Automatic Annotation of Human Actions in Video

Olivier Duchenne, Ivan Laptev, Josef Sivic, Francis Bach and Jean Ponce
INRIA / École Normale Supérieure, Paris, France
{olivier.duchenne,josef.sivic,francis.bach,jean.ponce}@ens.fr ivan.laptev@inria.fr

Abstract

This paper addresses the problem of automatic temporal annotation of realistic human actions in video using minimal manual supervision. To this end we consider two associated problems: (a) weakly-supervised learning of action models from readily available annotations, and (b) temporal localization of human actions in test videos. To avoid the prohibitive cost of manual annotation for training, we use movie scripts as a means of weak supervision. Scripts, however, provide only implicit, noisy, and imprecise information about the type and location of actions in video. We address this problem with a kernel-based discriminative clustering algorithm that locates actions in the weakly-labeled training data. Using the obtained action samples, we train temporal action detectors and apply them to locate actions in the raw video data. Our experiments demonstrate that the proposed method for weakly-supervised learning of action models leads to significant improvement in action detection. We present detection results for three action classes in four feature length movies with challenging and realistic video data.

1. Introduction

Identifying human actions in video is a challenging computer vision problem and the key technology for many potential video mining applications. Such applications become increasingly important with the rapid growth of personal, educational, and professional video data.

Action recognition has a long history of research with significant progress reported over the last few years. Most of recent works, however, address the problem of action classification, i.e., "what actions are present in the video?" in contrast to "where?" and "when?" they occur. In this paper, similar to [7, 14, 21, 24] we aim at identifying both the classes and the temporal location of actions in video.

Recent papers on action recognition report impressive results for evaluations in controlled settings such as in Weizman [1] and in KTH [19] datasets. At the same time, state-of-the-art methods only achieve limited performance in real scenarios such as movies and surveillance videos as demonstrated in [13, 22]. This emphasises the importance of realistic video data with human actions for the training and evaluation of new methods.

Figure 1. Video clips with *OpenDoor* actions provided by automatic script-based annotation. Selected frames illustrate both the variability of action samples within a class as well as the imprecise localization of actions in video clips.

In this work we use realistic video data with human actions from feature length movies. To avoid the prohibitive cost of manual annotation, we propose an automatic and scalable solution for training and use movie scripts as a means of weak supervision. Scripts enable text based retrieval of many action samples but only provide imprecise action intervals as illustrated in Figure 1. Our main technical contribution is to address this limitation with a new weakly-supervised discriminative clustering method which segments actions in video clips (Section 3). Using the resulting segments with human actions for training, we next turn to temporal action localization. We present action detection results in highly challenging data from movies, and demonstrate improvements achieved by our discriminative clustering method in Sections 4 and 5.

1.1. Related work

This paper is related to several recent research directions. With respect to human action recognition, similar to [6, 13, 17, 19] and others, we adopt a bag-of-features framework, represent actions by histograms of quantized local space-time features and use an SVM to train action models. Our automatic video annotation is based on video alignment of scripts used in [5, 8, 13]. Similar to [13], we use scripts to find coarse temporal locations of actions in the training data. Unlike [13], however, we use clustering to discover precise action boundaries from video.

Unsupervised action clustering and localization has been addressed in [24] by means of normalized cuts [16]. Whereas this direct clustering approach works in simple settings [24], we find it is not well suited for actions with large intra-class variation and propose an alternative discriminative clustering approach. [17] deals with unsupervised learning of action classes but only considers actions in simple settings and does not address temporal action localization as we do in this work. Recently and independently of our work, Buehler *et al.* [2] considered learning sign language from weak TV video annotations using multiple instance learning.

Our weakly supervised clustering is also related to the work on learning object models from weakly annotated images [4, 15]. The temporal localization of training samples in videos, addressed in this work, is also similar in spirit to weakly supervised learning of object part locations in the context of object detection [9].

Several previous methods address temporal localization of actions in video. Whereas most of them evaluate results in simple settings [7, 21, 24], our work is more related to [14] that detects actions in a real movie. Differently to [14] our method is weakly supervised and enables the learning of actions with minimal manual supervision.

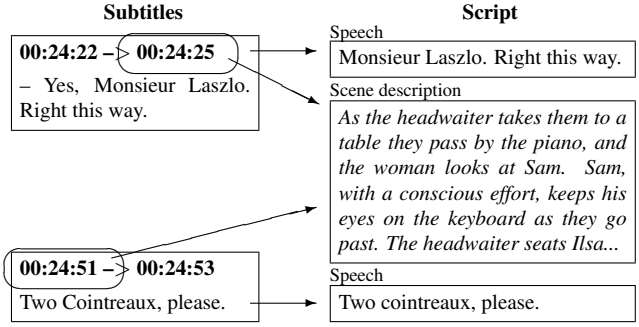
2. Automatic supervision for action recognition

Manual annotation of many classes and many samples of human actions in video is a time-consuming process. For example, common actions such as hand shaking or kissing occur only a few times per movie on average, thus collecting a fair-sized number of action samples for training requires annotation of tens or hundreds of hours of video. In this situation, the automatic annotation of training data is an interesting and scalable alternative which will be addressed in this paper.

Script-based retrieval of action samples. In this work, we follow [13] and automatically collect training samples with human actions using video scripts.[1] Whereas [13] uses supervised text classification for localizing human actions in scripts, here we avoid manual text annotation altogether. We use the OpenNLP toolbox [18] for natural language processing and apply part of speech (POS) tagging to identify instances of nouns, verbs and particles. We also use named entity recognition (NER) to identify people's names. Given results of POS and NER we search for patterns corresponding to particular classes of human actions such as (*/PERSON .* opens/VERB .* door/NOUN). This procedure automatically locates instances of human actions in text:

... **Jane** jumps up and **opens** the **door** ...
... **Carolyn opens** the front **door** ...
... **Jane opens** her bedroom **door** ...

Temporal localization of actions in video. Scripts describe events and their order in video but usually do not provide time information. Following [5, 8, 13] we find temporal localization of dialogues in scripts by matching script text with the corresponding subtitles using dynamic programming. The temporal localization of human actions and scene descriptions is then estimated from the time intervals of surrounding speech as illustrated below:



Automatic script alignment only provides coarse temporal localization of human actions, especially for episodes with rare dialogues. In addition, incorrect ordering of actions and speech in scripts and the errors of script/subtitle alignment often result in unknown temporal offsets. To overcome temporal misalignment, we increase the estimated time boundaries of scene descriptions. In this work we denote parts of the video corresponding to scene descriptions as *video clips*. Table 1 illustrates manually evaluated accuracy of automatic script-based annotation in video clips with increasing temporal extents.

Training an accurate action classifier requires video clips with both accurate labels and precise temporal boundaries. The high labelling accuracy in Table 1, however, is bound to the imprecise temporal localization of action samples. This trade-off between accuracies in labels and temporal localization of action samples comes from the aforementioned

[1] Note that we use scripts as a means of weak supervision at the training stage only. Scripts are not available at the test stage of visual action annotation. We use movie scripts publicly available from www.dailyscript.com, www.movie-page.com and www.weeklyscript.com

Figure 2. Space-time interst points detected for multiple video resolutions and three frames of a StandUp action. The circles indicate spatial locations and scales of space-time patches used to construct bag-of-features video representaion (see text for more details).

Clip length (frames)	100	200	400	800	1600
Label accuracy	19%	43%	60%	75%	83%
Localization accuracy	74%	37%	18%	9%	5%

Table 1. Accuracy of automatic script-based action annotation in video clips. Label accuracy indicates the proportion of clips containing labeled actions. Localization accuracy indicates the proportion of frames corresponding to labeled actions. The evaluation is based on the annotation of three actions classes: StandUp, SitDown and OpenDoor in fifteen movies selected based on their availability.

problems with imprecise script alignment. In this paper we target this problem and address visual learning of human actions in a *weakly supervised setting* given imprecise temporal localization of training samples. Next, we present a weakly supervised clustering algorithm to automatically localize actions in training samples.

3. Human action clustering in video

To train accurate action models we aim at localizing human actions inside video clips provided by automatic video annotation. We assume most of the clips contain at least one instance of a target action and exploit this redundancy by clustering clip segments with consistent motion and shape. In Section 3.1 we describe our video representation. Sections 3.2 and 3.3, respectively, formalize the clustering problem and describe the discriminative clustering procedure. Finally, Section 3.4 details the baseline k-means like algorithm and Section 3.5 experimentally evaluates and compares the two clustering algorithms.

3.1. Video representation

We use a bag-of-features representation motivated by its recent success for object, scene and action classification [6, 13, 17, 19]. We detect local space-time features using an extended Harris operator [12, 13] applied at multiple spatial and temporal video resolutions.[2] The resulting patches correspond to local events with characteristic motion and shape in video as illustrated in Figure 2. For each detected patch, we compute the corresponding local motion and shape descriptors represented by histograms of spatial gradient orientations and optical flow respectively. Both descriptors are concatenated into a single feature vector and quantised using k-means vector quantisation and a visual vocabulary of size N=1000. We represent a video segment by its ℓ_1-normalized histogram of visual words.

[2]We use publicly available implementation of feature detector and descriptor from `http://www.irisa.fr/vista/actions`

3.2. Joint clustering of video clips

Our goal is to jointly segment video clips containing a particular action—that is, we aim at separating what is common within the video clips (i.e., the particular action) from what is different among these (i.e, the background frames). Our setting is however simpler than general co-segmentation in the image domain since we only perform *temporal* segmentation. That is, we look for segments that are composed of contiguous frames.

For simplicity, we further reduce the problem to separating one segment per video clip (the action segment) from a set of *background video segments*, taken from the same movie or other movies, and which are unlikely to contain the specific action. We thus have the following learning problem: We are given M video clips $c_1, \ldots, c_M$ containing the action of interest but at unknown position within the clip as illustrated in Figure 3. Each clip c_i is represented by n_i temporally overlapping segments centered at frames $1, \ldots, n_i$ represented by histograms $h_i[1], \ldots, h_i[n_i]$ in $\mathbb{R}^N$. Each histogram captures the ℓ_1-normalized frequency counts of quantized space-time interest points, as described in section 3.1, i.e. it is a positive vector in $\mathbb{R}^N$ whose components sum to 1. We are also given P background video segments represented by histograms $h^b_1, \ldots, h^b_P \in \mathbb{R}^N$. Our goal is to find in each of the M clips i one specific video segment centered at frame $f_i \in \{1, \ldots, n_i\}$ so that the set of M histograms $h_i[f_i]$, $i = 1, \ldots, M$ form one cluster while the P background histograms form another cluster as illustrated in figure 4.

3.3. Discriminative clustering

In this section, we formulate the above clustering problem as a minimization of a discriminative cost function [23]. First, let us assume that correct segment locations f_i, $i \in$

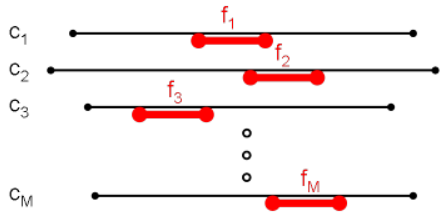


Figure 3. Illustration of the temporal action clustering problem. Given a set of M video clips $c_1, \ldots, c_M$ containing the action of interest at unknown position, the goal is to temporally localize a video segment in each clip containing the action.

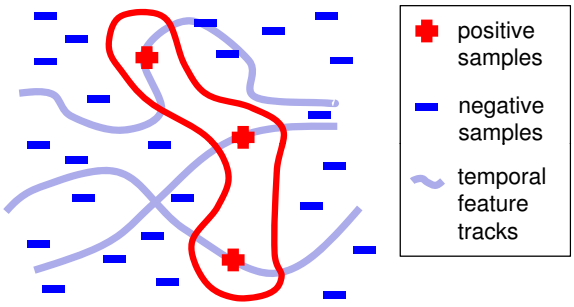


Figure 4. In feature space, positive samples are constrained to be located on temporal feature tracks corresponding to consequent temporal windows in video clips. Background (non-action) samples provide further constrains on the clustering.

$\{1, \ldots, M\}$, are known (i.e., we have identified the locations of the actions in video). We can now consider a support vector machine (SVM) [20] classifier aiming at separating the identified action video segments from the given background video segments, which leads to the following cost function

$$J(f, w, b) = C_+ \sum_{i=1}^{M} \max\{0, 1 - w^\top \Phi(h_i[f_i]) - b\} + C_- \sum_{i=1}^{P} \max\{0, 1 + w^\top \Phi(h_i^b) + b\} + \|w\|^2, \quad (1)$$

where $w \in \mathcal{F}$ and $b \in \mathbb{R}$ are parameters of the classifier and Φ is the implicit feature map from $\mathbb{R}^N$ to feature space $\mathcal{F}$, corresponding to the intersection kernel between histograms, defined as [10]

$$k(x, x') = \sum_{j=1}^{N} \min(x_j, x'_j). \quad (2)$$

Note that the first two terms in cost function (1) represent the hinge loss on positive and negative training data weighted by factors C_+ and C_- respectively, and the last term is the regularizer of the classifier. Note that training the SVM with locations f_i known and fixed corresponds to minimizing $J(f, w, b)$ with respect to classifier parameters w, b.

However, in the clustering setup considered in this work, where the locations f_i of action video segments within clips are unknown, the goal is to minimize the cost function (1) both with respect to the locations f_i and the classifier parameters w, b, so as to separate positive action segments from (fixed) negative background video segments. Denoting by $H(f) = \min_{w \in \mathcal{F}, b \in \mathbb{R}} J(f, w, b)$ the associated optimal values of $J(f, w, b)$, the cost function $H(f)$ now characterizes the separability of a particular selection of action video segments f from the (fixed) background videos. Following [11, 23], we can now optimize $H(f)$ with respect to the assignment f.

We consider a coordinate descent algorithm, where we iteratively optimize $H(f)$ with respect to position f_i of the action segment in each clip, while leaving all other components (positions of other positive video segments) fixed. In our implementation, which uses the LibSVM [3] software, in order to save computing time, we re-train the SVM (updating w and b) only once after an optimal f_i is found in each clip.

Note that the position f_i of an action segment within clip c_i can be parametrized using a binary indicator vector z_i, with 1 at position f_i and zero otherwise. This representation naturally leads to a continuous relaxation of the clustering problem by allowing z_i to have any (i.e. non-binary) positive values, which sum to one. We use the idea of continuous relaxation for initialization of the coordinate descent algorithm. Initial histogram h_i^0 for each video clip c_i is set to the average of all segment histograms $h_i[f_i]$ within the clip. Using the relaxed indicator notation, this corresponds to initializing z_i with a small fixed value for all segments, equal to one over the number of segments in the clip.

3.4. Clustering baseline: modified k-means

To illustrate the difficulty of the clustering task addressed in this paper, we consider a baseline method in terms of a modified k-means algorithm. This type of algorithm has been used previously for weakly-supervised spatial object category localization in image sets [4].

We consider the joint distortion measure of assigning some of the candidate segments to a mean $\mu \in \mathcal{F}$, while assigning all other segments (and the background) to a mean $\nu \in \mathcal{F}$. Using the indicator vector notation, we minimize the following function with respect to z and μ, ν:

$$\sum_{i=1}^{M} \sum_{j=1}^{n_i} \left[z_{ij} \|\Phi(h_i[j]) - \mu\|^2 + (1 - z_{ij}) \|\Phi(h_i[j]) - \nu\|^2 \right] + \sum_{i=1}^{P} \|\Phi(h_i^b) - \nu\|^2. \quad (3)$$

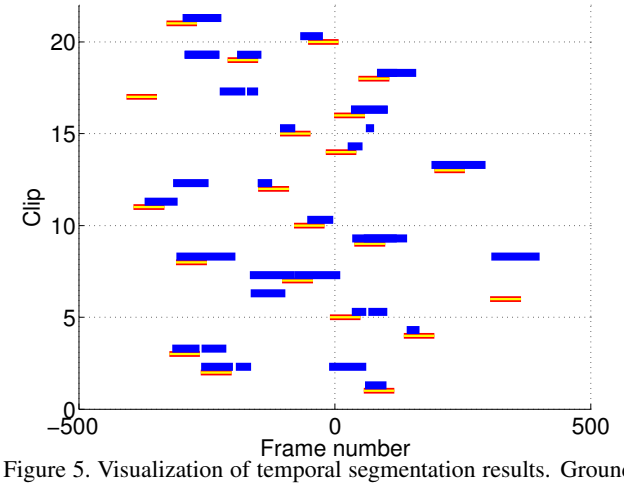


Figure 5. Visualization of temporal segmentation results. Ground truth action locations are shown by thick blue bars and locations automatically obtained by temporal clustering are shown as red-yellow bars. The action is approximately localized in 19 out 21 cases. See text for details.

This cost function can be (locally) minimized alternatively with respect to all z_i, $i \in \{1, \ldots, M\}$ and to μ and ν. In our experiments, as shown in Section 3.5, it does not lead to good solutions. Our interpretation is that this type of algorithm, like regular k-means, relies heavily on the metric $\|\Phi(x) - \Phi(x')\|$ without the possibility of adapting it to the data, which discriminative clustering can do.

3.5. Evaluation of clustering performance

In this section we apply the discriminative clustering algorithm described above to temporal segmentation of "drinking" actions in the movie "Coffee and Cigarettes". The quality of the segmentation is evaluated in terms of localization accuracy. Section 4 then evaluates the benefit of automatic action segmentation for the supervised temporal action detection.

The clustering algorithm is evaluated on a set of 21 drinking actions. The remaining 38 actions are left out for testing detection performance. Our test and training videos do not share the same scenes or actors. For both the training and test set the ground truth action boundaries were obtained manually. To evaluate the algorithm in controlled settings, we simulate script-based weak supervision by extending the ground truth action segments by random amounts of between 0 and 800 frames on each side. The negative data is obtained by randomly sampling segments of a fixed size from the entire movie. The size of positive segments is kept fixed at 60 frames.

Our clustering algorithm converges in a few (3-5) iterations both in terms of the cost given in (1) and the localization accuracy (discussed below). Automatically localized segments are shown in Figure 5 and example frames from several clips are shown in Figure 6. Note the significant variation of appearance between the different actions.

The temporal localization accuracy is measured by the percentage of clips with relative temporal overlap to ground truth action segments greater than 0.2. The best overlap score of 1 is achieved when the automatically found action video segment aligns perfectly with the ground truth action, and 0 in the case of no temporal overlap. This relatively loose threshold of 0.2 is used in order to compensate for the fact that temporal boundaries of actions are somewhat ambiguous and not always accurately defined. Using this performance measure discriminative clustering correctly localizes 19 out of 21 clips, which corresponds to an accuracy of 90%. There are two missed actions (6 and 18 in figure 5). Clip 6 contains a significant simultaneous motion of another person not performing the target action. In clip 18 drinking is mismatched for smoking (the two actions are visually quite similar).

The 90% accuracy achieved by discriminative clustering is a significant improvement over the k-means algorithm described in section 3.4, which fails completely on this data and achieves accuracy of only 5%. We have also implemented a variation of the k-means method minimizing the sum of distances between all positive examples [4] instead of sum of distances to the mean, but obtained similarly low performance. This could be attributed to the fact that discriminative clustering selects relevant features within the histograms. This is important in our setting where histograms can be polluted by background motion or other actions happening within the same frame.

4. Temporal action detection in video

In this section we experimentally evaluate, in a controlled setting, whether the action classifier trained on automatically clustered action segments can be used to improve the performance of temporal action detection in new unseen test videos with no textual annotation.

Temporal sliding window detection Similar to object detection in image domain, we train a SVM classifier to classify a short video segment as to whether it contains the action of interest, and apply the classifier in a sliding window manner over the entire video. The classifier output is then processed using a standard non-maximum suppression algorithm.

Evaluation We use the same test data as in [14] formed by 35,973 frames of the movie Coffee and Cigarettes containing 38 drinking actions. In all cases we consider sliding windows with temporal scales of 60, 80 and 100 frames, and the negative training data is formed by 5,000 video segments randomly sampled from the training portion of the movie. Similar to object detection, performance is measured using precision-recall curve and average precision (AP).

Figure 6. Examples of temporally localized "drinking" actions in the movie "Coffee and Cigarettes" by the proposed discriminative clustering algorithm. Each row shows example frames from the entire video clip. Example frames of automatically localized actions within the clips are shown in red. Note the appearance variation between the different instances of the action.

We investigate, how the detection performance changes with the increasing size of the training video segments containing the positive training examples. The goal of this experiment is to simulate inaccurate temporal localization of actions in clips obtained from text annotation. Figure 7 (top) shows precision-recall curves for training clip sizes varying between 800 frames and the precise ground truth (GT) boundaries of the positive actions. The decreasing performance with increasing training clip size clearly illustrates the importance of temporal action localization in the training data. The performance is also compared to the approach of Laptev and Pérez [14] which in addition spatially localizes actions in video frames. For comparison, however, we here only consider temporal localization performance of [14]. Measured by average precision, the spatio-temporal sliding window classifier of Laptev and Perez performs slightly better (AP of 0.49) compared to the temporal sliding window classifier considered in this work (AP of 0.40). It should be noted, however, that [14] requires much stronger supervision in the form of spatio-temporal localization of training actions in the video. Finally, Figure 7 (bottom) shows the precision-recall curve for training from localized action segments obtained automatically using our discriminative clustering method (Section 3) compared with training on entire video clips. Note the clear improvement in detection performance when using temporally localized training samples obtained with the clustering algorithm.

5. Experiments

In this section we test our full framework for automatic learning of action detectors including (i) automatic retrieval of training action clips by means of script mining (Section 2), (ii) temporal localization of actions inside clips using discriminative clustering (Section 3) and (iii) Supervised temporal detection of actions in test videos (Section 4). To train an action classifier we use fifteen movies[3] aligned with the scripts and choose two test action classes OpenDoor and SitDown based on their high frequency in our data. Our only manual supervision provided to the

[3]Our fifteen training movies were selected based on their availability as well as the quality of script alignment. The titles of the movies are: American Beauty; Being John Malkovich; Casablanca; Forrest Gump; Get Shorty; Its a Wonderful Life; Jackie Brown; Jay and Silent Bob Strike Back; Light Sleeper; Men in Black; Mumford; Ninotchka; The Hustler; The Naked City and The Night of the Hunter.

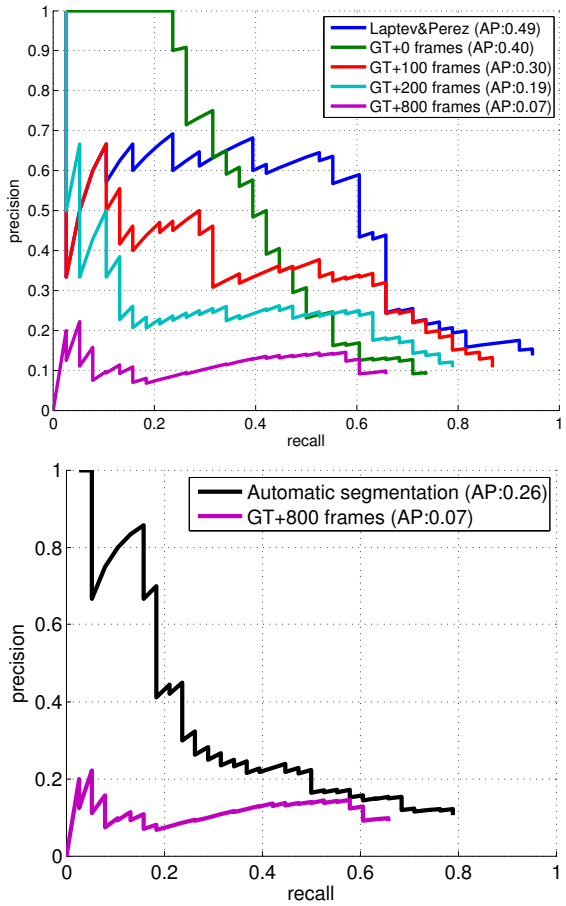


Figure 7. Action detection performance on "Coffee and Cigarettes" test set for (top) increasing training clip sizes and (bottom) automatically segmented training data compared with training from unsegmented clips.

system consists of text patterns for the actions defined as (*/PERSON .* opens/VERB .* door/NOUN) and (*/PERSON .* sits/VERB .* down/PARTICLE). Matching these text patterns with scripts results in 31 and 44 clips with OpenDoor and SitDown actions respectively. We use these clips as input to the discriminative clustering algorithm and obtain segments with temporally localized action boundaries. The segments are passed as positive training samples to train an SVM action classifier. To compare performance of the method we also train two action classifiers using positive training samples corresponding to (a) entire clips and (b) ground truth action intervals.

To test detection performance we manually annotated all 93 OpenDoor and 86 SitDown actions in three movies: Living in oblivion, The crying game and The graduate. Detection results for the three different methods and two action classes are illustrated in terms of precision-recall curves in Figure 9. The comparison of detectors trained on clips and on action segments provided by the clustering clearly indicates the improvement achieved by the discriminative clustering algorithm for both actions. Moreover, the performance of automatically trained action detectors is comparable to the detectors trained on the ground truth data. We emphasize the large amount (450.000 frames in total) and high complexity of our test data illustrated with a few detected action samples in Figure 9.

6. Conclusions

We described a method for training temporal action detectors using minimal manual supervision. In particular, we addressed the problem of weak action supervision and proposed a discriminative clustering method that overcomes localization errors of actions in script-based video annotation. We presented results of action detection in challenging video data and demonstrated a clear improvement of our clustering scheme compared to action detectors trained from imprecisely localized action samples. Our approach is generic and can be applied to training of a large variety and number of action classes in an unsupervised manner.

Acknowledgments. This work was partly suppported by the Quaero Programme, funded by OSEO, and by the MSR-INRIA laboratory.

References

[1] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. In *ICCV*, pages II: 1395–1402, 2005.

[2] P. Buehler, M. Everingham, and A. Zisserman. Learning sign language by watching tv (using weakly aligned subtitles). In *CVPR*, 2009.

[3] C. C. Chang and C. J. Lin. *LIBSVM: a library for support vector machines*, 2001. http://www.csie.ntu.edu.tw/~cjlin/libsvm.

[4] O. Chum and A. Zisserman. An exemplar model for learning object classes. In *CVPR*, 2007.

[5] T. Cour, C. Jordan, E. Miltsakaki, and B. Taskar. Movie/script: Alignment and parsing of video and text transcription. In *ECCV*, pages IV: 158–171, 2008.

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

[7] A. Efros, A. Berg, G. Mori, and J. Malik. Recognizing action at a distance. In *ICCV*, pages 726–733, 2003.

[8] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... Buffy – automatic naming of characters in TV video. In *BMVC*, 2006.

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008.

[10] M. Hein and O. Bousquet. Hilbertian metrics and positive definite kernels on probability measures. In *Proc. AISTATS*, 2005.

[11] A. Howard and T. Jebara. Learning monotonic transformations for classification. In *NIPS*, 2007.

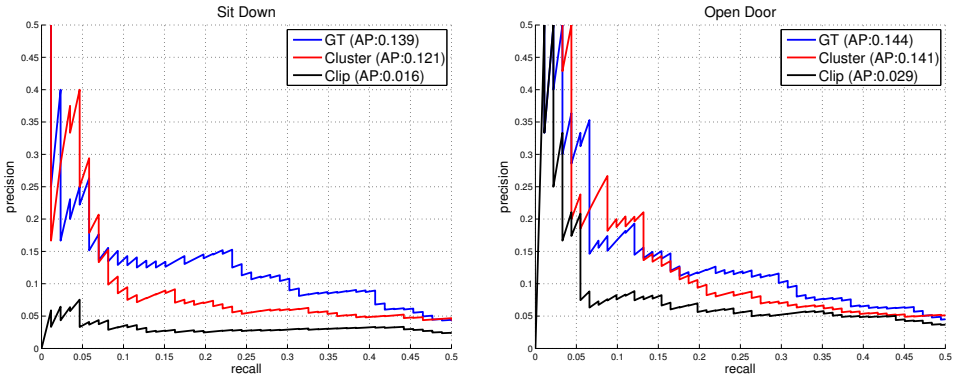


Figure 8. Precision-recall curves corresponding to detection results for two action classes in three movies. The three compared methods correspond to detectors trained on ground truth intervals (GT), clustering output (Cluster) and clips obtained from script mining (Clip). Note that the axes are scaled between 0 and 0.5 for better clarity.

Figure 9. Examples of action samples detected with the automatically trained action detector in three test movies.

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

[13] I. Laptev, M. Marszałek, C. Schmid, and B. Rozenfeld. Learning realistic human actions from movies. In *CVPR*, 2008.

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

[15] S. Lazebnik, C. Schmid, and J. Ponce. Semi-local affine parts for object recognition. In *BMVC*, volume 2, pages 959–968, 2004.

[16] A. Ng, M. Jordan, and Y. Weiss. On spectral clustering: Analysis and an algorithm. In *NIPS*, 2001.

[17] J. Niebles, H. Wang, and L. Fei-Fei. Unsupervised learning of human action categories using spatial-temporal words. In *BMVC*, 2006.

[18] OpenNLP. http://opennlp.sourceforge.net.

[19] C. Schuldt, I. Laptev, and B. Caputo. Recognizing human actions: A local SVM approach. In *ICPR*, 2004.

[20] J. Shawe-Taylor and N. Cristianini. *Kernel Methods for Pattern Analysis*. Camb. U. P., 2004.

[21] E. Shechtman and M. Irani. Space-time behavior based correlation. In *CVPR*, pages I:405–412, 2005.

[22] Trecvid evaluation for surveillance event detection, National Institute of Standards and Technology (NIST), 2008. http://www-nlpir.nist.gov/projects/trecvid.

[23] L. Xu, J. Neufeld, B. Larson, and D. Schuurmans. Maximum margin clustering. In *Adv. NIPS*, 2004.

[24] L. Zelnik-Manor and M. Irani. Event-based analysis of video. In *CVPR*, pages II:123–130, 2001.