Actom Sequence Models for Efficient Action Detection

Adrien Gaidon, Zaid Harchaoui, Cordelia Schmid

Actom Sequence Models for Efficient Action Detection

Adrien Gaidon Zaid Harchaoui Cordelia Schmid
LEAR - INRIA Grenoble, LJK
http://lear.inrialpes.fr/people/*last-name*

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 characteristic for the action. Our model 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. We refer to our model as Actom Sequence Model (ASM). Training requires the annotation of actoms for action clips. 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" [12] and the dataset of [3]. We show that our ASM method outperforms the current state of the art in temporal action detection.

1. Introduction

We propose an approach for finding *if and when* an action is performed in a large video database. We focus on actions of short duration (typically a few seconds), like sitting down or opening a door, and several hours of real-world video data (*e.g.* movies). Our approach is based on decomposing actions into sequences of key atomic action units, which we refer to as *actoms*.

Many actions can be naturally defined in terms of a composition of simpler blocks (see figure 1). We can observe that the displayed actions are easy to recognize given such a sequential description. Obtaining this decomposition of actions in atomic units is challenging, especially in realistic videos, due to viewpoint, pose, appearance and style variations. Furthermore, these atomic action units are action dependent and can be motions, poses or other visual patterns [20]. Therefore, many successful approaches for real-world video data avoid this decomposition and model an action as a spatio-temporal cuboid represented with a global bag-of-features (*e.g.* [3,11,19]). However, these models discard the temporal information inherent to actions and are, thus, not well adapted to distinguish actions with several motion or posture changes (*e.g.* opening a door) and symmetric actions (*e.g.* sitting down *vs.* standing up). Discriminative models are of particular importance in the context of action detection, where searching through a large volume of data for actions of short duration can result in many false alarms.

Figure 1. Examples of actom annotations for two actions.

Following recent work [17, 18, 20], we argue that actions are by essence temporal phenomena and modeling their temporal structure leads to a more discriminative representation. As observed by Schindler and Van Gool [20], it is important to understand how actions can be decomposed into the right basic units. In this work, we propose to model an action as a small sequence of *actoms*. These action atoms are key components of short duration, whose sequence is characteristic of the action.

We make the following contributions. First, we introduce a temporally structured representation of actions in videos, called Actom Sequence Model (ASM) (section 2). Our representation encodes in a flexible way the temporal ordering constraints between actoms. Actoms are specific to each action class and obtained by manual annotation. We observed that the annotation cost for actoms is comparable to specifying beginning and end of a training clip. Note that manual annotations are required at training time only. Second, we propose a simple yet efficient parameter-free

approach for multi-scale detection, based on a non parametric generative model of inter-actom spacings (section 3). We show that our approach outperforms the state of the art on two recent challenging benchmarks for action detection from [3, 12] (section 4).

1.1. Related work

Previous work on incorporating temporal information into action models have often focused on simple datasets where the video conditions are controlled. Nowozin *et al*. [18] use a sequence of individual local features which are assigned to an arbitrarily fixed number of uniformly sized temporal bins. Based on the popular work of Viola and Jones [24], Jung *et al*. [8] propose a two-stage feature selection procedure using Adaboost to obtain local spatio-temporal regions of interest. Schindler and Van Gool [20] model actions as the concatenation of per-frame shape and motion features and show that only a few contiguous frames are necessary for recognition. These models are not flexible and even in simple controlled video settings (*e.g*. on KTH [21]), they only perform similarly or worse than a global bag-of-features representation.

Some approaches deal with realistic video data, *i.e*. with challenging uncontrolled video conditions. In Laptev *et al*. [11] bag-of-features with spatio-temporal pyramids for different, manually selected, regular spatial and temporal grids are combined in a multi-channel Gaussian kernel. Such an approach is shown to improve over the standard bag-of-features, but the temporal structure of actions is not explicitly modeled and, thus, not adapted to the action. In Duchenne *et al*. [3] and Satkin and Hebert [19], it is observed that temporal boundaries of actions are not precisely defined in practice, whether they are obtained automatically using "weak" supervision or by hand. Both of these works deal with realistic data and improve the quality of annotated action clips by automatically refining their temporal boundaries to obtain smaller sub-clips, which they represent using an orderless bag-of-features. However, they only model the temporal extent of actions, not their temporal structure.

Related to our work, Niebles *et al*. [17] observe that temporal information is important to understand motion and discover motion parts based on a latent model [6]. Their model is tailored to the classification of long duration activities (*e.g*. "triple-jump"), represented by a composition of actions described with a bag-of-features. In contrast to our model, their representation is not sequential, but they only induce a loose hierarchical structure by introducing a temporal displacement penalization factor in part comparisons. Such an approach is not well adapted to short actions, as illustrated by their results. Also dealing with long-term activities, Laxton *et al*. [13] use dynamic Bayesian networks with a Viterbi-like inference algorithm on manually designed, complex, per-activity hierarchies of predefined contextual cues and object detectors to recognize complex interactions with objects.

The majority of the existing approaches focus on the problem of classification of already temporally segmented video clips. However, detection algorithms are more relevant for many applications (*e.g*. robotics, retrieval, video-surveillance) [9, 19]. This is a more challenging problem and current state-of-the-art methods [3, 9, 12] use sliding window approaches, which rely on multi-scale heuristics. In contrast, our method proposes a principled and efficient solution by marginalizing over all possible actom sequences for a given test video.

Our method is similar in spirit to state-of-the-art approaches for facial expression recognition from videos. Facial expression recognition is currently performed using label information defined by the Facial Action Coding System (FACS) [4], which segments facial expressions into predefined "action units", complemented with temporal annotations such as *onset*, *peak*, *offset*. However, except *e.g*. [23], most approaches only use "peak" frames for classification [1]. We propose here a principled framework to handle class labels as well as temporal labels for action recognition.

2. Modeling actions as sequences of actoms

An action is decomposed into a small, temporally ordered sequence of 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 mechanism (described in section 2.1) of temporally-localized features (see section 2.2). We model an action as a sequence of actoms by concatenating the per-actom representations in temporal order to form our sparse sequential model, called Actom Sequence Model (ASM), see Figure 2 for an illustration. We explain how we obtain training data for the ASM in section 2.3.

2.1. The Actom Sequence Model

We define the time-span of an actom with a *radius*, measured in number of frames 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 $s = (t_1, \cdots, t_n)$ with $n \geq 2$ is parametrized by the amount of overlap $\rho \in [0, 1]$, between adjacent actoms:

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

where d_i is the distance (in frames) to the closest actom:

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

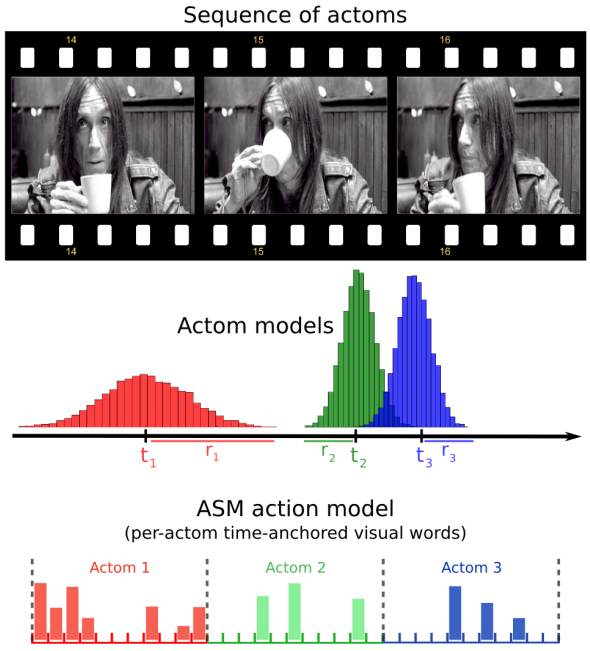

Figure 2. Construction of our ASM action model using actom-based annotations and a temporal weighting scheme for aggregating local features in a sparse temporally structured bag-of-features.

This defines a symmetric neighborhood around each temporal location that is specific to each actom of each action. Visual features are computed only within the forward and backward time range defined by the actom's radius. They are, then, accumulated in per-actom histograms of visual words, similar to the traditional bag-of-features approach (see section 2.2 for more details).

Defining the actom's time-span relatively to its closest neighbour has the advantage of either (i) allowing adjacent actoms to overlap and share features while ensuring the temporal order, which makes the model robust to inaccurate temporal localization of actoms, or (ii) to have gaps between actoms, which allows to represent non-continuous actions. An adaptive time-span makes the model naturally robust to variable action duration and speed as well as to temporal discontinuities. The only assumption we make is with respect to the temporal ordering of the actoms.

We also introduce a *temporally weighted assignment scheme*. As opposed to the usual time-independent binning process, we propose to aggregate weighted contributions of per-actom features, where each feature at temporal location t falling in the vicinity of actom $a_i = (t_i, r_i)$ (*i.e.* if $|t - t_i| \leq r_i$) is weighted by its temporal distance to the actom:

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

Hence, the more the time-stamp of a feature is remote from the time-stamp of its corresponding actom, the more its weight in the soft assignment scheme decreases.

This scheme offers an intuitive way to tune the bandwidth σ of the weighting window, through a coefficient $0 < p < 1$, which can be interpreted as a probability using the Chebyshev inequality for a random variable X of mean μ and finite standard deviation σ: $\mathbf{P}(|X - \mu| \geq k\sigma) \leq 1/k^2$, for any $k > 0$. Rewriting this equation with $X = t$, $\mu = t_i$ and $r_i = k\sigma$, we obtain:

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

The parameter p can be interpreted as the "peakyness" of our soft-assignment scheme. It allows to encode a prior on the amount in % of probability mass of features falling in the actom time range. Thus, the larger p, the smaller the σ values will be, *i.e.* the more centered the weighting will be, while smaller p means "flatter" Gaussians for the soft voting scheme in Equation 2.

To summarize, from a sequence of actom locations $s = (t_1, \cdots, t_n)$, we derive our *ASM representation* by (i) computing visual features only in the time-span of those actoms (defined by their radii parametrized by the ρ parameter in Equation. 1), (ii) then computing their contributions (defined by Equation. 2 and the parameter p) to per-actom temporally weighted histograms and (iii) appending these histograms into a temporally ordered sequence which is our ASM representation of videos (cf. Figure 2): $\mathbf{x} = (x_{1,1}, \cdots, x_{1,k}, \cdots, x_{n,1}, \cdots, x_{n,k})$, where

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

is the sum over the i^{th} actom's time-span $[t_i - r_i, t_i + r_i]$ of the weighted number $c_j(t)$ of local features of frame t assigned to visual word j. The ASM vector $\mathbf{x}$ is then L_1-normalized. In the following section, we describe the local features we use to represent visual information inside actoms.

2.2. Local visual information in actoms

Following recent results on action recognition in challenging video conditions [3, 11], we use sparse space-time features [10] to represent local motion and dynamic appearance. Using a multi-scale space-time extension of the Harris operator, we detect spatio-temporal interest points (STIPs), that we represent with a concatenation of coarse histograms of oriented gradient (HoG) and optical flow (HoF). The parameters we use are similar to the settings of [25].

Once a set of local features has been extracted, we quantize them using a visual vocabulary. In our experiments, we cluster a subset of 100,000 features randomly sampled from the training videos. Similar to [3], we use the k-means algorithm with a number of clusters set to $k = 1000$. We then assign each feature to the closest (using Euclidean distance) visual vocabulary word.

2.3. Actom annotations

Actoms are action specific and it is therefore impossible to build an *a priori* universal vocabulary of actoms. Here, actoms are annotated manually for a set of training actions[1]. An actom is annotated by a timestamp in the corresponding video. This temporal location is selected such that it contains, in the time range defined by its surrounding frames, static visual information (*e.g.* pose or object appearance) and/or dynamic information (motion) that is representative of a part of the action. Annotations are manually obtained only for the positive training examples. See Figure 1 for examples of actom frames. During the annotation process, we ensure a semantic consistency in the choice of actoms across different scenes where the same action is observed. This means that the n-th actom of an action has exactly one interpretation, like "recipient containing liquid coming into contact with lips" for the drinking action. The definition of actoms given as annotation guidelines can be obtained either from a dictionary or from an expert (*e.g.* for a sport action like a tennis serve). Note that an actom is a visual phenomenon that is semantically relevant and not a learned discriminative part of the action. It is therefore possible to interpret a predicted actom sequence.

Such annotations have the following practical advantages. First, we observed that only a few actoms are necessary to characterize an action. We used three actoms per action example in our experiments. Second, these annotations can be obtained in the same amount of time (linear in the number of frames) as temporal boundaries. Third, it is easier to obtain consistent actom annotations than precise action boundaries. Indeed, the temporal extent of actions is generally difficult to determine precisely. For instance, a person walking towards a door before opening it may or may not be considered as a part of the action "Open Door". In addition, inaccurate boundary annotations significantly degrade the recognition performance (as outlined by [3] for detection and [19] for classification). 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 3 quantitatively illustrates this claim by showing that the ground truth annotations for the action "sitting down" have a smaller variance, with respect to action duration, when actoms are annotated instead of beginning and end frames. In addition, we observed that the distribution of actoms are much more similar between the training and test sets. Note, that we annotated ground truth test actoms only to compute detection performance in terms of actoms detected (*c.f.* section 4). As described in the following section, our detection algorithm does not require manually annotated test actoms, but automatically detects them.

[1] Annotations available at http://lear.inrialpes.fr/data.

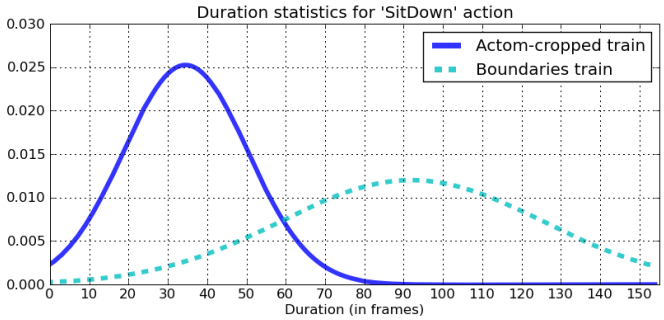

Figure 3. Frequencies of action durations obtained from manual annotations for the action "sitting down". "Actom-cropped" represents the durations when cropped according to actoms. "Boundaries" depict the duration of ground truth annotations from [3], obtained by labeling beginning and end frames of the action.

3. Temporal action detection framework

In order to detect if and when a target action is performed in hours of video, we train an action classifier for our ASM descriptor (described in section 3.1) and detect actions in test video sequences by using automatically generated test actom sequences as explained below (section 3.2).

3.1. Action classifier

Binary action classification consists of classifying descriptions of videos as belonging to an action category or not. A state-of-the-art machine learning tool to solve this problem efficiently is Support Vector Machines (SVMs). We use the LIBSVM library with appropriate class balancing [15] and probability outputs [14].

Kernels on actions. In conjunction with SVMs, we use the *intersection kernel between histograms* [16], defined for any $x = (x_1, \ldots, x_N)$ and $x' = (x'_1, \ldots, x'_N)$ as $K(x, x') = \sum_{j=1}^{N} \min(x_j, x'_j)$.

Negative training examples. As mentioned previously, positive training examples for an action are obtained by manual annotation. Recent experimental results, *e.g.* in [7], suggest that rather than using positive examples only, it is beneficial to use "weak" negative training examples for better discriminative performance. We adopt here the simple strategy of [7], which consists in randomly sampling clips of temporal extent comparable to the positive training examples. Though we exclude the annotated training positives, this strategy might still generate a significant number of false negatives as (i) we are interested in common actions that might frequently occur in the training videos, (ii) only a few of these samples are annotated as training positives in order to limit the annotation effort (the rest is missed). Hence, in contrast to the usual approach (see [3]), we propose to sample only a few random negative clips. This prevents the dataset from including too many false negative examples which could harm the recognition performance,

while also leading to a smaller computation cost. In practice, we observed that 300 negative examples, that is between 3 and 6 times the number of positive examples, gave good results for all of our experiments. Note, that this rules out the possibility to mine so-called "hard negative" examples for a re-training stage [2], as we do not know beforehand which examples are truly negative.

3.2. Detection with actoms

To detect actions in a test sequence, we apply our SVM classifier in a sliding window manner. However, instead of sliding a temporal window of some fixed scale, we shift the *temporal location of the middle actom* t_m, where $m = \lfloor n/2 \rfloor$ and n is the number of actoms for the action category. As an action is of short duration w.r.t. the whole duration of the test sequence, we use a small temporal shift (5 frames in our experiments). We, then, build our ASM representation by automatically generating candidate actom sequences centered on t_m.

Generative temporal model. We first learn a generative model of the temporal structure by estimating the distribution of relative actom spacings from the training sequences: $\{\mathbf{\Delta}_i = (t_{i,2} - t_{i,1}, \ldots, t_{i,n} - t_{i,n-1}), i = 1 \ldots N^+\}$, where n is the number of actoms of the action category and N^+ is the number of positive localized training actom sequences $(t_{i,1}, \ldots, t_{i,n})$. We use a non-parametric kernel density estimation algorithm with Gaussian kernels whose bandwidth h is automatically set according to Scott's factor [22]. We choose this method instead of histograms because it has the advantage of yielding a smooth estimate, and instead of Gaussian mixture models, which require parameter tuning. We obtain a multivariate continuous distribution of density:

$$\mathcal{D} \sim \frac{1}{N h^{n-1}} \sum_{i=1}^{N} \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{||\mathbf{\Delta} - \mathbf{\Delta}_i||^2}{2h^2}\right) \quad (5)$$

over inter-actom distances $\mathbf{\Delta} = (t_2 - t_1, \ldots, t_n - t_{n-1})$. As we deal with discrete time steps (frames), we then discretize this distribution in the following way. We first sample 10,000 points from our estimated density. We, then, quantize them, using a quantization step equivalent to our sliding window step-size. We compute the corresponding discrete multi-variate distribution by using histograms, truncate its support by removing outliers (points with a probability smaller than 2%) and re-normalize the probability estimates. This results in a discrete multi-variate distribution

$$\hat{\mathcal{D}} = \{(\hat{\mathbf{\Delta}}_j, \hat{p}_j)\,, \;\; j = 1 \cdots K\}, \quad \hat{p}_j = \mathbf{P}(\hat{\mathbf{\Delta}}_j) \quad (6)$$

with a support limited to only a few likely candidate actom spacings $\hat{\mathbf{\Delta}}_j = (\hat{t}_{j,2} - \hat{t}_{j,1}, \cdots, \hat{t}_{j,n} - \hat{t}_{j,n-1})$. In practice, we obtain $K \approx 10$.

This distribution $\hat{\mathcal{D}}$ is learned at training time and is used to generate candidate actom sequences at test time. For a given central actom location t_m, we use our generative model of inter-actom spacings $\hat{\mathcal{D}}$ to compute the probability of an action occurring at this temporal location by marginalizing over all candidate actom sequences:

$$\begin{aligned} & \mathbf{P}(\text{action at } t_m) \\ = \; & \textstyle\sum_{j=1}^{K} \mathbf{P}(\text{action at } t_m \mid \hat{\mathbf{x}}_\mathbf{j})\, \mathbf{P}(\hat{\mathbf{x}}_\mathbf{j}) \\ = \; & \textstyle\sum_{j=1}^{K} f_{\text{ASM}}(\hat{t}_{j,1}, \ldots, t_m, \hat{t}_{j,n})\, \hat{p}_j \end{aligned} \quad (7)$$

where f_{ASM} is the a posteriori probability estimate returned by our SVM classifier trained on ASM models. This mechanism provides a principled solution to perform multi-scale detection, instead of the usual multi-scale sampling heuristic [12]. Furthermore, it can be used to detect not only an action, but also its actoms, by simply taking the maximum a posteriori $\text{argmax}_j\, f_{\text{ASM}}(\hat{t}_{j,1}, \ldots, \hat{t}_{j,n})\hat{p}_j$.

Detection window. When a time-range and not just a temporal location is required (*e.g.* for evaluation), we define a detection window centered on a detection at frame t_m. As we marginalize over $\hat{\mathcal{D}}$, we consider a window encompassing all frames used in the score computation. This defines a single scale per action category, which only depends on the largest actom spacings from $\hat{\mathcal{D}}$.

Non-maxima suppression. As the temporal shift between two detections is small in practice, we use a non-maxima suppression algorithm to remove close detections. We recursively (i) find the maximum of the scores and (ii) delete detection windows with lower scores, that temporally overlap with this maximum. Windows are considered as overlapping if the intersection over the union of the frames is larger than 20%.

4. Experiments

4.1. Datasets

We use two challenging movie datasets for action detection: the "Coffee and Cigarettes" dataset [12] and the "DLSBP" dataset [3].

The "Coffee and Cigarettes" dataset consists of a single movie composed of 11 short stories, each with different scenes and actors. The dataset was introduced by Laptev and Pérez [12]. They evaluated action localization for "drinking". Evaluation for "smoking" was later added by [9]. The training sets consists of 41 drinking and 70 smoking examples, from six different short stories of the "Coffee and Cigarettes" movie, 32 drinking and 8 smoking clips from the movie "Sea of love" and 33 drinking examples recorded in a lab. The test sets are two short stories (approx. 36,000 frames) for drinking, with a total of 38 drinking actions, and three short stories (approx. 32,000 frames) for smoking, with a total of 42 positive samples. Note that there is no overlap between the training and test sets, both in terms of scenes and actors.

The "DLSBP" dataset, introduced by Duchenne *et al*. [3], consists of two action categories "OpenDoor" and "SitDown". The training sets includes 38 "OpenDoor" and 51 "SitDown" actions sequences obtained from 15 movies. Three movies are used as test set, containing a total number of 91 "OpenDoor" actions and 86 "SitDown" ones.

All examples for both datasets are manually annotated with temporal boundaries. However, as previously mentioned in section 2.3, the annotations are often slightly inaccurate, generally beginning before the action and ending after it happened. Note that the chance level, *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.

4.2. Evaluation framework

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 [3, 9, 12], referred to as OV20, which states that a window matches a ground truth action, if they overlap (intersection over union) by more than 20%. Note that it is the same as our overlap ratio for non-maxima suppression. We use the original ground truth annotations (beginning and end frames) provided by the dataset authors.

However, this criterion is loose and as temporal boundaries are imprecise, it might quantify how well the temporal context is learned, instead of the action itself. Therefore, in conjunction to the previous one, we used a complementary, more restrictive 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*. Consequently, we also annotated actoms for the test positives to assess ground truth according to OVAA. These annotations are not used at test time.

Similarly to [3, 9, 12], temporal detection performance is measured in terms of precision and recall by computing the Average Precision (AP). Note, that 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, the others are considered as false positives, similarly as for object detection (*e.g*. in the Pascal VOC challenge [5]).

Regarding our model hyper-parameters, we used the same voting parameter value $p = 75\%$ and the same actom overlap ratio value $\rho = 75\%$ for our ASM models in all of our experiments (for all datasets and all actions). These values yield good results in practice and show the flexibility of our model, which can be discriminative, even though its parameters are not specifically tuned for each action.

4.3. Bag-of-features baselines

We compare our approach to two baseline models: the standard bag-of-features, and its extension with a regular temporal grid, both described below.

To make the results comparable, we use the same visual features, vocabularies and kernel as the ones used for our ASM model (*c.f*. section 2.2). For the positive training samples, we crop the original annotations 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 [19]. Furthermore, we use the same random training negative samples (300 in total) as the ones used by our ASM approach.

At test time, bag-of-features-based sliding window approaches require the *a priori* definition of multiple temporal scales. Similarly to our generative model of actom locations, we learned the scales from the training set. We obtained the following scales: $\{20, 40, \cdots, 100\}$ for drinking and smoking, and $\{20, 40, \cdots, 140\}$ for both opening a door and sitting down. Regarding the step-size by which the windows are translated, we used 5 frames for all of our experiments, including for our ASM-based approach. We finally apply a non-maxima suppression post-processing to the windows, similar to the one described in section 3.2 and commonly used in the literature (*e.g*. in [9]).

In addition to the global bag-of-features baseline (referred to as "BOF"), we evaluate an extension with regular temporal grids [11]. 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 (beginning, middle and end) of equal duration. Then, a "BOF" is computed for each part and the three histograms are concatenated. In the following, this method is referred to as "BOF T3".

4.4. Detection results

Detection results are reported in table 1 for the "Coffee and Cigarettes" dataset and in table 2 for the "DLSBP" dataset. We report detection results for our method (ASM), the two baselines (BOF and BOF T3) and recent state-of-the-art results. Figure 4 shows frames of the top 5 results for "drinking" and "Open Door" obtained with our method. Some examples of automatically detected actoms with our ASM method are depicted in figure 5.

For the "Coffee and Cigarettes" dataset, the method of Laptev and Pérez [12] (referred to as LP) is trained for spatio-temporal localization with stronger supervision in the form of spatially and temporally localized actions. We compare to the interpolation of spatio-temporal detection results obtained by Laptev and Pérez [12] to the temporal domain, as reported in [3]. These results are only available for the drinking action. Kläser *et al*. [9] additionally publish results for the smoking action. The method of [9], referred

Figure 4. Frames of the top 5 actions detected with our ASM method for the "drinking" (top row) and "Open Door" (bottom row) actions.

Method	"Drinking"	"Smoking"
matching criterion: OV20		
DLSBP [3]	40	NA
LP [12]	49	NA
KMSZ [9]	54.1	24.5
BOF	36 ($\pm$1)	19 ($\pm$1)
BOF T3	44 ($\pm$2)	23 ($\pm$3)
ASM	**57** ($\pm$3)	**31** ($\pm$2)
matching criterion: OVAA		
BOF	11 ($\pm$2)	1 ($\pm$0)
BOF T3	18 ($\pm$3)	4 ($\pm$1)
ASM	**50** ($\pm$5)	**22** ($\pm$2)

Table 1. Action detection results on the "Coffee and Cigarettes" dataset in Average Precision (in %). ASM refers to our method. Where possible, we report the mean and standard deviation of the performance over 5 independent runs with varying negative training samples.

Method	"Open Door"	"Sit Down"
matching criterion: OV20		
DLSBP [3]	13.9	14.4
BOF	12.2	14.2
BOF T3	11.5	17.7
ASM	**16.4**	**19.8**
matching criterion: OVAA		
BOF	9.9	5.8
BOF T3	5.1	13.1
ASM	**14.9**	**16.7**

Table 2. Action detection results on the "DLSBP" dataset in Average Precision (in %). ASM refers to our method.

to as KMSZ, is based on regular temporal grids inside human tracks. Note that their set-up is slightly different than ours, as they report results for spatio-temporal detection, that quantify the performance of both their human tracker and their temporal action detection approach restricted to human tracks. On the "DLSBP" dataset, we compare to the original results of the authors in [3]. They use a similar set-up to our BOF baseline. In the following, we discuss how our ASM model compares to both our bag-of-features baselines and the state of the art.

Comparison to bag-of-features. According to both evaluation criteria, we perform better than the bag-of-features baseline. The improvement is significant when using the OV20 criterion ($+21\%$ for drinking, $+12\%$ for smoking, $+4.2\%$ for "Open Door" and $+5.6\%$ for "Sit Down"). In contrast to our method, the BOF baseline sees its performance strongly degraded when changing the matching criterion from OV20 to the more restrictive OVAA (*e.g.* -25% for drinking). This proves that it learns more the temporal context than the action itself. Our ASM model, on the other hand, is more accurately detecting all action components and the relative gap in performance w.r.t. the baselines increases significantly when changing from OV20 to OVAA (*e.g.* from $+21\%$ to $+39\%$ for drinking and from $+12\%$ to $+21\%$ for smoking).

Rigid *vs*. adaptive temporal structure. Contrary to using ASM, incorporating a rigid temporal grid does not always improve over global bag-of-features. Though, in some cases, it can noticeably improve detection performance compared to "BOF", it is significantly outperformed by our adaptive model. This confirms that the temporal structure of actions varies and, therefore, needs to be represented with a flexible model that can adapt to different durations, speeds and interruptions. Thus, regular temporal grids are not well suited to represent temporal structure, whereas our method leverages temporal information and consistently improves performance over non-structured or rigidly structured methods.

Comparison to state-of-the-art. Our method clearly outperforms the state-the-art approaches, for all actions of the two datasets. For instance, on the drinking action, we gain up to $+17\%$ AP with respect to DLSBP [3]. On the more challenging "DLSBP" dataset, we improve by $+2.5\%$ for "Open Door" and by $+5.4\%$ for "Sit Down". Our ASM method even outperforms other methods trained with more complex supervision, like bounding boxes ($+8\%$ w.r.t. LP [12]) or human tracks with regular grids ($+3\%$ w.r.t. KMSZ [9]). This shows, that appropriately modeling the temporal structure of actions is crucial for performance.

Figure 5. Frames of automatically detected test actom sequences.

5. 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, our approach outperforms the state of the art, even if more complicated models with spatial localization are used. This illustrates the power of our temporal model.

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 also like to thank Ivan Laptev for the "DLSBP" dataset.

References

[1] 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. 3202

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 3205

[3] O. Duchenne, I. Laptev, J. Sivic, F. Bach, and J. Ponce. Automatic annotation of human actions in video. In *ICCV*, 2009. 3201, 3202, 3203, 3204, 3205, 3206, 3207

[4] P. Ekman and W. V. Friesen. *Facial Action Coding System*. Consulting Psychologists Press, 1978. 3202

[5] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The Pascal Visual Object Classes (VOC) Challenge. *IJCV*, 88(2):303–338, 2010. 3206

[6] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *PAMI*, 2009. 3202

[7] A. Gaidon, M. Marszałek, and C. Schmid. Mining visual actions from movies. In *BMVC*, 2009. 3204

[8] S. Jung, Y. Guo, H. Sawhney, and R. Kumar. Multiple cue integrated action detection. In *HCI*, 2007. 3202

[9] A. Kläser, M. Marszałek, C. Schmid, and A. Zisserman. Human focused action localization in video. In *SGA*, 2010. 3202, 3205, 3206, 3207

[10] I. Laptev. On space-time interest points. *IJCV*, 64(2–3):107–123, 2005. 3203

[11] I. Laptev, M. Marszalek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008. 3201, 3202, 3203, 3206

[12] I. Laptev and P. Perez. Retrieving actions in movies. In *ICCV*, 2007. 3201, 3202, 3205, 3206, 3207

[13] B. Laxton, J. Lim, and D. Kriegman. Leveraging temporal, contextual and ordering constraints for recognizing complex activities in video. In *CVPR*, 2007. 3202

[14] H. Lin, C. Lin, and C. Weng. A note on Platt's probabilistic outputs for support vector machines. *Machine Learning*, 38:267–276, 2007. 3204

[15] Y. Lin, G. Wahba, H. Zhang, , and Y. Lee. Statistical properties and adaptive tuning of support vector machines. *Machine Learning*, 4:115–136, 2002. 3204

[16] S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 3204

[17] J. C. Niebles, C.-W. Chen, , and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 3201, 3202

[18] S. Nowozin, G. Bakir, and K. Tsuda. Discriminative subsequence mining for action classification. In *ICCV*, 2007. 3201, 3202

[19] S. Satkin and M. Hebert. Modeling the temporal extent of actions. In *ECCV*, 2010. 3201, 3202, 3204, 3206

[20] K. Schindler and L. Van Gool. Action snippets: How many frames does human action recognition require. In *CVPR*, 2008. 3201, 3202

[21] C. Schüldt, I. Laptev, and B. Caputo. Recognizing human actions: a local SVM approach. In *ICPR*, 2004. 3202

[22] D. Scott. *Multivariate density estimation: theory, practice, and visualization*. Wiley, 1992. 3205

[23] T. Simon, M. Nguyen, F. De la Torre, and J. Cohn. Action unit detection with segment-based SVM. In *CVPR*, 2010. 3202

[24] P. Viola and M. Jones. Rapid object detection using a boosted cascade of simple features. In *CVPR*, 2001. 3202

[25] H. Wang, M. M. Ullah, A. Kläser, I. Laptev, and C. Schmid. Evaluation of local spatio-temporal features for action recognition. In *BMVC*, 2009. 3203