Temporal Localization of Actions with Actoms

Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid

Temporal Localization of Actions with Actoms

Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid

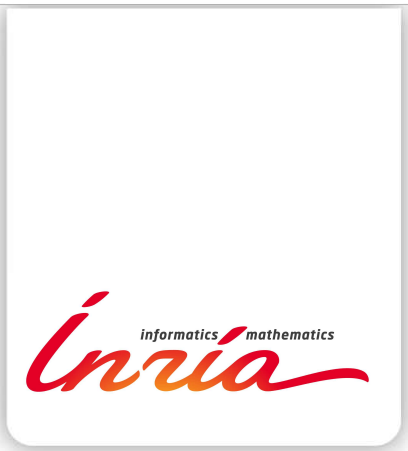
RESEARCH REPORT

N° 7930

April 2012

Project-Teams LEAR and MSR-INRIA

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



Temporal Localization of Actions with Actoms

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

Project-Teams LEAR and MSR-INRIA

Research Report n° 7930 — version 2 — initial version April 2012 — revised version January 2013 — 31 pages

Abstract: We address the problem of localizing actions, such as 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 an action as a sequence of histograms of actom-anchored visual features, which can be seen as a temporally structured extension of the bag-of-features. Training requires the annotation of actoms for action examples. At test time, actoms are localized 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 action localization "Coffee and Cigarettes" and the "DLSBP" dataset. We also adapt our approach to a classification-by-localization 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 localization, as well as baselines that localize actions with a sliding window method.

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

Localisation 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, Localisation

Contents

Figure 1: *Examples of actom annotations for two actions.*

1 Introduction

Automatic understanding of video content is a challenging and important problem due to the massive increase in available video data. In particular, action recognition is an active topic in the computer vision community (*cf.* [1–3] 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]. Several works have reported that the bag-of-features model (BOF) can efficiently describe actions in such real-world video conditions [6–10]. However, it 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. This allows for efficient *temporal action localization*, *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 especially important for the localization of 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 by 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 should be adapted to each action. 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. Actoms are an intermediate layer between *motion primitives*, like spatio-temporal interest points, and *actions*.

We make the following contributions. First, in Section 3, we introduce a temporally structured representation of actions, called Actom Sequence Model (ASM). It encodes the temporal ordering 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 concurrent subevents. 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 localization using a sliding central frame approach, which we, then, extend to a classification by localization scenario. Note that in addition to localizing actions, our approach can return the most likely actoms of localized actions, as illustrated in Figure 2.

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 localization: the "Coffee and Cigarettes" [11] and "DLSBP" [8] datasets. We also demonstrate the applicability of our approach in a classification by localization setup on a larger set of actions from the Hollywood 2 dataset [12].

2 Related work

In the following, we review the main action localization methods that model temporal aspects of videos.

Based on the observation that digital videos are successions of frames, **sequential approaches** represent actions as sequences of states, *e.g.* poses. Inspired by speech recognition, *sequential probabilistic models* [13–21] use dynamic probabilistic graphical models — *e.g.* Hidden Markov Models (HMM) [22] — to learn temporal transitions between hidden states. Also operating on sequence representations of actions, *exemplar-based methods* [23–25] directly compute an alignment score between an action and template sequences. In general, they require less training data and provide more flexibility, as they can handle non-linear speed variations using Dynamic Time Warping (DTW) [26]. A limitation of sequential approaches, however, is their difficulty to represent concurrent subevents. Some models address this issue — *e.g.* coupled HMMs [15] or Dynamic Bayesian Networks [16] — by relying on a complex and domain-specific structure manually specified by experts. In addition, sequential *localization* methods often rely on higher-level probabilistic models, for instance by combining multiple HMMs [17, 27]. These models require a large amount of training examples in order to model all events that might occur, including non-actions.

Figure 2: *Actom frames of localized test sequences.*

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, 28]. Some models operate directly on spatio-temporal volumes. For instance, Kim and Cipolla [29] directly compare videos using Tensor Canonical Correlation Analysis. Alternatively, several approaches [5, 28, 30, 31] rely on silhouettes in order to obtain spatio-temporal templates, but their use is limited to simple or controlled video conditions. Other approaches [32–36] focus on optical flow information to obtain action templates. For instance, Efros *et al.* [34] compute blurred optical flow features inside tracks of soccer players. As volumetric approaches rely on a similarity measure between video volumes, they typically localize actions by matching sub-volumes with a set of candidate templates. For instance, Ke *et al.* [36] 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 [23] with DTW and in [37] with HMMs. Volumetric methods, however, often require the videos to be spatio-temporally aligned. Hence, these techniques are not robust to occlu-

sions, partial observations, and significant viewpoint and duration variations. In addition, volumetric approaches tend to assume that the video volume spanned by an action is contiguous. Consequently, they are not adapted to actions with interruptions or with multiple interacting actors, such as kissing.

Models using **local features** [4, 6, 7, 10–12, 38–45] represent videos as collections of local X-Y-T patches. These representations are, in general, more robust than sequential or volumetric ones, especially under real-world video conditions, *e.g.* in movies [6, 11, 12], TV shows [46, 47], Youtube clips [44, 48], or sports broadcasts [31] (*cf.* [49] for a recent evaluation). There are two main families of localization techniques based on local features: local classifiers, which reason directly on local features, and global approaches, which operate on representations aggregating features over video sub-volumes.

Local classifiers [11, 44, 45, 50–53] measure the importance of each feature for a particular action. For instance, Yuan *et al.* [53] detect spatio-temporal interest points, which cast votes based on their point-wise mutual information with the action category. These approaches have difficulties distinguishing between actions sharing common motion or appearance primitives, as they focus mostly on local aspects.

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, 10, 40]. Inspired by text document classification [54, 55], a video is holistically represented with a histogram of occurrences of local features quantized over a "visual vocabulary" obtained by unsupervised learning. Statistical learning methods like Support Vector Machines (SVM) [56] can then be applied to learn a BOF classifier. Localization is generally performed by applying this classifier in a sliding window manner. Though powerful, this model discards most of the temporal information inherent to actions. Therefore, it is not well adapted to discriminate between actions distinguished by their structure, *e.g.* opening and closing a door. This type of confusion leads to many high-score false alarms when scanning videos to localize an action.

Related to our work, Duchenne *et al.* [8] and Satkin and Hebert [9] observe that temporal boundaries of actions are not precisely defined in practice. Furthermore, they show that inaccurate boundary annotations can significantly degrade the recognition performance. Therefore, they propose to improve the quality of annotated action clips by automatically cropping their temporal boundaries. They model the *temporal extent* of actions, whereas our algorithm learns a generative model of an action's *temporal structure*.

In order to model the structure of actions, Laptev *et al.* [6] combine multiple BOF models extracted for different rigid spatio-temporal grids selected manually. This approach is shown to slightly improve over the standard BOF one. However, the structure of actions is fixed and not explicitly modeled. In contrast, we use a flexible model and soft-assignment to multiple temporal grids learned from training data and adapted to the action. Furthermore, we encode a richer structure that captures different potential execution styles, whose prior probabilities are estimated. We show that, compared to rigid grids, this results in significant gains in localization performance.

Latent variable approaches [21, 57, 58] are another family of popular methods used to recognize actions, which is related to our work. They consist in modeling parameters of a structured model as hidden variables, which are typically estimated using a latent SVM [59]. For instance, Niebles *et al.* [57] discover temporal parts and learn a SVM classifier per video segment at latent temporal locations. These methods rely on (spatio-)temporal boundary annotations, and automatically infer the latent structure of training videos, whereas we assume it is given by annotators. At test time, latent variable methods infer the best structure. In contrast, we adopt a simpler and more robust Bayesian approach by marginalizing over a learned structure prior. In addition, we decouple the learning of the structure and content models, whereas latent variable approaches learn them jointly by solving a complex non-convex optimisation problem with many parameters. Note that this decoupling also allows to interpret the learned temporal prior.

Our method is also similar in spirit to state-of-the-art approaches for facial expression recognition from videos. They use label information defined by the Facial Action Coding System (FACS) [60], which segments facial expressions into predefined "action units", complemented with temporal annotations such as *onset*, *peak*, *offset*. Most approaches, however, model only the peak frame [61] or a single video segment [62]. Furthermore, as the complexity of generic human actions makes the construction of universal action units impractical, we investigate user-defined, action-specific actoms.

A preliminary version of this work appeared in [63].

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 — *cf.* 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.

3.1 Local visual information in actoms

Following recent results on action recognition in challenging video conditions [6, 8], we extract sparse space-time features [41]. 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 [64].

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 [8], we use the k-means algorithm with a number of clusters set to $v = 1000$ for our localization experiments, while, similar to [49], we use $v = 4000$ for our classification-by-localization experiments. We then assign each feature to the closest visual word.

3.2 The Actom Sequence Model (ASM)

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*:

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

with $\rho \in [0, 1)$ and δ_i 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 using a *temporally weighted assignment scheme*. The contribution of a 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:

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

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:

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

The probability $p \in [0, 1]$ is the "peakyness" of our soft-assignment scheme, and replaces σ as parameter of our model. It encodes a prior on the probability mass of features contained in an actom's time range. In our experiments, we set the ρ and p parameters per-class by maximizing localization performance on a held out, unsegmented, validation video.

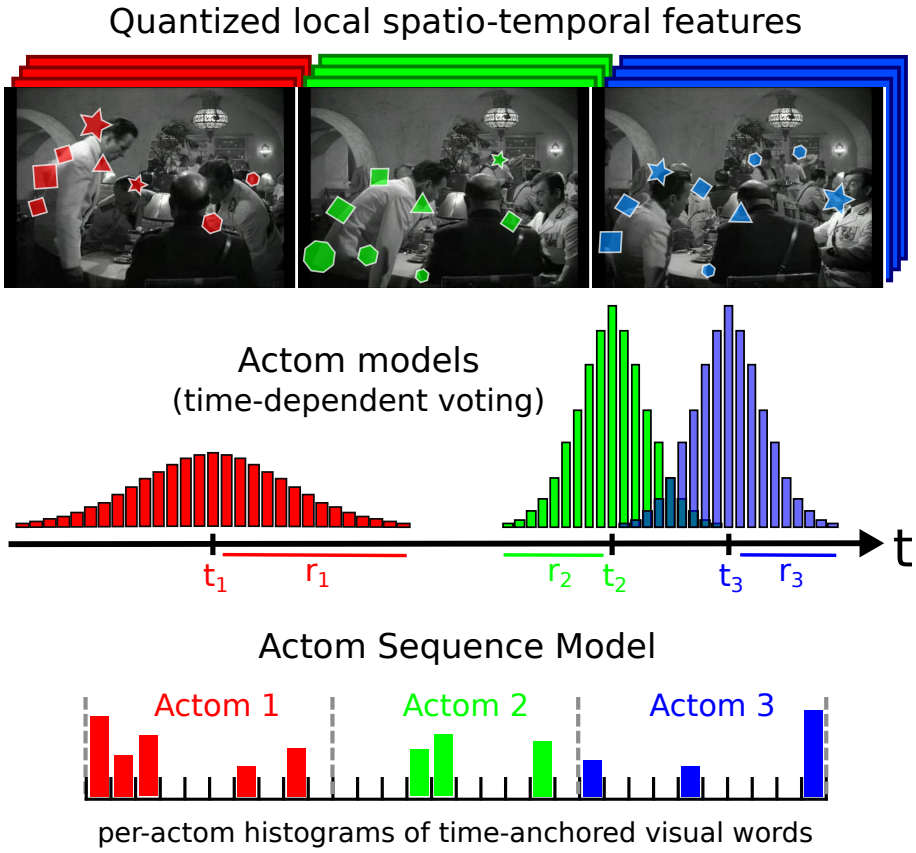


Figure 3: *Construction of ASM using actom-based annotations and a temporal weighting scheme for aggregating local features in a sparse temporally structured bag-of-features.*

In summary, we derive our ASM model from a sequence of a actom locations by (i) computing visual features only in the actoms's time-spans 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 $[t_i - r_i, t_i + r_i]$. Similar to spatio-temporal pyramids [6], the ASM vector $\mathbf{x}$ is then L_1-normalized.

Our aforementioned parametrization has multiple advantages. First, an

adaptive time-span makes the model naturally robust to variable action duration and speed. Second, 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 sub-events. Third, our parametrization also allows for gaps between actoms, which is useful for discontinuous actions. In conclusion, ASM encodes a flexible temporal ordering of the actoms, which can represent a sequence of mutually exclusive steps, as well as actions involving concurrent sub-events. Note, however, that we do not use the spatial location of the spatio-temporal features, and cannot, therefore, distinguish between actors in the same video.

3.3 Actom annotations

Our supervised learning approach relies on positive training examples: videos containing the action with manually annotated actoms. An actom annotation is a frame in the corresponding video. This temporal location is selected such that its neighboring frames contain visual information representative of a part of the action. Annotators are asked to mark *as few as possible* key moments (at least two), from which the action can be *unambiguously* recognized, without any a priori definition of these key moments. We also ask for *semantic consistency* in the choice of actoms across different video clips: the i^{th} actom of an action should, if possible, have a unique interpretation, like "recipient containing liquid coming into contact with lips" for the drinking action. However, we can still model actions when an actom has several possible meanings across training examples. Our learning stage can, indeed, account for multiple modes of execution (styles) of an action (*cf.* Section 4).

Due to annotator differences, occlusions, and intra-class variability, we obtained a variable number of temporal parts for the videos of a category. However, in our experiments, close to 90% of the annotations contained exactly 3 actoms corresponding to start, middle, and end phases of the action. The remaining 10% annotations contained either 2 actoms (*e.g.* due to a scene cut) or 4 (*e.g.* for a slow performance). In those cases, we simply interpolated an additional actom or dropped one, as our approach relies on a constant number of parts for all videos of an action. As shown in Figure 1, such subdivisions are semantically meaningful for short actions and interactions like opening a door. Note that their temporal locations (focusing on a part of the action) and uneven spacings (reflecting speed variations) make these annotations significantly different from a simple uniform temporal binning.

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 with a low estimated probability. 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.

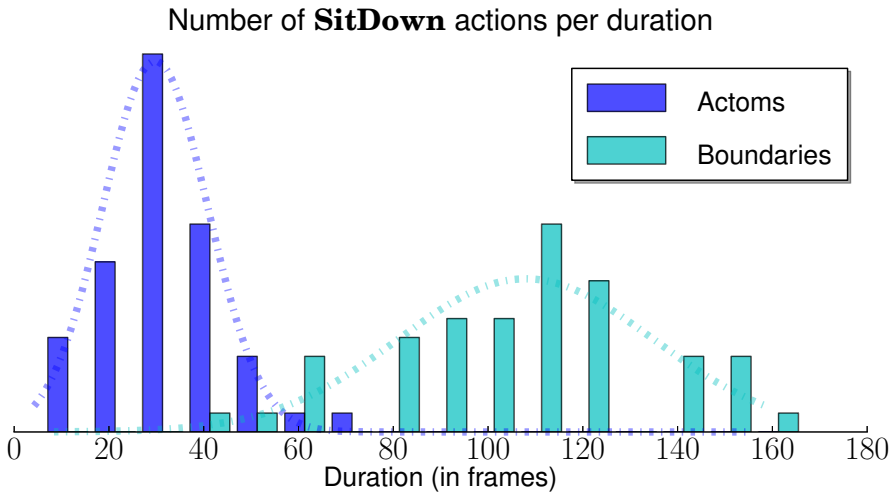


Figure 4: *Frequencies of action durations obtained from manual annotations for Sit Down actions. "Actoms": extent of windows encompassing entire actom sequences. "Boundaries": duration annotations of [8]. Using actoms leads to smaller annotation variability.*

The average annotation time per action is comparable for labeling actoms or the overall temporal extent — between 10 and 30 seconds per action. In addition, we observed that consistent actom annotations seem 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 starts to open can be unambiguously determined. Figure 4 quantitatively illustrates this claim. It suggests 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. Finally, an actom is a visual phenomenon that is deemed semantically relevant by annotators. Therefore, our model is both interpretable and, as suggested by our experiments, discriminative.

Our approach is adapted to many common action categories, especially simple interactions. However, not all actions can be clearly decomposed as a sequence of actoms, in particular very fast actions (*e.g.* punching) and non-sequential high-level activities (*e.g.* fighting, cooking). In these cases, we simply use a regular temporal binning to get evenly spaced actoms between the annotated temporal boundaries. Note that periodic actions (*e.g.* walking) are annotated either using a fixed number of periods (typically only one if the action is slow and decomposable), or by annotating the same actoms across multiple periods (for fast actions like running).

4 Temporal action detector learning

In the following, we detail the training phase of our localization 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 localization 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) [56] 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, 65]. For efficiency reasons, we choose to use the intersection kernel [66]. It is defined for any $x = (x_1, \ldots, x_v)$ and $x' = (x'_1, \ldots, x'_v)$ as $K(x, x') = \sum_{j=1}^{v} \min(x_j, x'_j)$.

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 [67] 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:

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

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 [68, 69]. 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, and use it as a prior during localization in order to marginalize out the latent structure. We estimate a distribution over inter-actom spacings from the training actom 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. To limit the annotation cost, only few actom annotations are available in practice (typically $n_+ \leq 100$). In addition, they can significantly differ from one another. Therefore, *directly* estimating a *discrete* distribution on the available actom spacings — *e.g.* using histograms or mean-shift — yields a too sparse estimate with an emphasis on rare configurations. Instead, we make the assumption that there is an underlying *smooth* distribution, which we estimate via non-parametric kernel density estimation (KDE) [70, 71]. This assumes that there is a *continuum of execution styles* responsible for the structural variety of an action, and it allows to interpolate unseen, but likely, temporal structures.

We use KDE with Gaussian kernels whose bandwidth h is automatically set using Scott's factor [72]: $h = n_+^{-1/(a+4)}$. We obtain a continuous distribution $\mathcal{D}$ over inter-actom distances $\mathbf{\Delta} = (t_2 - t_1, \ldots, t_a - t_{a-1})$:

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

In practice, however, integrating over this distribution during localization is computationally expensive. Therefore, we propose to approximate it with a discrete one obtained 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 5 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 allows users to control a trade-off between the coarseness of the model of the temporal structure and its computational complexity. We use $s = 10$ for all actions in our experiments, as we observed this to be a good compromise between computational efficiency and localization accuracy (*cf.* Section 6.6).

4.3 Negative training examples

Our localization 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 videos, and filter out those intersecting the annotated positives. To be consistent with the localization stage at test time, we randomly sample actoms according to the learned temporal structure $\hat{\mathcal{D}}$.

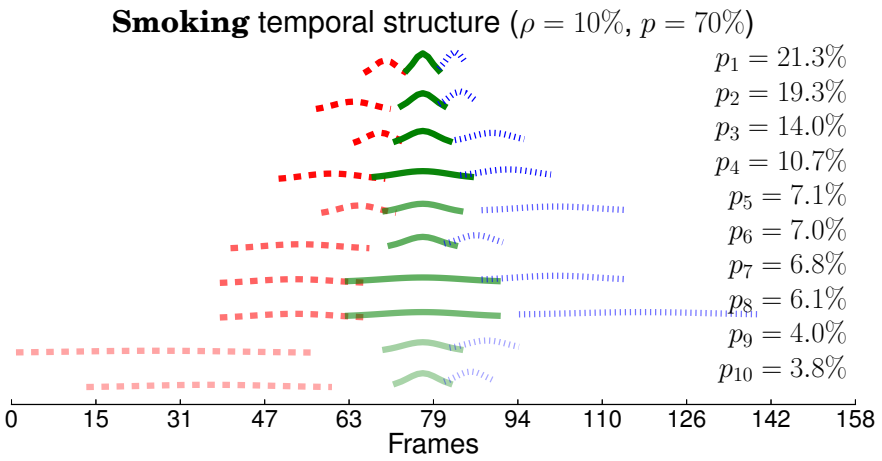


Figure 5: *Temporal structure learned for the "smoking" action from the "Coffee & Cigarettes" dataset. The candidate actom models* $(\hat{\mathbf{\Delta}}_j, p_j) \in \hat{\mathcal{D}}$ *are sorted by their estimated prior probability* p_j*.*

There are, however, several practical issues. First, the number of random negatives needed to learn a good detector is not a priori obvious. Second, the unlabeled part of the database from which these negatives are sampled might still contain an unknown number of positives that were not annotated. Indeed, the automatic techniques used to help in the acquisition of positive training data (*e.g.* [6, 46]) might miss many action examples. Therefore, randomly sampled windows have a non-negligible chance of containing the action of interest, and our negative examples might contain a significant number of false negatives compared to the number of true positives. Note that this also rules out the possibility to mine so-called "hard negative" examples for a re-training stage [73]. In practice, we sample only 2^m times more negatives than annotated positives, with $m \in \{0, 1, 2, 3, 4, 5, 6\}$ obtained by validation on held-out videos.

5 Localization with actoms

In this section, we describe our temporal localization 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 localization

To localize 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.

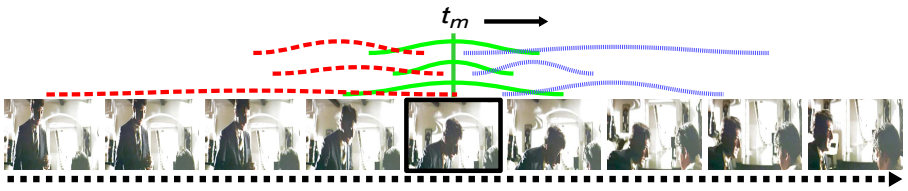


Figure 6: *Sliding central frame temporal localization. The probability of an action being performed at frame t_m is evaluated by marginalizing over all actom candidates learned with our model of the temporal structure (Eq. 8).*

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:

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

where f_{ASM} is the a posteriori probability estimate returned by our SVM classifier trained on ASM models (Eq. 5). See Figure 6 for an illustration.

Alternatively, taking the maximum a posteriori allows to not only localize an action, but also its most likely temporal structure (*cf.* Figure 2). We have experimentally observed that, for localization, 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 (*cf.* Figure 5). Therefore, the redundancy in $\hat{\mathcal{D}}$ makes marginalizing over actom candidates robust to inaccurate actom placements.

Note that our approach differs from the usual multi-scale sampling heuristic [11]. An action's possible structures, durations, and their prior probabilities are, indeed, learned from the data and modeled by $\hat{\mathcal{D}}$. In contrast, traditional sliding-window approaches manually specify a fixed set of window sizes, with no prior probability distribution.

5.2 Post-processing

Our algorithm returns a probability of localization 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 localization 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 unique scale per action category, which only depends on the largest actom spacings in $\hat{\mathcal{D}}$.

As the temporal shift between two localizations can be small in practice, we also use a non-maxima suppression algorithm to remove overlapping localization 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 localization

Although designed for temporal localization, 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 localization, 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 localization scores to provide a global decision for the clip. In our experiments, we found that max-pooling — *i.e.* taking the best localization score as classification score — yields good results. Indeed, marginalizing over actom candidates limits the number and the score of spurious false localizations, 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.

6.1 Datasets

We use two challenging movie datasets for action localization the "Coffee and Cigarettes" dataset [11] and the "DLSBP" dataset [8]. We also use the "Hollywood 2" dataset [12] for our classification by localization experiments. These datasets are originally provided with annotations in the form of temporal boundaries around actions.

"Coffee and Cigarettes" [11]. 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" [8]**.** 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 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 localization, *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" [12]. 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. This benchmark contains particularly challenging video conditions due to large visual variability across movies.

6.2 Evaluation criteria

For temporal localization, 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 [8, 11, 74], 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 start and end frame annotations provided by the dataset authors. This criterion, however, does not guarantee that a localization will contain enough of the action to be judged relevant by a user. For instance, a localization 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*. The OVAA criterion comes from the definition of actoms as the minimal set of subevents needed to recognize an action. Hence, a correct localization 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 in practice.

We use both criteria in our localization 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 localizations are considered as false positives. This is similar to the evaluation of object detection, *e.g.* in the Pascal VOC challenge [75]. Note that for classification by localization, 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. For localization experiments, we additionally

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 [9]. 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 (*cf.* 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 step to the windows, similar to the one described in Section 4.2, and commonly used in the literature, *e.g.* in [74].

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 Localization results

We report temporal localization results in Table 1 for the "Coffee and Cigarettes" dataset and in Table 2 for the "DLSBP" dataset. We compare our method (ASM), two baselines (BOF and BOF T3), and recent state-of-the-art results. We report the mean and standard deviation of the performance over five independent runs with different random negative training samples. Figure 7 shows frames of the top five results for "drinking" and "open door" obtained with our method. Some examples of automatically localized actoms with our ASM method are depicted in Figure 2.

Figure 7: *Frames of the top 5 actions localized with ASM for "Drinking" (top row) and "Open Door" (bottom row).*

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 localizing 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 supports our claim that the variable temporal structure of actions needs to be represented with a flexible model that can adapt to different durations, speeds, and interruptions.

Comparison to the state of the art. The method of Laptev and Pérez [11] is designed for spatio-temporal localization, which is a more difficult problem. However, they also rely on stronger supervision in the form of spatio-temporally localized actions. We compare to the mapping of their results to the temporal domain as reported in [8] (*cf.* row "LP-T" in Table 1). Similarly, Kläser *et al.* [74] use a human tracker trained on additional data, and learn from spatio-temporally localized training examples. The mapping of their results to the temporal domain are reported in the "KMSZ-T" row of Table 1. On the "DLSBP" dataset, we compare to the "ground truth" results of the authors in [8]. The differences between their approach and our BOF baseline lie mostly in the negative training samples and in the visual vocabulary.

According to our experiments, ASM consistently outperforms these approaches, including the spatio-temporal localization methods trained with more complex supervision, like bounding boxes (+14% with respect to LP-T [11]), or human tracks (+4% and +7% with respect to KMSZ-T [74]). This suggests that accurately modeling the temporal structure of actions can boost localization performance.

Method	"Drinking"	"Smoking"
matching criterion: OV20		
DLSBP [8]	40	N.A.
LP-T [11]	49	N.A.
KMSZ-T [74]	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)

Table 1: *Action localization results on "Coffee and Cigarettes", in Average Precision.*

Method	"Open Door"	"Sit Down"
matching criterion: OV20		
DLSBP [8]	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)

Table 2: *Action localization results on the "DLSBP" dataset, in Average Precision.*

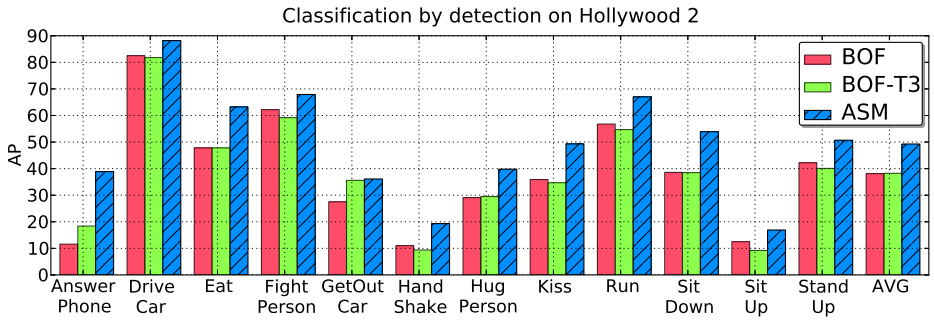


Figure 8: *Classification by localization results, in Average Precision (AP), on the "Hollywood 2" dataset [12]. "BOF" and "BOF-T3" are sliding-window approaches using BOF and its temporally structured extension. Our approach is "ASM". "AVG" contains the average performance over all classes (BOF:* 38.1%*, BOF T3:* 38.2%*, ASM:* 49.3%*).*

6.5 Classification-by-localization results

Figure 8 contains the per class classification-by-localization results on the "Hollywood 2" dataset. The BOF baselines are using the same sliding window approach as in the previous localization results.

On average over all classes, ASM improves by +11% over both BOF baselines, which have similar performance. The improvement yielded by ASM is most noticeable on 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 support our view 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 approximately the same results of 45% AP for three different models: BOF, BOF T3, and ASM with uniformly spread actoms. Note that the similar performance of all three global models seem to indicate that the benefits of ASM do not only lie in its use of soft-voting (*cf.* Section 6.6 for a more in-depth analysis of the contributions of each stage of our method). In comparison, ASM with classification-by-localization achieves 49% AP. This +4% gain is less significant than for temporal localization. This is due to the fact that classification of pre-segmented videos is an easier problem. Indeed, classification with BOF improves by +9% over the classification-by-localization results with the same BOF model. In addition, global models use context information, whereas the more local representations used for localization focus only on the action.

Finally, our performance is significantly lower than the current state of the art ($AP = 58.3\%$) [10], which uses a global BOF model with densely sampled local trajectories (tracklets) and a combination of multiple descriptors. How-

ever, our ASM model can also straightforwardly incorporate these features and multiple descriptors, in the same manner as described in [10], for an expected performance boost comparable to the one observed for the BOF model (+13.1% by switching from sparse spatio-temporal interest points to dense tracklets, all other things being equal).

6.6 Parameter study

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

ASM parameters. In Table 3, we report localization results for ASM with learned parameters and for different parameter configurations. These results seem to indicate that *learning action-specific ASM overlap and peakyness parameters yields the most accurate models*, resulting in increased localization performance. Note that the learned parameters change significantly from one action to the other. For instance, the learned parameters for "Smoking" are $\rho = 10\%$ and $p = 70\%$, denoting clearly separated actoms, whereas for "Sit Down" we obtain $\rho = 60\%$ and $p = 50\%$, denoting actoms sharing a significant amount of frames.

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 localization results on the "Coffee and Cigarettes" (C&C) and "DLSBP" datasets.*

Temporal model complexity. We studied the impact of the complexity of the estimated temporal structure, measured by the support size s of $\hat{\mathcal{D}}$ (*cf.* Equation 7). 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 find that using 10 actom structures yields a good compromise for most classes, and *using more structures does not increase localization performance* (observed differences are not statistically significant), while significantly increasing localization time. See Figure 9 for an illustration

on average over the categories from the "DLSBP" dataset. Note that if $s < 5$, then the model is too simple, and the performance gap between the OV20 and OVAA results is large.

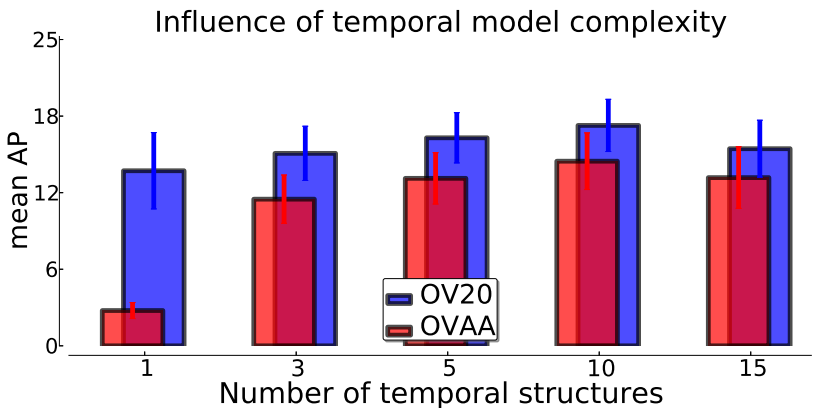


Figure 9: *Average localization performance of ASM on "DLSBP" v.s. number of candidate temporal structures.*

Sliding central frame. In Table 4, we report the localization 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 localization is performed by marginalizing over this 1D distribution on temporal extents. In contrast, sliding window also uses multiple scales learned from the training data, but it does not leverage a learned prior on those scales. Our results suggest that *our sliding central frame approach consistently outperforms the sliding window one*. Note also that ASM significantly outperforms all BOF baselines, even those with sliding central frames.

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

Number of actoms and manual annotations. We compared ASM with $a \in \{1, 2, 3, 4, 5\}$ actoms, by sub-sampling or interpolating training actoms from the manually annotated ones ($a = 3$). For $a = 1$, we use the middle actom only. The actom's range is then defined as the extent of the window surrounding all three actoms. For $a = 2$, we use a actoms 1 and 3. For $a \in \{4, 5\}$, we linearly interpolate between annotated actoms to generate the additional ones. We report the average results over all classes and datasets of these experiments in Figure 10 (AP for different overlap ratios) and Figure 11 (precision-recall curves with 20% and 50% overlap). In addition, these figures contain localization results using our ASM approach with training actoms spread *uniformly* between the original manually annotated temporal boundaries.

According to our experiments, there is no statistically significant difference between the results of BOF and ASM with one actom when using the sliding central frame approach: t-test p-values for equality of the means are above 10% for all actions and across all overlap ratios (see also "BOF scfr" *v.s.* "ASM $a = 1$" in Figure 10 and Figure 11). This indicates that *all frames are important to include in the model*, including the ones at the beginning and ending of the action. Therefore, *it is not the temporal weighting that is, by itself, responsible for the performance gains observed*.

In fact, our experiments (*cf.* Figure 10 and Figure 11) suggest that the main sources of improvement are the ones described below (performance is reported in mean average precisions over all actions and matching overlap ratios).

(1) Using **manually annotated actoms**. Best result with uniform actoms (ASM with two actoms): 12.2%, whereas with manual ones (ASM with three actoms): 19.7% (+7.5%). ASM with uniform actoms yields results similar to "BOF T3" with the sliding central frame approach. This suggests that *temporal boundaries do not provide enough information to model the temporal aspects of an action*.

(2) Using a **sequence of parts**. Best performance with one part (BOF scfr): 13.4%, whereas with multiple parts (ASM with three actoms): 19.7% (+6.3%). Note that the differences between using three, four, or five actoms are not statistically significant (equality t-test p-values are larger than 5% in most cases). In contrast, using only two actoms is, on average, significantly worse than using three or more actoms (p-values lower than 0.01%). Therefore, it seems that *as long as the manually labeled (meaningful) actoms are part of the ASM model*, adding more actoms to get a finer-grained temporal structure is not helpful.

(3) Using our **ASM model**: +4.5% improvement *w.r.t.* the best baseline (BOFT3 scfr). This increase in performance can be observed across all overlap ratios (*cf.* Figure 10), and across almost all recall rates for various matching overlap criteria (*cf.* Figure 11).

(4) **Marginalizing over a learned prior of the temporal structure** (sliding central frame). Gains from sliding window to sliding central frame: +3.2% for BOF-T3 (+5.2% for BOF). See also Table 4.

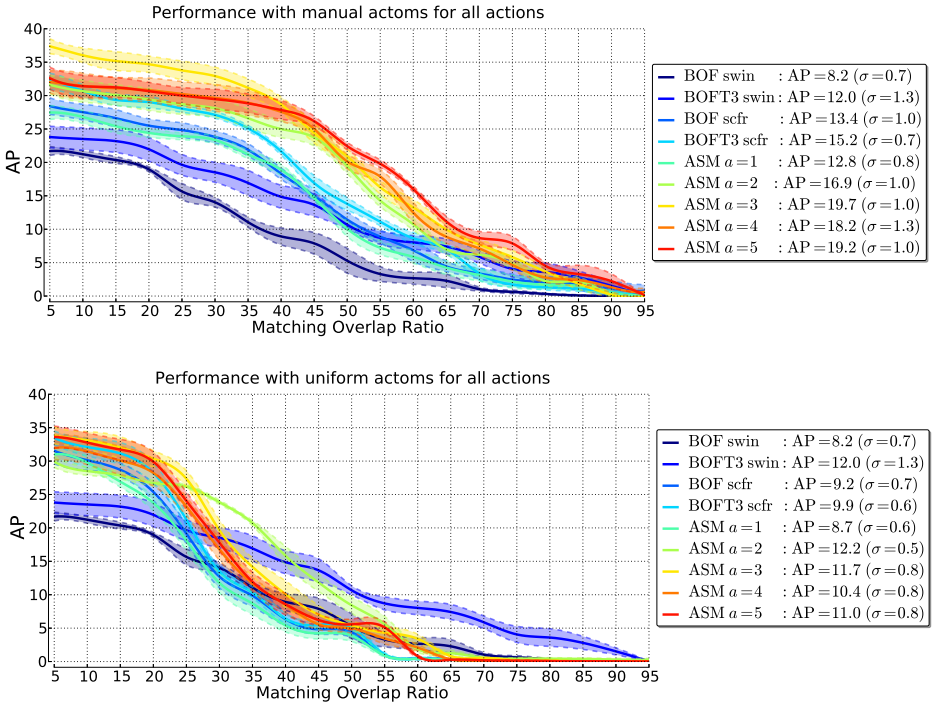


Figure 10: *Average Precisions (AP) for different overlap ratios. Each tube denotes the mean and standard deviation of the AP over all action categories from the Coffee & Cigarettes and DLSBP datasets. We use the* manual *or* uniform *actom annotations, and compare (i) the baselines ("BOF", "BOF-T3") with sliding window ("swin"), (ii) BOF and BOF-T3 using our sliding central frame approach ("scfr"), and (iii) our ASM approach with $a \in \{1, 2, 3, 4, 5\}$ actoms. The numbers in the legend (AP, σ) denote the average and standard deviation of the performance of each method over all overlap ratios.*

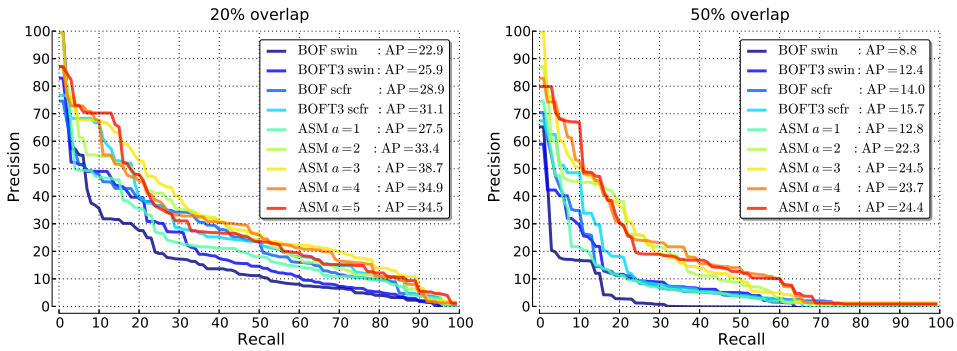


Figure 11: *Average precision-recall curves obtained over all actions for two overlap ratios.*

7 Conclusion

In this paper, we introduced the Actom Sequence Model (ASM). This model describes an action with a temporal sequence of user-defined temporal parts called actoms. It is discriminative, as it represents an action by several components instead of one average representation like 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. We also describe a sliding central frame approach for localization that is based on a generative model of an action's temporal structures learned at training time. Experimental results on real-world video data suggest that our approach outperforms the bag-of-features, its extension with a fixed temporal grid, and the state of the art.

Future work includes the design of actom representations based on human tracks. Such a description would eliminate background clutter, focus on the action, and allow to differentiate between multiple actors. In addition, reducing the annotation cost at training time would allow to scale our approach to more actions and to actions with more actoms. Another direction of research consists in using ASM as a building block of more complex representations, *e.g.* hierarchical decompositions [76], in order to model the temporal aspects of actions composing high-level activities like "cooking".

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. K. Aggarwal and M. S. 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] J. C. Niebles, H. Wang, and L. Fei-Fei, "Unsupervised learning of human action categories using spatial-temporal words," *IJCV*, 2008.

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

[9] S. Satkin and M. Hebert, "Modeling the Temporal Extent of Actions," in *ECCV*, 2010.

[10] H. Wang, A. Kläser, C. Schmid, and L. Cheng-Lin, "Action Recognition by Dense Trajectories," in *CVPR*, 2011.

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

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

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

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

[15] N. M. Oliver, B. Rosario, and A. P. Pentland, "A Bayesian computer vision system for modeling human interactions," *PAMI*, 2000.

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

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

[18] C. C. Chen and J. K. Agarwal, "Modeling Human Activities as Speech," in *CVPR*, 2011.

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

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

[21] K. Tang, L. Fei-Fei, and D. Koller, "Learning latent temporal structure for complex event detection," in *CVPR*, 2012.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

[47] A. Patron-Perez, M. Marszałek, A. Zisserman, and I. D. Reid, "High five: Recognising human interactions in TV shows," in *British Machine Vision Conference*, 2010.

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

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

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

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

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

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

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

[55] 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.

[56] B. Schölkopf and A. J. Smola, *Learning with Kernels*, 2002.

[57] J. C. Niebles, C. Chen, , and L. Fei-Fei, "Modeling Temporal Structure of Decomposable Motion Segments for Activity Classification," in *ECCV*, 2010.

[58] M. Raptis, I. Kokkinos, and S. Soatto, "Discovering Discriminative Action Parts from Mid-Level Video Representations," in *CVPR*, 2012.

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

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

[61] 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.

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

[63] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom Sequence Models for Efficient Action Detection," in *CVPR*, 2011.

[64] I. Laptev, "Spatio-Temporal Interest Point library," 2011. [Online]. Available: www.di.ens.fr/~laptev/interestpoints.html

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

[66] S. Maji, A. Berg, and J. Malik, "Classification Using Intersection Kernel Support Vector Machines is efficient," in *CVPR*, 2008.

[67] Y. Lin, G. Wahba, H. Zhang, , and Y. Lee, "Statistical Properties and Adaptive Tuning of Support Vector Machines," *Machine Learning*, 2002.

[68] J. Platt, "Probabilistic outputs for support vector machines," *Bartlett P. Schoelkopf B. Schurmans D. Smola, AJ, editor, Advances in Large Margin Classifiers*, 2000.

[69] H. T. Lin, C. J. Lin, and C. Weng, "A note on Platt's probabilistic outputs for support vector machines," *Machine Learning*, 2007.

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

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

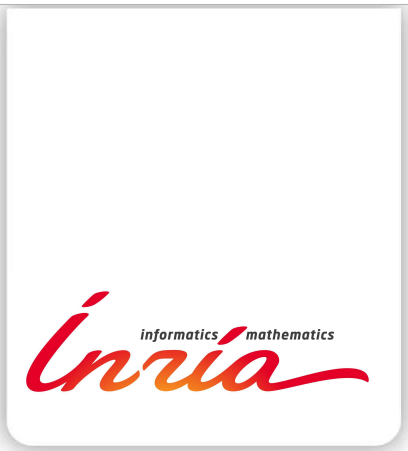
[72] D. W. Scott, *Multivariate density estimation: theory, practice, and visualization*. Wiley, 1992.

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

[74] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman, "Human Focused Action Localization in Video," in *SGA*, 2010.

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

[76] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," in *BMVC*, 2012.



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