sliding window method combined with bag-of-features.

Key-words: Action recognition, Video analysis, Temporal detection

* MSR-INRIA joint center
† LEAR team, INRIA Grenoble

RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Détection d'Actions à l'aide de Séquences d'Actoms

Résumé : Cet article s'intéresse au problème de la détection temporelle d'actions — comme "ouvrir une porte" — dans des bases de données contenant des heures de vidéo. Nous proposons un modèle basé sur des suites d'actions atomiques, appelées "actoms". Ces actoms sont des sous-événements interprétables qui caractérisent l'action à modéliser. Notre modèle, nommé "Actom Sequence Model" (ASM), décrit la structure temporelle d'une action par le biais d'une suite d'histogrammes de descripteurs locaux localisés temporellement. Cette représentation est une extension flexible, parcimonieuse, discriminative et structurée du populaire "sac de mots visuels". La période d'apprentissage nécessite l'annotation manuelle d'actoms, sans que cela ne soit requis à l'étape de détection. En effet, les actoms de nouvelles vidéos sont automatiquement détectés à l'aide d'un modèle non-paramétrique de la structure temporelle d'une action, estimé à partir des exemples d'apprentissage. Nous présentons des résultats expérimentaux sur deux bases de données récentes pour la détection temporelle d'actions: "Coffee and Cigarettes" et "DLSBP". De plus, nous adaptons notre approche au problème de classification par détection et démontrons ses performances sur la base "Hollywood 2". Nos résultats montrent que l'utilisation d'ASM améliore les performances par rapport à l'état de l'art et par rapport à l'approche par fenêtre glissante avec sac de mots, couramment utilisée en détection.

Mots-clés : Reconnaissance d'actions, Analyse de vidéos, Détection Temporelle

Contents

Figure 1: Examples of actom annotations for two actions.

1 Introduction

The automatic understanding of video content is a challenging problem of critical importance. In particular, action recognition is an active topic in the computer vision community (*c.f.* [1, 2, 3] for three recent surveys). Although early experiments in simple video conditions have shown promising results [4, 5], recent efforts have tried to address more challenging video sources like movies [6]. For such realistic datasets, many state-of-the-art action models are based on the popular bag-of-features (BOF) representation [4, 6, 7, 8, 9, 10, 11]. A BOF representation suffers from the following limitations:

1. **orderless models**: BOF aggregates local features over the entire video volume and ignores the temporal ordering of the frames;

2. **segmented test videos**: test videos are assumed to be strictly containing the action, *i.e.* presented in the same fashion as the training examples.

In this paper, we propose a sequential action model that enforces a soft ordering between meaningful temporal parts. In addition, we provide an algorithm to learn the global temporal structure of actions, which allows for efficient *temporal action detection* — *i.e.* finding *if and when* an action is performed in a database of long and *unsegmented* videos. In particular, we focus on searching for actions of a few seconds, like sitting down, in several hours of real-world video data.

Discriminative models are of particular importance in the context of detecting short actions, where searching through a large volume of data can result in many false alarms. Our approach is based on the observation that a large number of actions can be naturally defined in terms of a composition of simpler temporal parts. For instance, Figure 1 illustrates that the displayed actions are easy to recognize given a short sequential description. Obtaining such a decomposition is challenging and its components are clearly action-specific. In this work, we propose to model an action as a small sequence of key atomic action units, which we refer to as *actoms*. These action atoms are semantically meaningful temporal parts, whose sequence is characteristic of the action. They are an intermediate layer between *motion primitives*, like spatio-temporal interest points, and *actions*.

Figure 2: Actom frames of detected test sequences.

We make the following contributions. First, in Section 3, we introduce a temporally structured representation of actions in videos, called Actom Sequence Model (ASM). Our representation encodes the temporal ordering constraints between actoms in a flexible way by concatenating per-part histograms. Furthermore, the robustness of ASM allows it to model actions with only approximately ordered or partially concurrent sub-events. Composed of local video features, actoms are specific to each action class and obtained by manual annotation, though only at training time. These annotations have the same cost as specifying start and end frames of actions, while being richer and more consistent across action instances.

Second, in Section 4, we propose a simple, yet efficient algorithm to learn the likely temporal structures of an action. We introduce a non-parametric generative model of inter-actom spacings and show how we include negative training examples in order to learn an ASM classifier.

In Section 5, we describe how we perform temporal action detection using a sliding central frame approach, which we, then, extend to a classification by detection scenario. Note, that in addition to detecting actions, our approach can return the most likely actoms of detected actions, as illustrated in Figure 2.

Finally, in Section 6, we investigate the importance of the components of our method and show that it outperforms the state of the art on two recent benchmarks for action detection: the "Coffee and Cigarettes" [12] and "DLSBP" [9] datasets. In addition, we demonstrate the applicability of our approach in a classification by detection setup on a larger set of actions from the Hollywood 2 dataset [13].

2 Related work

Incorporating temporal information in action models is critical for the understanding of human motion. In the following, we review action recognition approaches that model temporal aspects of videos and that are used for action detection: sequential approaches, volumetric approaches, and representations based on local features.

2.1 Sequential approaches

Based on the observation that digital videos are successions of frames, sequential approaches represent actions as sequences of states such as poses. There are two main groups of methods: sequential probabilistic models and exemplar-based techniques.

Sequential probabilistic models. Inspired from speech recognition, several works [14,15,16,17] use dynamic probabilistic graphical models — *e.g.* Hidden Markov Models (HMM) [18] — to learn temporal transitions between hidden states. An action is represented by a generative model over sequences of per-frame feature vectors. Recognition is performed based on the likelihood of an image sequence with respect to this model. For instance, Yamato *et al.* [14] recognize tennis actions using HMMs.

More recently, several approaches [19,20,21,22,23,24,25,26] attempt to recognize actions in more challenging settings. In general, they use a Viterbi-like inference algorithm to temporally segment videos. For instance, dealing with long-term interactions with objects, Laxton *et al.* [20] use Dynamic Bayesian Networks (DBN) with manually designed per-activity hierarchies of predefined contextual cues and object detectors. Some of theses approaches [25,26] are related to HMMs but learn a discriminative model from manually segmented training videos to perform action segmentation.

Exemplar-based approaches. Also operating on sequence representations of actions, exemplar-based methods [27,28,29,30,31] rely on directly comparing an action with template sequences by computing an alignment score. In general, they require less training data and provide more flexibility as they can handle non-linear speed variations by using Dynamic Time Warping (DTW) [32]. For instance, Darrell and Pentland [27] perform gesture recognition using DTW on a sequence of scores obtained by correlating a video with a set of per-view templates. More recently, Brendel and Todorovic [31] recognize actions as time series of a few snapshots of body parts obtained by video segmentation. They use a template-based classification approach by aligning a video with training examples using DTW.

Limitations. Sequential approaches cannot represent actions involving concurrent sub-events. Some extensions of HMMs address this issue — *e.g.* coupled HMMs [19] or more generic Dynamic Bayesian Networks [20] — but their complex, domain-specific structure needs to be manually specified by experts.

Additionally, sequential recognition methods often rely on representations requiring body-part estimation or background subtraction, which are difficult to obtain when faced with moving cameras, lighting changes and poor video quality.

Finally, although these methods are adapted to the classification of video clips, their application to temporal action detection is not straightforward. Detection can be performed by using higher-level probabilistic models, for instance by combining multiple HMMs — *c.f.* for instance [21,33] — but they require a large amount of training examples in order to model all events that might occur, including non-actions.

2.2 Volumetric approaches

In contrast to sequential approaches, volumetric methods view actions as 3D (X-Y-T) objects in a spatio-temporal video volume, thus treating space and time in a unified manner. These template-based approaches are successful on simple datasets with controlled video conditions [4, 34].

Space-time volume models. Some models operate directly on the videos themselves. For instance, Kim and Cipolla [35] directly compare video volumes using Tensor Canonical Correlation Analysis. Alternatively, several approaches [5, 34, 36, 37] rely on silhouettes in order to obtain spatio-temporal templates. For instance, Bobick and Davis [36] introduce motion history images (MHI), which are temporal templates representing the evolution of motion over time. Silhouettes provide useful information for action recognition, but their use is limited to simple or controlled video conditions. They are, indeed, difficult to obtain in the presence of complex dynamic backgrounds, and do not account for self-occlusions.

Other approaches [38, 39, 40, 41, 42] focus on optical flow information to obtain action templates. For instance, Efros *et al.* [40] compute blurred optical flow features inside tracks of soccer players. More recently, Ke *et al.* [42] over-segment videos and use optical flow and volumetric features to match space-time shapes.

Detection in video volumes. As volumetric approaches rely on a similarity measure between video volumes, they typically detect actions by matching sub-volumes with a set of candidate templates. For instance, Ke *et al.* [42] use a sliding-window approach with part-based template matching using pictorial structures. Note, that the sequential approaches can also be applied in a similar sliding window manner, such as in [27] with DTW and in [43] with HMMs.

Limitations. These methods require spatio-temporally aligned videos. Hence, they are not robust to occlusions, partial observations, and significant viewpoint and duration variations. They often require human localization, which is still an unsolved problem in real-world videos. As they are based on spatio-temporal shape or flow templates, volumetric approaches assume the contiguity of the video volume spanned by an action. Consequently, they are not adapted to actions with interruptions or with multiple interacting actors, such as kissing.

2.3 Local-features-based approaches

Due to the development of action databases from movies [6, 12, 13], TV shows [44, 45], Youtube clips [46, 47] or sports broadcasts [37], action recognition has recently focused on more challenging sources of videos. Methods based on local spatio-temporal features [4, 6, 7, 8, 11, 12, 13, 46, 48, 49, 50, 51, 52, 53] have demonstrated competitive performance on these datasets (*c.f.* [54] for a recent evaluation). Inspired by the recent progress of object recognition in 2D images, local-features-based approaches for action recognition represent videos as collections of local X-Y-T patches, such as spatio-temporal interest points [50]. As they make no assumptions about the global structure of actions and rely on powerful descriptors such as HOG [55], local features yield representations that are, in general, more robust than sequential or volumetric ones.

There are two main families of detection techniques based on local features: local classifiers, which deal with features individually, and global approaches, which aggregate features over video sub-volumes, *e.g.* using a bag-of-features model.

Local classifiers. Amongst methods processing each feature individually, voting-based approaches [53, 56, 57, 58] aim at measuring the importance of individual features for a particular action. They detect actions by extracting local features, each of which casts a vote for a particular action. For instance, Yuan *et al.* [58] detect spatio-temporal interest points which cast votes based on their point-wise mutual information with the action category. They, then, use a spatio-temporal branch and bound algorithm to efficiently localize actions. Other approaches [12, 46, 59]

aim at selecting the most relevant local features. For instance, Nowozin *et al.* [59] use a sequence of individual local features assigned to a fixed number of uniformly sized temporal bins.

Selecting local features has the advantage of being able to efficiently localize the action in space-time. Voting approaches also allow for the detection of multiple co-occurring actions and simple multi-actor interactions. However, they often assume that each local feature can provide enough information to recognize an action. Therefore, they are not discriminative enough to differentiate between complex actions sharing common motion or appearance primitives.

Bag-of-features. An alternative family of models uses the global distribution of features over a video volume. One of the most common and efficient representation is the bag-of-features (BOF) [4,6,7,8,11]. Inspired from text document classification [60,61], a vocabulary of prototype features — called "visual words" — is obtained by clustering, and a video is holistically represented as histograms of occurrences of local features quantized over this visual word vocabulary. Statistical learning methods like Support Vector Machines (SVM) [62] can then be applied to learn a BOF classifier. Detection is then generally performed by applying this classifier in a sliding window manner. Though powerful, this model discards the temporal information inherent to actions and is, thus, not well adapted to discriminate between actions distinguished by their structure, *e.g.* opening and closing a door, which can result in numerous high score false alarms during detection.

Our approach is based on BOF and detects actions with a sliding central frame technique. The most similar approaches are [9, 10], which rely on multi-scale heuristics with manually defined window sizes. In contrast, our algorithm leverages a learned generative model of an action's temporal structure. Furthermore, we improve upon existing temporally structured extensions of BOF. Laptev *et al.* [6] combine multiple BOF models extracted for different rigid spatio-temporal grids that are manually selected. Multiple coarse grids are combined in a multi-channel Gaussian kernel. Such an approach is shown to slightly improve over the standard BOF, but the temporal structure of actions is fixed and not explicitly modeled. On the contrary, we learn a temporal structure that is adapted to the action and show that, compared to a rigid grid, this results in significant gains in detection performance.

Related to our work, Niebles *et al.* [63] discover motion parts based on a latent model [64]. They learn a SVM classifier per motion segment at fixed temporal locations, whereas we do not rely on an intermediate recognition step and use our temporal decomposition to classify actions. In addition, they use a loose hierarchical structure that is tailored to the classification of long duration activities — *e.g.* "triple-jump" — but not well adapted to short actions, as illustrated by their results.

Finally, our method is similar in spirit to state-of-the-art approaches for facial expression recognition from videos. Facial expression recognition can be performed using label information defined by the Facial Action Coding System (FACS) [65], which segments facial expressions into predefined "action units", complemented with temporal annotations such as *onset*, *peak*, *offset*. Most approaches, however, only use peak frames for classification [66], except *e.g.* [67]. Furthermore, as the complexity of generic human actions makes the construction of universal action units impractical, we investigate user-annotated, action-specific training actoms.

A preliminary version of this work appeared in [68].

3 Actions as sequences of actoms

An action is decomposed into a few, temporally ordered, category-specific *actoms*. An actom is a short atomic action identified by its central temporal location, around which discriminative visual information is present. It is represented by a temporally weighted aggregation of local features, which are described in Section 3.1. We model an action as a sequence of actoms by concatenating the per-actom representations in temporal order — *c.f.* Figure 3 for an illustration. We refer to our sparse sequential model as the Actom Sequence Model (ASM) and define it in Section 3.2. We describe the process used to acquire training actom annotations in Section 3.3.

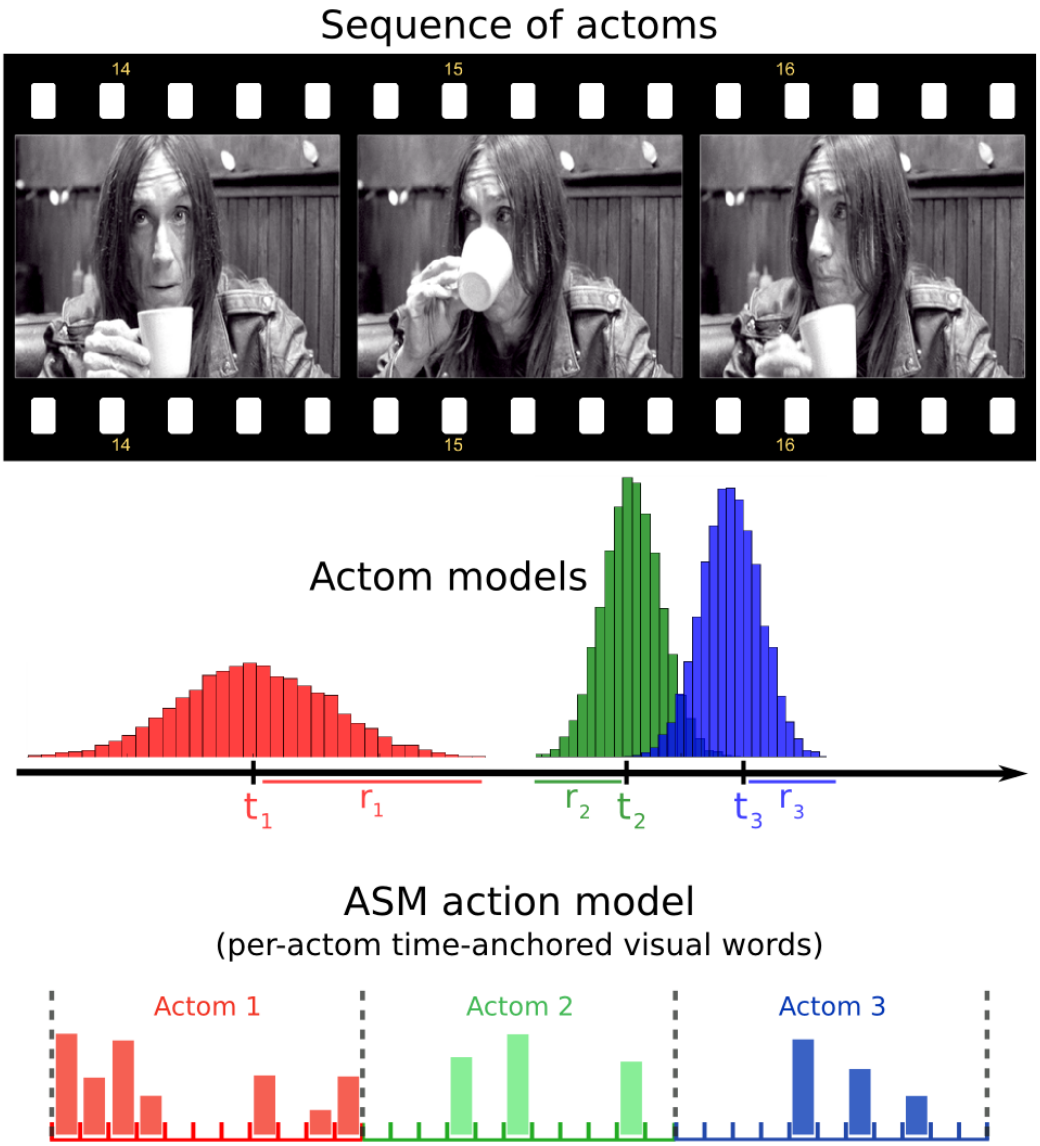


Figure 3: Construction of our ASM action model using actom-based annotations and a temporal weighting scheme for aggregating local features in a sparse temporally structured bag-of-features.

3.1 Local visual information in actoms

Following recent results on action recognition in challenging video conditions [6, 9], we extract sparse space-time features [50]. We use a multi-scale space-time extension of the Harris operator to detect spatio-temporal interest points (STIPs). They are represented with a concatenation of histograms of oriented gradient (HOG) and optical flow (HOF). We use the original STIP implementation available on-line [69].

Once a set of local features has been extracted, we quantize them using a visual vocabulary of size v. In our experiments, we cluster a subset of 10^6 features, randomly sampled from the training videos. Similar to [9], we use the k-means algorithm with a number of clusters set to $v = 1000$ for our detection experiments, while, similar to [54], we use $v = 4000$ for our classification-by-detection experiments. We then assign each feature to the closest visual word.

3.2 The Actom Sequence Model

We define the time-span of an actom with a *radius* around its temporal location. We propose an adaptive radius that depends on the relative position of the actom in the video sequence. The adaptive radius r_i, for the actom at temporal location t_i, in the sequence of a actom locations $(t_1, \cdots, t_a)$, is parametrized by the amount of overlap ρ between adjacent actoms:

$$r_i = \frac{\delta_i}{2 - \rho} \tag{1}$$

where ρ ranges between 0 and 1 and δ_i is the distance to the closest actom:

$$\delta_i = \begin{cases} t_2 - t_1 & \text{if} \quad i = 1 \\ t_a - t_{a-1} & \text{if} \quad i = a \\ \min(t_i - t_{i-1}, t_{i+1} - t_i) & \text{if} \quad 1 < i < a \end{cases}$$

This defines a symmetric neighborhood around the temporal location specific to each actom of an action. Visual features are computed only within the forward and backward time range defined by the actom's radius. They are accumulated in per-actom histograms of visual words.

Our model only assumes a weak temporal ordering of the actoms. In addition, defining the actom's time-span relatively to its closest neighbour has multiple advantages. On the one hand, it allows adjacent actoms to overlap and share features, while enforcing a soft temporal ordering. This makes the model robust to inaccurate temporal actom localizations and to partial orderings between concurrent sub-events. On the other hand, it also allows for gaps between actoms and can, therefore, represent discontinuous actions. Furthermore, an adaptive time-span makes the model naturally robust to variable action duration and speed.

We also introduce a *temporally weighted assignment scheme*. We propose to aggregate temporally weighted contributions of per-actom features. Each feature at temporal location t in the vicinity of the $i-th$ actom, *i.e.* if $|t - t_i| \leq r_i$, is weighted by its temporal distance to the actom:

$$w_i(t) = \frac{1}{\sigma\sqrt{2\pi}} \exp\left(-\frac{(t - t_i)^2}{2\sigma^2}\right) \tag{2}$$

Hence, features further from an actom's center vote with a smaller importance. This scheme offers an intuitive way to tune the bandwidth σ of the weighting window using the Chebyshev inequality. For a random variable X of mean μ and finite standard deviation σ, we know that $\mathbf{P}(|X - \mu| \geq k\sigma) \leq 1/k^2$, for any $k > 0$. Rewriting this equation with $X = t$, $\mu = t_i$ and $r_i = k\sigma$, we obtain:

$$\mathbf{P}(|t - t_i| < r_i) \leq p, \quad p = 1 - \frac{\sigma^2}{r_i^2} \tag{3}$$

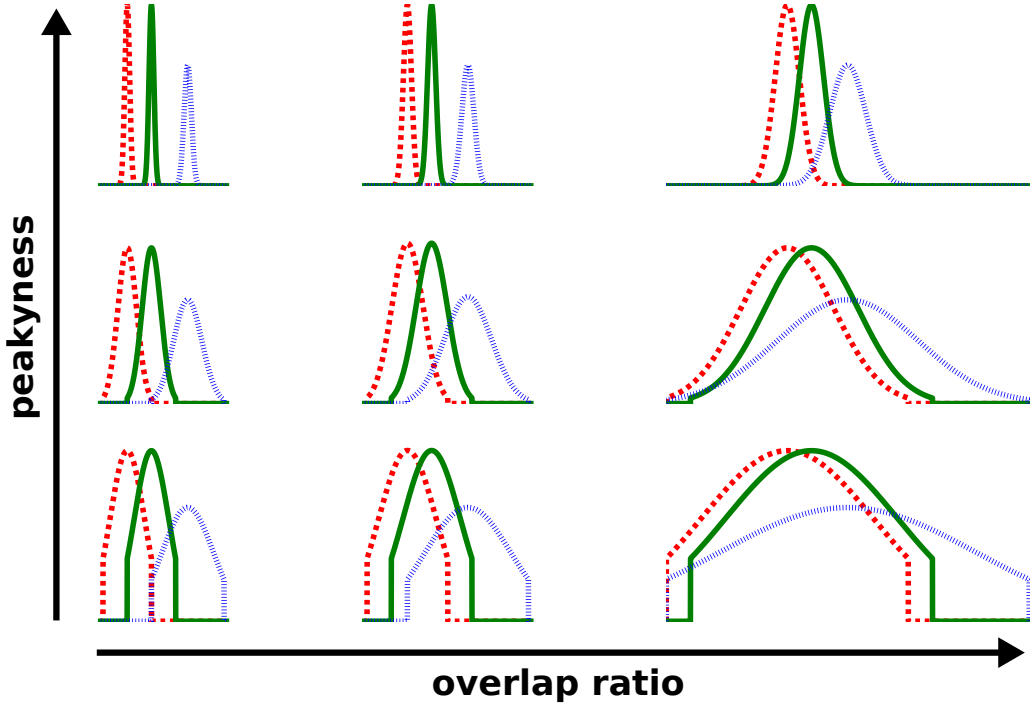


Figure 4: Illustration of ASM for the same actom locations but different ASM parameters. This shows the influence of the overlap ρ and peakyness p parameters on the per-frame weights. It can be observed that ASM encompasses a continuum of models, ranging from sequences of "soft" key-frames (upper left) to BOF-like models (lower right).

The probability p is the "peakyness" of our soft-assignment scheme and replaces σ as parameter of our model. It allows to encode a prior on the amount of probability mass of features falling in an actom's time range. See Figure 4 for an illustration of the influence of the overlap ρ and peakyness p on the actom-specific per-frame weights.

In our experiments, we set ρ and p parameters per-class by maximizing detection performance on a held out, unsegmented, validation video. We found that this hyper-parameter optimization scheme yielded better results than cross-validation on the segmented training clips. Cross-validation, indeed, only maximizes window classification performance, which is an easier problem than detection.

To summarize, we derive our ASM model from a sequence of a actom locations by (i) computing visual features only in the actoms's time-spans, which are parametrized by the ρ parameter (Eq. 1), (ii) computing the feature contributions to per-actom temporally weighted histograms (Eq. 2), and (iii) appending these histograms into a temporally ordered sequence which is our ASM representation of videos (cf. Figure 3): $\mathbf{x} = (x_{1,1}, \cdots, x_{1,v}, \cdots, x_{a,1}, \cdots, x_{a,v})$, where

$$x_{i,j} = \sum_{t=t_i-r_i}^{t_i+r_i} w_i(t) c_j(t) \tag{4}$$

is the weighted sum of the number $c_j(t)$ of local features at frame t assigned to visual word j, over the $i-th$ actom's time-span $[t_i - r_i, t_i + r_i]$. The ASM vector $\mathbf{x}$ is then L_1-normalized.

3.3 Actom annotations

In this section, we present how to obtain actom annotations and the advantages of this supervision over annotating beginning and ending frames of actions.

Obtaining annotations. An actom annotation is a time stamp in the corresponding video. This temporal location is selected such that its neighboring frames contain visual information that is representative of a part of the action. The number of actoms is fixed depending on the action category. We observed that only a few actoms are necessary to unambiguously recognize an action from their sequence (*c.f.* examples in Figure 1). We use three actoms per action example in our experiments. Note, that only positive training examples are manually annotated. In general, this corresponds to a small fraction of the training data — most action recognition benchmarks use in the order of 100 action examples per category. The initial noisy set of candidate training clips can be automatically obtained by using external data, *e.g.* with simple textual queries on movie transcripts [44].

During the annotation process, semantic consistency in the choice of actoms across different video clips is necessary: the $i-th$ actom of an action should have a single interpretation, like "recipient containing liquid coming into contact with lips" for the drinking action. This is ensured by giving precise guidelines to annotators and by making multiple annotators label or correct each example. Note, however, that our approach is robust to the violation of this assumption, *i.e.* we can still model actions, even when an actom has a few possible meanings across training examples.

After all the training examples are annotated once, we perform a simple outlier detection step using the temporal structure model described in Section 4.2. First, we learn a model of the temporal structure and estimate the likelihood of each annotation according to this model. We, then, resubmit for annotation the inconsistently annotated examples, *i.e.* those below a likelihood threshold (we use 2%). After these samples are re-annotated, we update the model of the temporal structure and re-estimate the likelihood of each annotation. We iterate this process up to three times.

Practical observations. Consistent actom annotations are easier to obtain than precise action boundaries. For instance, it is unclear whether a person walking towards a door before opening it is a part of the action "Open Door". In contrast, the time at which the door opens can be unambiguously determined.

Duchenne *et al.* [9] and Satkin and Hebert [10] observed that temporal boundaries of actions are not precisely defined in practice. Furthermore, they show that inaccurate boundary annotations significantly degrade the recognition performance. Therefore, they propose to improve the quality of annotated action clips by automatically cropping their temporal boundaries. However, they only model the temporal extent of actions, not their temporal structure.

On the contrary, actom annotations are well defined as a few frames of precise atomic events. Consequently, annotating actoms leads to smaller annotation variability. Figure 5 quantitatively illustrates this claim. It shows that the ground truth annotations for the action "sitting down" have a smaller duration variance when actoms are annotated instead of beginning and ending frames. We also observed that the average annotation time per action is comparable for both of these annotation types — we measured between 10 and 30 seconds per action.

In addition, an actom is a visual phenomenon that is deemed semantically relevant by annotators, and not an automatically learned part of the action. It is, therefore, always possible to interpret a predicted actom sequence. Moreover, we show that it leads to discriminative representations for action recognition.

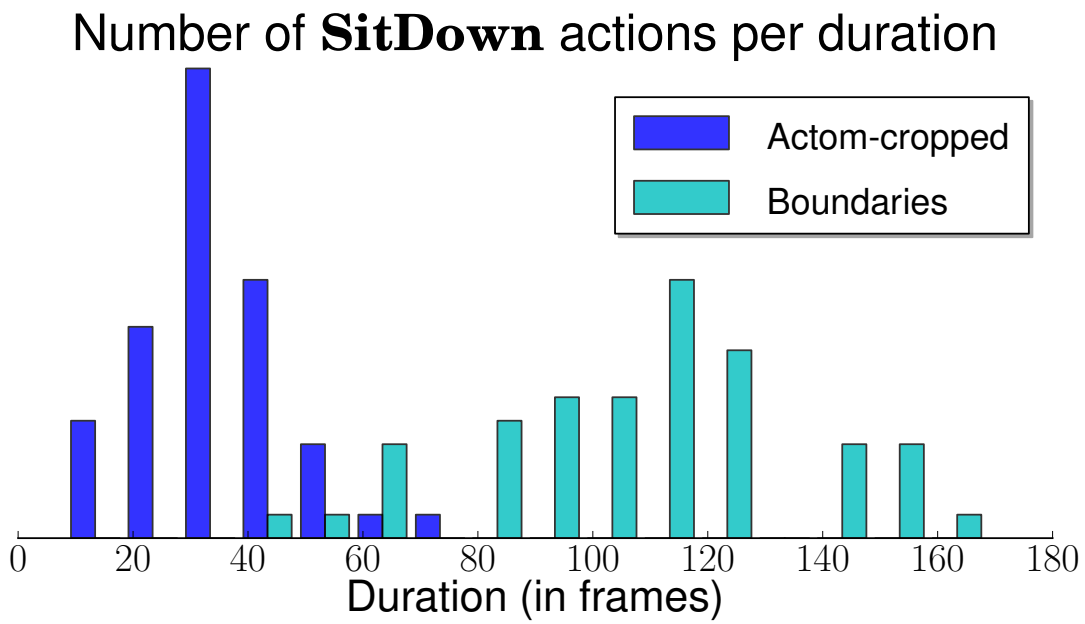


Figure 5: Frequencies of action durations obtained from manual annotations for the action "Sit Down". "Actom-cropped" represents the length of temporal windows surrounding actoms. "Boundaries" depict the duration of ground truth annotations from [9], obtained by labeling beginning and end frames of the action.

4 Temporal action detector learning

In the following, we detail the training phase of our detection algorithm. First, we give details on the action classifier operating on our ASM descriptor (Section 4.1). Then, we describe how we learn a generative model of an action's temporal structure (Section 4.2) in order to sample likely actom candidates at test time. Finally, we show how to use it to also obtain negative training examples (Section 4.3).

4.1 ASM classifier

Our detection method is similar to the sliding-window approach. It consists in applying a binary classifier at multiple temporal locations throughout the video, in order to determine the probability of the query action being performed at a particular moment. We use a Support Vector Machine (SVM) [62] trained to discriminate between the action of interest and all other visual content. As ASM is a histogram-based representation, we can use a non-linear SVM with the χ^2 or the intersection kernel [6, 70]. For efficiency reasons, we choose to use the intersection kernel [71]. It is defined for any $x = (x_1, \ldots, x_v)$ and $x' = (x'_1, \ldots, x'_v)$ as $K(x, x') = \sum_{j=1}^{v} \min(x_j, x'_j)$. Note, that, in our case, using a non-linear SVM does not prohibitively impact the detection speed, because the size of our training set is small.

In this set-up, the negative class spans all types of events except the action of interest. Therefore, more negative training examples than positive ones are necessary. We use a SVM with class-balancing [72] to account for this imbalance between the positive and negative classes.

Assume we have a set of labeled training examples $(x_1, y_1), \ldots, (x_n, y_n) \in \mathcal{X} \times \{-1, 1\}$, where $\mathcal{X}$ is the space of ASM models. Let n_+ denote the number of positive examples, n_- the number of negative examples, and $n = n_+ + n_-$ the total number of examples. The binary SVM classifier with class-balancing minimizes the regularized cost function:

$$\frac{1}{n} \sum_{i=1}^{n} L(y_i) \ell(y_i, f(x_i)) + \lambda \|w\|_{\mathcal{H}}^2 \tag{5}$$

with $f(x_i) = w^T \phi(x_i) + b$, $w \in \mathcal{H}$, $\mathcal{H}$ the feature space associated with the kernel K, $\phi : \mathcal{X} \to \mathcal{H}$ the corresponding feature map, $\ell(y, f) = \max(0, 1 - yf)$ the linear hinge loss, $L(+1) = 1/n_+$, $L(-1) = 1/n_-$, and λ a regularization parameter. In order to return probability estimates, we fit a sigmoid function to the decision function f learned by the SVM [73, 74]. Our ASM classifier evaluates the posterior probability of an action being performed, *knowing its actoms*.

4.2 Generative model of temporal structure

For unseen videos, we do not know the temporal locations of the actoms. Therefore, we learn a generative model of the temporal structure allowing to sample likely actom candidates at test time. The temporal structure we estimate is the distribution of inter-actom spacings from the training sequences: $\{\mathbf{\Delta}_i = (t_{i,2} - t_{i,1}, \ldots, t_{i,a} - t_{i,a-1}), i = 1 \ldots n_+\}$, where a is the number of actoms of the action category and n_+ is the number of positive training actom sequences.

In practice, we have only few actom annotations, typically $n_+ \leq 100$, which, in addition, can significantly differ from one another. Therefore, using histograms to model the actom spacings yields a too sparse estimate with many empty bins. Instead, we make the assumption that there is an underlying smooth distribution, which we estimate via non-parametric kernel density estimation (KDE) [75, 76]. This makes the assumption that there is a continuum of execution styles for the action of interest and it allows to correctly interpolate unseen, but likely, temporal structures.

We use KDE with Gaussian kernels whose bandwidth h is automatically set using Scott's factor [77]: $h = n_+^{-\frac{1}{a+4}}$. We obtain a continuous distribution $\mathcal{D}$ over inter-actom distances $\mathbf{\Delta} = (t_2 - t_1, \ldots, t_a - t_{a-1})$:

$$\mathcal{D} \sim \frac{1}{n_+ h^{a-1} \sqrt{2\pi}} \sum_{i=1}^{n_+} \exp\left(-\frac{||\mathbf{\Delta} - \mathbf{\Delta}_i||^2}{2h^2} \right). \tag{6}$$

As we deal with discrete time steps (frames), we discretize this distribution in the following way. First, we sample 10^4 points, randomly generated from our estimated density $\mathcal{D}$. Then, we quantize these samples by clustering them with k-means. This yields a set of s centroids $\{\hat{\mathbf{\Delta}}_j \ , \ j = 1 \cdots s\}$ and their associated Voronoi cells that partition the space of likely temporal structures. Finally, we compute histograms by counting the fraction $\hat{p}_j$ of the random samples drawn from $\mathcal{D}$ that belong to each cell j. This results in the discrete multi-variate distribution:

$$\hat{\mathcal{D}} = \{(\hat{\mathbf{\Delta}}_j, \hat{p}_j) \ , \quad j = 1 \cdots s\}, \quad \hat{p}_j = \mathbf{P}(\hat{\mathbf{\Delta}}_j). \tag{7}$$

As post-processing steps, we truncate the support of $\hat{\mathcal{D}}$ by removing structures with a probability smaller than 2% (outliers) and re-normalize the probability estimates. Figure 6 gives an example of the distribution $\hat{\mathcal{D}}$ learned for the "smoking" action.

Note, that s corresponds to the size of the support of $\hat{\mathcal{D}}$, *i.e.* the number of likely candidate actom spacings. This parameter controls a trade-off between the coarseness of the model of the temporal structure and its computational complexity. We used $s = 10$ for all actions in our experiments.

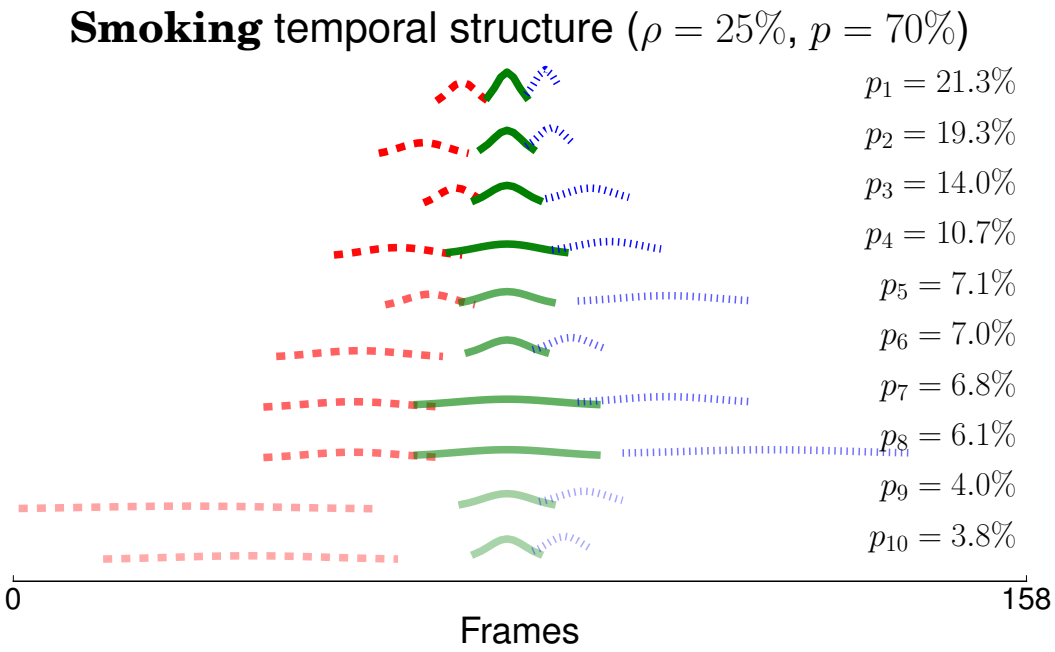


Figure 6: Temporal structure learned for "smoking" from the "Coffee & Cigarettes" dataset. The candidate actom models $(\hat{\mathbf{\Delta}}_j, p_j) \in \hat{\mathcal{D}}$ are sorted by their estimated prior probability p_j.

4.3 Negative training examples

Our detection method relies on a binary classifier discriminating between an action of interest and *all other* events. To obtain negative examples, we randomly sample clips from the unlabeled part of the training database and filter out those intersecting annotated training positives. To be consistent with the detection stage at test time, we randomly sample actoms according to the learned temporal structure $\hat{\mathcal{D}}$.

The number of random negatives needed to learn a good detector is not a priori obvious in our context, which is different from object detection. In the latter case, all objects in the training images are generally annotated. Therefore, sliding window detection methods sample as many negative windows as possible [55].

In our set-up, however, a significant part of the training videos are not annotated. Therefore, we do not know beforehand which examples are truly negative. Note, that this also rules out the possibility to mine so-called "hard negative" examples for a re-training stage [55]. Consequently, we sampled as negatives less than ten times the number of training positives — *c.f.* Section 6.6 for more details.

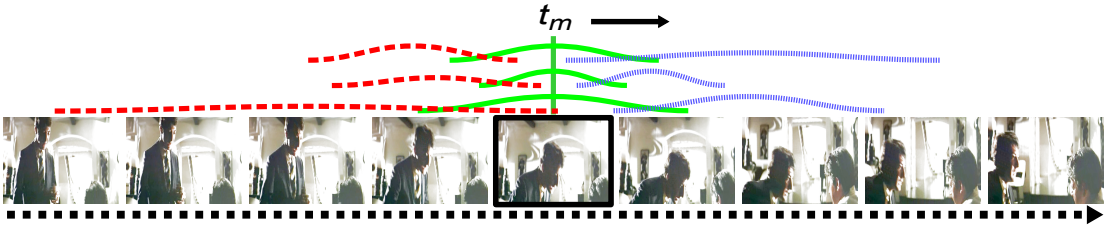


Figure 7: Sliding central frame temporal detection. The probability of an action being performed at frame t_m is evaluated by marginalizing over all actom candidates learned with our model of the temporal structure (Eq. 8).

5 Detection with actoms

In this section, we describe our temporal detection approach (Section 5.1), some post-processing steps (Section 5.2), and a strategy for action classification in approximatively pre-segmented videos (Section 5.3).

5.1 Sliding central frame detection

To detect actions in a test sequence, we apply our ASM classifier in a sliding window manner. However, instead of sliding a temporal window of fixed scale, we shift the *temporal location of the middle actom* t_m, where $m = \lfloor a/2 \rfloor$ and a is the number of actoms for the action category. We use a temporal shift of 5 frames in our experiments.

Given a central actom location t_m, we compute the probability of the action occurring at t_m by marginalizing over our generative model of actom spacings:

$$\begin{aligned} & \mathbf{P}(\text{action at } t_m) \\ = & \sum_{j=1}^{s} \mathbf{P}(\text{action at } t_m \mid \hat{\mathbf{\Delta}}_\mathbf{j})\, \mathbf{P}(\hat{\mathbf{\Delta}}_\mathbf{j}) \\ = & \sum_{j=1}^{s} f_{\text{ASM}}(\hat{t}_{j,1}, \cdots, t_m, \cdots, \hat{t}_{j,a})\, \hat{p}_j \end{aligned} \tag{8}$$

where f_{ASM} is the a posteriori probability estimate returned by our SVM classifier trained on ASM models (Eq. 5). See Figure 7 for an illustration.

Alternatively, taking the maximum a posteriori allows to not only detect an action, but also its most likely temporal structure (*c.f.* Figure 2). We have experimentally observed that, for detection, marginalizing yields more stable results than just taking the best candidate actoms. The temporal structures in $\hat{\mathcal{D}}$ are indeed related. This is a consequence of our assumption on smoothly varying styles of execution (*c.f.* Figure 6). Therefore, the redundancy in $\hat{\mathcal{D}}$ makes marginalizing over actom candidates robust to inaccurate actom placements.

Note, that the mechanism in equation 8 provides a principled solution to perform multi-scale detection, instead of the usual multi-scale sampling heuristic [12]. Both the sequential structure and the duration of the action is modeled by $\hat{\mathcal{D}}$, whereas traditional sliding-window approaches manually specify a fixed set of window sizes.

5.2 Post-processing

Our algorithm returns a probability of detection every $N-th$ frame. For video retrieval purposes, however, it is useful to return short video clips containing the queried action. Therefore, we define a detection window associated with each frame. This window has the score of its central frame

and it contains all frames used in the computation of this score. As we marginalize over $\hat{\mathcal{D}}$, this defines a single scale per action category, which only depends on the largest actom spacings in $\hat{\mathcal{D}}$. We obtain 95 frames for "drinking" and "smoking" in the "Coffee and Cigarettes" dataset, 85 frames for "opening a door" and 65 frames for "sitting down" in the "DLSBP" dataset.

In addition, as the temporal shift between two detections can be small in practice, we use a non-maxima suppression algorithm to remove overlapping detection windows. We recursively (i) find the maximum of the scores, and (ii) delete overlapping windows with lower scores. Windows are considered as overlapping if the Jaccard coefficient — the intersection over the union of the frames — is larger than 20%.

5.3 Classification by detection

Although designed for temporal detection, our method is also applicable to action classification. In both cases, the training data and learning algorithms are the same. The test data, however, differs. For detection, we process continuous streams of frames. In contrast, unseen data for classification come in the form of pre-segmented video clips.

The classification goal is to tell whether or not the action is performed in an unseen video clip, independently of *when* it is performed. Consequently, after applying our sliding central frame approach to label every $N-th$ frame of a new test clip, we pool all detection scores to provide a global decision for the clip.

In our experiments, we found that max-pooling — *i.e.* taking the best detection score as classification score — yields good results. Indeed, marginalizing over actom candidates limits the number and the score of spurious false detections, thanks to the redundancy in the learned temporal structure.

6 Experimental evaluation

This section presents experimental results comparing our ASM-based approach with BOF-based alternatives and the state of the art. In Section 6.1, we introduce the datasets used in our experiments. We, then, describe how we measure the detection performance (Section 6.2), and what baseline detection methods we compare to (Section 6.3). Our detection results are reported in Section 6.4, while our classification results are reported in Section 6.5. Finally, we quantify and discuss the influence of the parameters of our method in Section 6.6.

6.1 Datasets

We use two challenging movie datasets for action detection: the "Coffee and Cigarettes" dataset [12] and the "DLSBP" dataset [9]. We also use the "Hollywood 2" dataset [13] for our classification by detection experiments. These datasets are provided with annotations in the form of temporal boundaries around actions.

"Coffee and Cigarettes" [12]. This dataset consists of a single movie composed of 11 short stories. It is designed for the localization of two action categories: "drinking" and "smoking". The training sets contain 106 drinking and 78 smoking clips. The two test sets are two short stories (approx. 36,000 frames) containing 38 drinking actions and three short stories (approx. 32,000 frames) containing 42 smoking actions. There is no overlap between the training and test sets, both in terms of scenes and actors.

"DLSBP" [9]. Named after its authors, this dataset consists of two action categories: "Open Door" and "Sit Down". The training sets include 38 "Open Door" and 51 "Sit Down" examples

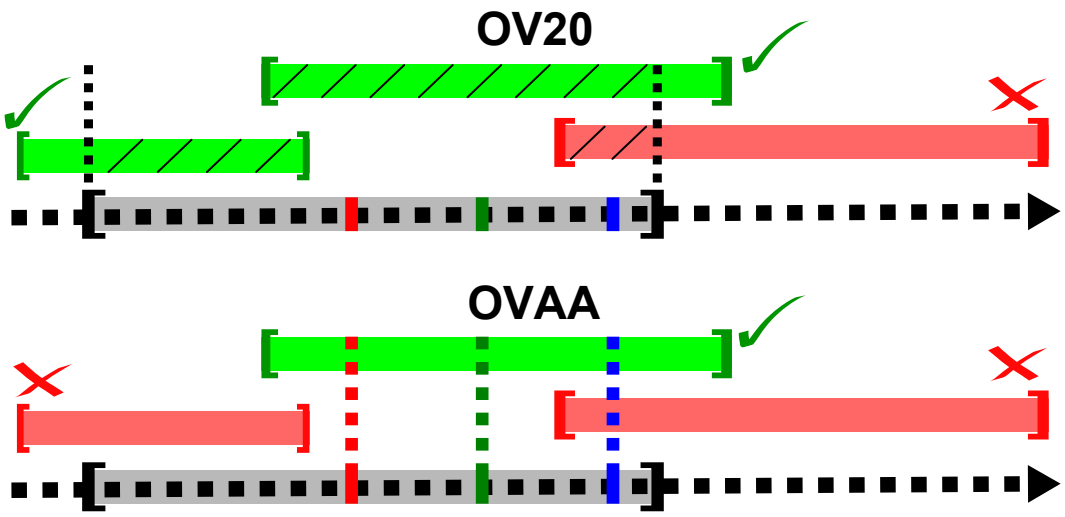


Figure 8: Overlap criteria used for detection: OV20 — which corresponds to an intersection over union of 20% with the ground truth — and OVAA — for which a match needs to contain all ground truth actom frames.

extracted from 15 movies. Three movies are used as test set (approx. 440,000 frames), containing a total of 91 "Open Door" and 86 "Sit Down" actions. This dataset is more challenging than "Coffee and Cigarettes", because the test data is larger by one order of magnitude, the actions are less frequent, and the video sources are more varied. Note that the chance level for detection, *i.e.* the probability of randomly finding the positives, is of approximatively 0.1% for the "Coffee and Cigarettes" dataset, and 0.01% for the "DLSBP" dataset.

"Hollywood 2" [13]. This classification dataset consists of 1707 video clips — 823 for training, 884 for testing — extracted from 69 Hollywood movies. There are 12 categories: answering a phone, driving a car, eating, fighting, getting out of a car, hand shaking, hugging, kissing, running, sitting down, sitting up, and standing up. We did not include the "fighting" category in our evaluation, because it could not be annotated with consistent actoms. Indeed, fighting *scenes* in movies are not short sequentially defined actions, but involve various actions in no particular order, *e.g.* punching, kicking, falling.

6.2 Evaluation criteria

For temporal detection, we use two evaluation criteria to determine if a test window is matching a ground truth action. We, first, consider the most commonly used criterion [9, 12, 78], referred to as OV20: a window matches a ground truth action if the Jaccard coefficient (intersection over union) is more than 20%. We use the original ground truth beginning and end frame annotations provided by the dataset authors.

This criterion does not guarantee that a detection will contain enough of the action to be judged relevant by a user. For instance, a detection relevant according to OV20 may contain a person walking towards a door, but not the door opening itself.

Therefore, in addition to OV20, we introduce a more precise matching criterion based on ground truth actom annotations. Referred to as OVAA, for "overlap all actoms", it states that a test window matches a ground truth test action only if it contains *the central frames of all ground truth actoms*. See Figure 8 for an illustration of the overlap criteria. The OVAA criterion comes

from the definition of actoms as the minimal set of sub-events needed to recognize an action. Hence, a correct detection must contain all actoms.

In consequence, we also annotate actoms for the positive test examples to assess ground truth according to OVAA. These annotations are not used at test time. Note, that a single window covering the entire test sequence will always match the ground truth according to the OVAA criterion. This bias, however, is not present in our comparisons, as all methods have comparable window sizes of approximatively 100 frames or less.

We use both criteria in our detection evaluation, as they provide complementary insights into the experimental results. If after non-maxima suppression there are multiple windows matching the same ground truth action, we only consider the one with the maximal score as a true positive, while the other detections are considered as false positives. This is similar to the evaluation of object detection, *e.g.* in the Pascal VOC challenge [79]. Note, that for classification by detection, no matching criterion is required as we return one score for each test video. In all cases, we measure performance in terms of precision and recall by computing the Average Precision (AP).

6.3 Bag-of-features baselines

We compare our approach to two baseline methods: the standard bag-of-features (BOF), and its extension with a regular temporal grid. To make the results comparable, we use the same visual features, vocabularies and kernel as the ones used for our ASM model. In addition, for detection experiments, we crop the original annotations of the positive training samples around the training actoms, which we extend by a small offset: half the inter-actom distances for each sequence. This step was shown to improve performance by Satkin and Hebert [10]. Furthermore, we use the same random training negative samples as the ones used by our ASM approach. This allows BOF-based methods to also use actom information, and, thus, makes the OVAA matching criterion agnostic.

At test time, BOF-based sliding window approaches require the *a priori* definition of multiple temporal scales. We learned the scales from the training set using a generative model similar to the one used for actoms (*c.f.* Section 4.2). Regarding the step-size by which the windows are shifted, we used 5 frames for all of our experiments. We finally apply a non-maxima suppression post-processing to the windows, similar to the one described in Section 4.2 and commonly used in the literature, *e.g.* in [78].

In addition to the global BOF baseline, we evaluate its extension with regular temporal grids [6]. We use a fixed grid of three equally sized temporal bins, which in practice gave good results and is consistent with our number of actoms. First, the video is cut in three parts of equal duration — beginning, middle and end. A BOF is then computed for each part and the three histograms are concatenated. In the following, this method is referred to as "BOF T3".

6.4 Detection results

We report temporal detection results in table 1a for the "Coffee and Cigarettes" dataset and in table 1b for the "DLSBP" dataset. We compare our method (ASM), two baselines (BOF and BOF T3), and recent state-of-the-art results. Where possible, we report the mean and standard deviation of the performance over five independent runs with different random negative training samples. Figure 9 shows frames of the top five results for "drinking" and "open door" obtained with our method. Some examples of automatically detected actoms with our ASM method are depicted in Figure 2. In the following, we discuss how our ASM model compares to both our bag-of-features baselines and the state of the art.

Figure 9: Frames of the top 5 actions detected with ASM for "Drinking" (top row) and "Open Door" (bottom row).

Method	"Drinking"	"Smoking"
matching criterion: OV20		
DLSBP [9]	40	NA
LP-T [12]	49	NA
KMSZ-T [78]	59	33
BOF	36 (±1)	17 (±2)
BOF T3	44 (±2)	20 (±3)
ASM	**63** (±3)	**40** (±4)
matching criterion: OVAA		
BOF	10 (±3)	1 (±0)
BOF T3	21 (±4)	3 (±1)
ASM	**62** (±3)	**27** (±3)

(a) Coffee and Cigarettes

Method	"Open Door"	"Sit Down"
matching criterion: OV20		
DLSBP [9]	14	14
BOF	8 (±3)	14 (±3)
BOF T3	8 (±1)	17 (±3)
ASM	**14** (±3)	**22** (±2)
matching criterion: OVAA		
BOF	4 (±1)	3 (±1)
BOF T3	4 (±1)	6 (±2)
ASM	**11** (±3)	**19** (±1)

(b) DLSBP

Table 1: Action detection results in Average Precision (in %). ASM refers to our method.

Comparison to bag-of-features. We perform better than BOF according to both evaluation criteria. The improvement is significant for the OV20 criterion: +27% for "Drinking", +23% for "Smoking", +6% for "Open Door", and +8% for "Sit Down". BOF is also less precise than our approach. Indeed, the performance of BOF drops when changing the matching criterion from OV20 to the more restrictive OVAA — *e.g.* −26% for "Drinking". In contrast, our ASM model is more accurately detecting all action components and the relative gap in performance with respect to the baseline increases significantly when changing from OV20 to OVAA: from +27% to +52% for "Drinking", and from +8% to +16% for "Sit Down".

Rigid *v.s.* adaptive temporal structure. The flexible temporal structure modeled by ASM allows for more discriminative models than BOF T3. Using the fixed temporally structured extension of BOF increases performance, but is outperformed by our model on all actions. This confirms that the variable temporal structure of actions needs to be represented with a flexible model that can adapt to different durations, speeds, and interruptions.

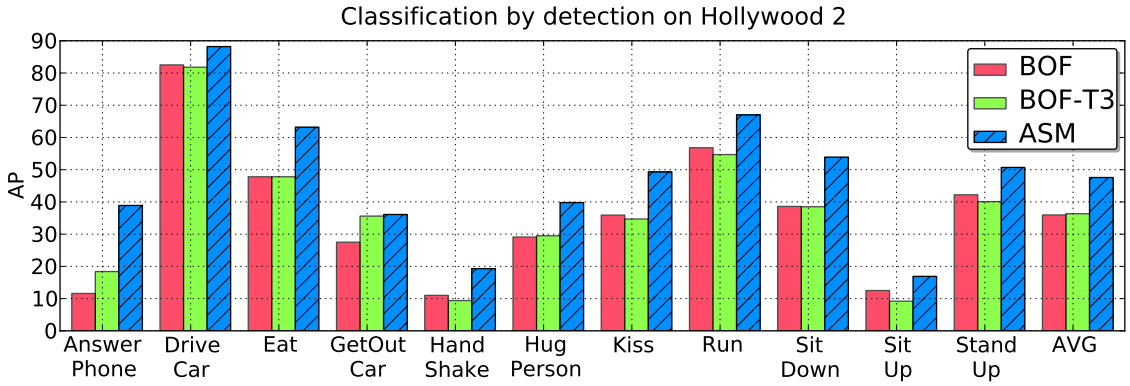


Figure 10: Classification by detection results, in Average Precision (AP), on the "Hollywood 2" dataset [13]. "BOF" and "BOF-T3" are sliding-window approaches using BOF and its temporally structured extension. Our approach is "ASM". "AVG" contains the average performance over all classes; BOF: 36%, BOF T3: 36%, ASM: 48%.

Comparison to the state of the art. The method of Laptev and Pérez [12] is trained for spatio-temporal localization with stronger supervision in form of spatially and temporally localized actions. They only report results for the "Drinking" action of the "Coffee and Cigarettes" dataset. We compare to the mapping of their spatio-temporal detection results to the temporal domain as reported in [9], *c.f.* row "LP-T" in table 1a. Similarly, Kläser *et al.* [78] learn from spatio-temporally localized training examples and additionally publish localization results for the "Smoking" action. The mapping of their results to the temporal domain are reported in the "KMSZ-T" row of table 1a. On the "DLSBP" dataset, we compare to the original "ground truth" results of the authors in [9]. They use a similar set-up to our BOF baseline. The differences between their approach and our BOF baseline lies mostly in the negative training samples and, to a lesser extent, in the visual vocabulary.

Our experiments show that ASM outperforms these state-the-art approaches, for all actions of the two datasets. Our method even outperforms methods trained with more complex supervision, like bounding boxes, +14% with respect to LP-T [12], or human tracks, +4% and +7% with respect to KMSZ-T [78]. This shows that appropriately modeling the temporal structure of actions is crucial for performance.

6.5 Classification-by-detection results

Figure 10 contains the per class classification by detection results on the "Hollywood 2" dataset. Note, that the BOF baselines are using the same sliding window approach as in the previous detection results.

On average over all classes, ASM improves by +12% over both BOF baselines, which have similar performance — BOF T3 only marginally improves by +0.4% with respect to BOF. The improvement yielded by ASM is noticeable on the classes with a clear sequential nature such as "Answer Phone", "Hug Person" or "Sit Down". Interestingly, ASM always improves performance, even when BOF T3 yields poorer results than just BOF, *e.g.* for "Hand Shake" and "Stand Up". Once again, these results show that a flexible model of the temporal structure is required in order to recognize real-world actions.

We also evaluate baseline classification methods similar to [6], where a single model is computed over the entire duration of each test video. On average over all classes, we obtained

	C&C		DLSBP	
	OV20	OVAA	OV20	OVAA
BOF (s-win)	27.5	5.0	11.0	3.5
BOF (s-cfr)	35.5	21.5	12.5	9.0
BOF T3 (s-win)	32.0	12.0	12.5	5.0
BOF T3 (s-cfr)	37.0	26.5	14.0	9.5
ASM (s-cfr)	51.5	44.5	18.0	15.0

Table 2: Sliding window (s-win) *v.s.* sliding central frame (s-cfr). Average of the detection results on the "Coffee and Cigarettes" (C&C) and "DLSBP" datasets.

approximately the same results of 45% AP for three different models: BOF, BOF T3, and ASM with uniformly spread actoms and $\rho = p = 75\%$. Note, that the similar performance of all three global models shows that the benefits of ASM do not only lie in its use of soft-voting.

In comparison, ASM with classification by detection achieves 48% AP. This +3% gain is less significant than for temporal detection, because classification of pre-segmented videos is an easier problem. Indeed, classification with BOF improves by +9% over the classification by detection results with the same BOF model. In addition, global models use context information, whereas the more local representations used for detection focus only on the action.

6.6 Parameter study

We measured the impact of the different components of our approach: (i) the sliding central frame detection method compared to the sliding window technique, (ii) manual actom annotations compared to uniformly spread ones, (iii) the ASM parameters, (iv) the number of candidate temporal structures learned, and (v) the number of training negatives.

Sliding central frame. First, we found that our sliding central frame approach consistently outperforms the traditional sliding window one. Therefore, marginalizing over a generative model of the temporal structure is preferable to the commonly used scale sampling heuristics. This can be observed in table 2, where we report the detection results using BOF models in conjunction with our sliding central frame approach. In this case, we adopt the same method as described in Section 5.1: a prior on the action duration is learned with the algorithm from Section 4.2 and detection is performed by marginalizing over this 1D distribution on temporal extents. In contrast, the sliding window approach also uses multiple scales learned from the training data, but it does not marginalize over a generative model of these scales. Note also, that ASM still outperforms BOF baselines with sliding central frames.

Manual training actoms. Second, we computed the detection results using our ASM approach with training actoms spread uniformly between the manually annotated temporal boundaries. We observed that detection results are significantly worse than when using manually annotated training actoms. Indeed, ASM with these uniform actoms yields results similar to "BOF T3" with the sliding central frame approach. This shows that temporal boundaries do not provide enough information to model the temporal aspects of an action.

ASM parameters		C&C		DLSBP	
ρ	p	OV20	OVAA	OV20	OVAA
low	high	40.3	34.5	11.4	9.0
high	low	39.0	30.9	12.5	10.0
high	high	45.5	34.8	15.0	11.2
low	low	49.8	42.8	15.1	11.9
learned		51.5	44.5	18.0	15.0

Table 3: Impact of the ASM parameters: ρ (overlap) and p (peakyness). Average of the detection results on the "Coffee and Cigarettes" (C&C) and "DLSBP" datasets.

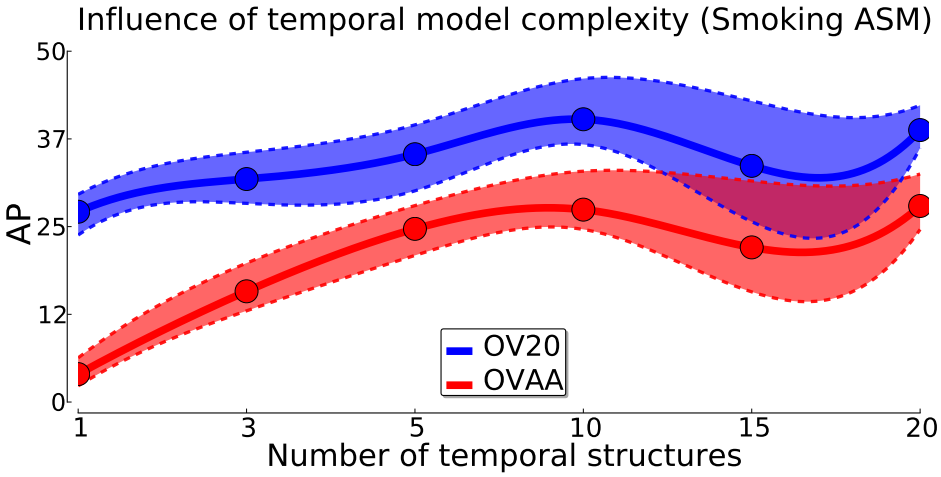


Figure 11: Minimum, average, and maximum detection performance for action "Open Door" *v.s.* size of the support of $\hat{\mathcal{D}}$ (number of candidate temporal structures).

ASM parameters. Third, we studied the impact of the ASM parameters on performance. In table 3, we report detection results for ASM with learned parameters — *c.f.* an example in Figure 6 — and for different parameter configurations — corresponding to the four corners in Figure 4. These results show that learning action-specific ASM overlap and peakyness parameters yields the most accurate models, resulting in increased detection performance. Note, that the learned parameters change from one action to the other. For instance, the learned parameters for "Smoking" are $\rho = 25\%$ and $p = 70\%$, denoting clearly separated actoms, whereas for "Sit Down" we obtain $\rho = 120\%$ and $p = 50\%$, denoting actoms sharing a significant amount of frames.

Temporal model complexity. In addition, we studied the impact of the complexity of the temporal structure model — measured by the support size s of $\hat{\mathcal{D}}$, *c.f.* Eq. 7 — on the detection performance. This parameter controls a trade-off between the precision of the model and, as we marginalize over this distribution, the computational complexity at test time. We found that $s = 10$ candidate actom structures yields a good compromise for most classes — *c.f.* Figure 11 for an illustration using the "Smoking" action. On the one hand, if $s < 5$, then the model is too simple and the performance gap between the OV20 and OVAA results is large. On the other hand, if $s > 15$, then results are equivalent to $5 \leq s \leq 15$ but at a higher computational cost.

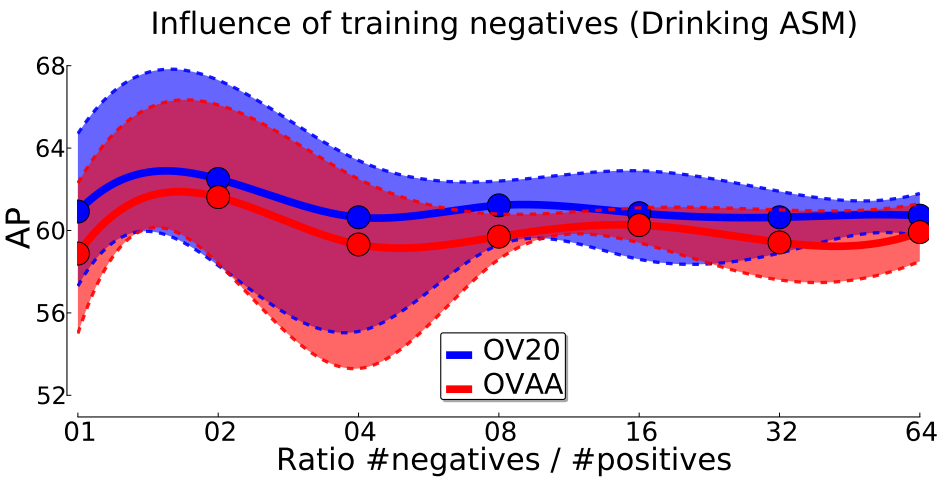


Figure 12: Minimum, average, and maximum detection performance for action "Drinking" *v.s.* the number of random training negatives. The x-axis is in log-scale.

Training negatives. Finally, we measured the detection performance as a function of the number of random training negatives used — see for instance Figure 12 for the "Drinking" action. We found that between one and ten times the number of positives was sufficient to reach satisfactory performance for all classes. We sampled twice more negatives than positives for the Coffee and Cigarettes dataset, and eight times more for the DLSBP dataset. This choice allows to maintain a high true positive over false negative ratio, while limiting the imbalance factor between classes and speeding up detection.

Sampling only few training negatives, however, yields unstable results, as illustrated by the large standard deviations reported in our experiments. This instability can be controlled using a simple bagging approach — *i.e.* averaging classifiers over different negative training sets as suggested in [80] — at the expense of an increase in computational complexity.

7 Conclusion

In this paper, we introduced the Actom Sequence Model (ASM). This model describes an action with a temporal sequence of actoms, which are characteristic of the action. It is discriminative, as it represents an action by several components instead of one average representation as in the bag-of-features. It is flexible, as our temporal representation allows for varying temporal speed of an action as well as interruptions within the action. Experimental results show that our approach outperforms the bag-of-features as well as its extension with a fixed temporal grid. Furthermore, ASM improves over the state of the art, including more sophisticated models using spatial localization.

Acknowledgments

This work was partially funded by the MSR/INRIA joint project, the European integrated project AXES and the PASCAL 2 Network of Excellence. We would like to thank Ivan Laptev for the "DLSBP" dataset.

References

[1] R. Poppe, "A survey on vision-based human action recognition," *Image Vision Computing*, 2010.

[2] D. Weinland, R. Ronfard, and E. Boyer, "A survey of vision-based methods for action representation, segmentation and recognition," *CVIU*, 2010.

[3] J. Aggarwal and M. Ryoo, "Human activity analysis: A review," *ACM Comput. Surv.*, 2011.

[4] C. Schüldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local SVM approach," in *ICPR*, 2004.

[5] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *CVPR*, 2005.

[6] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld, "Learning realistic human actions from movies," in *CVPR*, 2008.

[7] P. Dollár, V. Rabaud, G. Cottrell, and S. Belongie, "Behavior recognition via sparse spatio-temporal features," in *VS-PETS*, 2005.

[8] J. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, 2008.

[9] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce, "Automatic annotation of human actions in video," in *ICCV*, 2009.

[10] S. Satkin and M. Hebert, "Modeling the temporal extent of actions," in *ECCV*, 2010.

[11] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin, "Action recognition by dense trajectories," in *CVPR*, 2011.

[12] I. Laptev and P. Perez, "Retrieving actions in movies," in *ICCV*, 2007.

[13] M. Marszalek, I. Laptev, and C. Schmid, "Actions in contexts," in *CVPR*, 2009.

[14] J. Yamato, J. Ohaya, and K. Ishii, "Recognizing human action in time-sequential images using hidden markov model," in *CVPR*, 1992.

[15] T. Starner and A. Pentland, "Real-time American Sign Language recognition from video using Hidden Markov Models," in *International Symposium on Computer Vision*, 1995.

[16] A. Wilson and A. Bobick, "Learning visual behavior for gesture analysis," in *International Symposium on Computer Vision*, 1995.

[17] M. Brand, N. Oliver, and A. Pentland, "Coupled hidden markov models for complex action recognition," in *CVPR*, 1997.

[18] L. Rabiner and R. Schafer, "Introduction to digital speech processing," *Foundations and trends in signal processing*, 2007.

[19] N. Oliver, B. Rosario, and A. Pentland, "A bayesian computer vision system for modeling human interactions," *PAMI*, 2000.

[20] B. Laxton, J. Lim, and D. Kriegman, "Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video," in *CVPR*, 2007.

[21] F. Lv and R. Nevatia, "Single view human action recognition using key pose matching and Viterbi path searching," in *CVPR*, 2007.

[22] Z. Zeng and Q. Ji, "Knowledge based activity recognition with dynamic bayesian network," in *ECCV*, 2010.

[23] K. Kulkarni, E. Boyer, R. Horaud, and A. Kale, "An unsupervised framework for action recognition using actemes," in *ACCV*, 2010.

[24] C. Chen and J. Agarwal, "Modeling human activities as speech," in *CVPR*, 2011.

[25] Q. Shi, L. Cheng, L. Wang, and A. Smola, "Human action segmentation and recognition using discriminative semi-markov models," *IJCV*, 2011.

[26] M. Hoai, Z. Lan, and F. De la Torre, "Joint segmentation and classification of human actions in video," in *CVPR*, 2011.

[27] T. Darrell and A. Pentland, "Space-time gestures," in *CVPR*, 1993.

[28] S. Niyogi and E. Adelson, "Analyzing and recognizing walking figures in XYT," in *CVPR*, 1994.

[29] D. Gavrila and L. Davis, "Towards 3-d model-based tracking and recognition of human movement: a multi-view approach," in *International Workshop on Automatic Face and Gesture recognition*, 1995.

[30] A. Veeraraghavan, R. Chellappa, and A. Roy-Chowdhury, "The function space of an activity," in *CVPR*, 2006.

[31] W. Brendel and S. Todorovic, "Activities as time series of human postures," in *ECCV*, 2010.

[32] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *Transactions on Acoustics, Speech and Signal Processing*, 1978.

[33] M. Brand and V. Kettnaker, "Discovery and segmentation of activities in video," *PAMI*, 2000.

[34] L. Gorelick, M. Blank, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," *PAMI*, 2007.

[35] T. Kim and R. Cipolla, "Canonical correlation analysis of video volume tensors for action categorization and detection," *PAMI*, 2009.

[36] A. Bobick and J. Davis, "The recognition of human movement using temporal templates," *PAMI*, 2001.

[37] M. Rodriguez, J. Ahmed, and M. Shah, "Action mach: a spatio-temporal maximum average correlation height filter for action recognition," in *CVPR*, 2008.

[38] R. Polana and R. Nelson, "Low level recognition of human motion," in *IEEE Workshop on Nonrigid and Articulate Motion*, 1994.

[39] E. Shechtman and M. Irani, "Space-time behavior based correlation," in *CVPR*, 2005.

[40] A. Efros, A. Berg, G. Mori, and J. Malik, "Recognizing action at a distance," in *ICCV*, 2003.

[41] K. Schindler and L. Van Gool, "Action snippets: How many frames does human action recognition require," in *CVPR*, 2008.

[42] Y. Ke, R. Sukthankar, and M. Hebert, "Volumetric features for video event detection," *IJCV*, 2010.

[43] A. Wilson and A. Bobick, "Parametric Hidden Markov Models for gesture recognition," *PAMI*, 1999.

[44] A. Gaidon, M. Marszałek, and C. Schmid, "Mining visual actions from movies," in *BMVC*, 2009.

[45] A. Patron-Perez, M. Marszałek, A. Zisserman, and I. D. Reid, "High five: Recognising human interactions in TV shows," in *BMVC*, 2010.

[46] J. Liu, J. Luo, and M. Shah, "Recognizing realistic actions from videos "in the wild"," in *CVPR*, 2009.

[47] N. Ikizler-Cinbis, R. Cinbis, and S. Sclaroff, "Learning actions from the web," in *ICCV*, 2009.

[48] O. Chomat and J. Crowley, "Probabilistic recognition of activity using local appearance," in *CVPR*, 1999.

[49] L. Zelnik-Manor and M. Irani, "Event-based analysis of video," in *CVPR*, 2001.

[50] I. Laptev, "On space-time interest points," *IJCV*, 2005.

[51] G. Willems, T. Tuytelaars, and L. Van Gool, "An efficient dense and scale-invariant spatio-temporal interest point detector," in *ECCV*, 2008.

[52] D. Han, L. Bo, and C. Sminchisescu, "Selection and context for action recognition," in *ICCV*, 2009.

[53] A. Gilbert, J. Illingworth, and R. Bowden, "Action recognition using mined hierarchical compound features," *PAMI*, 2010.

[54] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid, "Evaluation of local spatio-temporal features for action recognition," in *BMVC*, 2009.

[55] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.

[56] G. Willems, J. Becker, T. Tuytelaars, and L. Van Gool, "Exemplar-based action recognition in video," in *BMVC*, 2009.

[57] A. Yao, J. Gall, and L. Van Gool, "A hough transform-based voting framework for action recognition," in *CVPR*, 2010.

[58] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *PAMI*, 2011.

[59] S. Nowozin, G. Bakir, and K. Tsuda, "Discriminative subsequence mining for action classification," in *ICCV*, 2007.

[60] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *ICCV*, 2003.

[61] G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray, "Visual categorization with bags of keypoints," in *Workshop on statistical learning in computer vision, ECCV*, 2004.

[62] B. Schölkopf and A. J. Smola, *Learning with Kernels*. MIT Press, 2002.

[63] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei, "Modeling temporal structure of decomposable motion segments for activity classification," in *ECCV*, 2010.

[64] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *PAMI*, 2009.

[65] P. Ekman and W. V. Friesen, *Facial Action Coding System*. Consulting Psychologists Press, 1978.

[66] J. Cohn and T. Kanade, "Use of automated facial image analysis for measurement of emotion expression," in *Handbook of emotion elicitation and assessment*. Oxford UP Series in Affective Science, 2006.

[67] T. Simon, M. Nguyen, F. De la Torre, and J. Cohn, "Action unit detection with segment-based SVM," in *CVPR*, 2010.

[68] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom sequence models for efficient action detection," in *CVPR*, 2011.

[69] I. Laptev, "STIP: Spatio-Temporal Interest Point library." [Online]. Available: www.di.ens.fr/~laptev/interestpoints.html

[70] M. Hein and O. Bousquet, "Hilbertian metrics and positive definite kernels on probability measures," in *AISTATS*, 2005.

[71] S. Maji, A. C. Berg, and J. Malik, "Classification using intersection kernel support vector machines is efficient," in *CVPR*, 2008.

[72] Y. Lin, Y. Lee, and G. Wahba, "Support vector machines for classification in nonstandard situations," *Machine Learning*, 2002.

[73] J. Platt, "Probabilistic outputs for support vector machines," *Advances in Large Margin Classifiers*, 2000.

[74] H. Lin, C. Lin, and C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, 2007.

[75] M. Rosenblatt, "Remarks on some nonparametric estimates of a density function," *The Annals of Mathematical Statistics*, 1956.

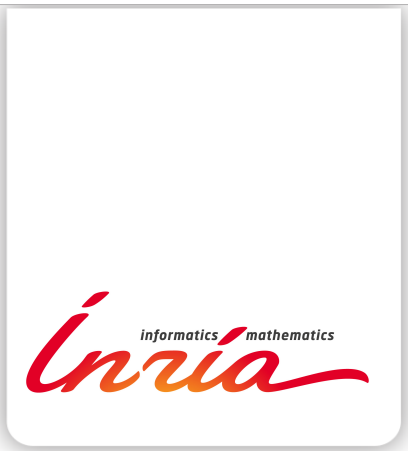
[76] L. Wasserman, *All of statistics: a concise course in statistical inference*. Springer Verlag, 2004.

[77] D. Scott, *Multivariate density estimation: theory, practice, and visualization*. Wiley, 1992.

[78] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman, "Human focused action localization in video," in *SGA*, 2010.

[79] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *IJCV*, 2010.

[80] F. Mordelet and J.-P. Vert, "A bagging SVM to learn from positive and unlabeled examples," Tech. Rep., 2010. [Online]. Available: http://hal.archives-ouvertes.fr/hal-00523336



RESEARCH CENTRE
GRENOBLE – RHÔNE-ALPES

Inovallée
655 avenue de l'Europe Montbonnot
38334 Saint Ismier Cedex

Publisher
Inria
Domaine de Voluceau - Rocquencourt
BP 105 - 78153 Le Chesnay Cedex
inria.fr

ISSN 0249-6399