FAST CASCADED ACTION LOCALIZATION IN VIDEO USING FRAME ALIGNMENT

Andrei Stoian⋆ *Marin Ferecatu*⋆ *Jenny Benois-Pineau*† *Michel Crucianu*⋆

⋆CEDRIC-CNAM, †LABRI-University of Bordeaux 1

ABSTRACT

Locating human actions in videos is challenging because of the complexity and variability of human motions, as well as of the amount of video data to be searched. We propose a method that detects and locates a set of actions in a video database by taking into account their temporal structure at the frame level. While other methods aggregate frames into action parts, we leverage the complementarity between aggregation and frame level comparison of sequences. Combining these two techniques in a cascade, we aim to address large scale retrieval. Evaluation on popular datasets show state of the art results, as well as efficient detection and low storage requirements.

***Index Terms*—** Action Localization, Tracklets, Cascade, Global Alignment, Time Warp

1. INTRODUCTION

Action detection is an important problem in computer vision, with many applications in video surveillance, in audiovisual content indexing and retrieval, and in human-computer interfaces. So-called *intermediate* level actions are composed of a series of atomic parts and can vary in complexity, *e.g.*from 'smoking' to 'pole vaulting'. Solutions vary from the most simple—averaging all descriptions in a temporal window—to weighted temporal segmentation [1] and tree-like structures [2]. Complex background, variability in point of view, occlusions and low video quality also pose a challenge for action detection in video.

We aim to perform action-based indexing of large-scale cultural video databases in order to support broad access to such content. For this, our primary focus is on the scalable detection of *intermediate* level actions. The method put forward in this work brings in three novel contributions: (1) to take advantage of the temporal information, we represent actions as time series and compare them using the Global Alignment (GA) kernel [3], (2) to find a better balance between efficiency and effectiveness, we propose a cascaded approach that employs both aggregated and frame level information, and (3) to improve time series comparisons with the GA kernel, we introduce a novel feature selection method for sparse multivariate time series.

Since a large enough ground truth based on a cultural video collection was not yet available for evaluation, we turned to existing benchmarks featuring a similar type of actions and content. In this work we evaluate our method on two datasets: Smoking and Drinking [4] and MSR Action II [5]. For the latter, our evaluation is performed with a cross-dataset approach: the model is trained on KTH [6] and tested on MSRII.

In Section 2 we discuss related work and briefly introduce Dynamic Time Warping and the Global Alignment kernel. In Section 3 we describe our approach and in Section 4 we present its experimental validation while Section 5 presents our conclusions.

2. RELATED WORK

2.1. Action Description and Localisation

Actions are modelled using either global descriptions of spatio-temporal volumes of the video or sets of local features describing spatio-temporal patches. With local features, modelling relies on their statistical distribution over a volume of the video.

Local features describing the dynamics of video patches such as trajectory-based features ([7]) are designed to describe both the trajectories and the spatial neighbourhoods of salient points. These descriptors are based on local shape (histogram of gradients, HoG), optical flow (histogram of optical flow, HoF) and optical flow gradient (motion boundary histogram, MBH [8]). The distribution of the local features for each frame is usually represented by a Bag of Visual Words (BoVW) histogram.

Gaidon *et al.*[1] use a sequential model of the action volume in which a soft ordering between 'meaningful temporal parts' is imposed. Klaser *et al.*[9] put forward a two stage approach for localizing human actions. First, a person detector (fast linear SVM) allows to filter out uninteresting windows. Second, an action detector is learned using HoG-Track descriptions and applied to improve the detection performance of the first stage. Oneata *et al.*[10] achieve state of the art results for temporal localisation by using Fisher Vectors to describe the distribution of trajectory features per frame.

This work is supported by the French National Research Agency grant ANR-11-IS02-001.

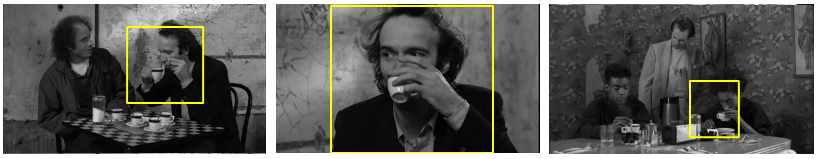

Fig. 1. Sample from the “Drinking and Smoking” dataset. Variability (left to right) of viewpoints (side, front, oblique view), of action ‘size’ as proportion of frame (medium, high, low) and of action length (50, 48 and 77 frames).

In [11] the authors use Branch and Bound (B&B) search to locate actions, iterating through all interesting detection volumes one by one. Scoring is based on the mutual information between the bag of STIP features in the sub-volume and the training set features, thus requiring the computation of nearest neighbours in feature space. [12] proposes a fast random forest scoring method that removes the need of the costly nearest neighbour search.

2.2. Temporal matching for action detection

The least expensive methods for comparing time series rely on temporal summarisation, either by averaging or by extracting compact descriptors of dynamic behaviour. However, temporal averaging loses potentially important information regarding both relative durations and temporal ordering.

Dynamic Time Warping (DTW, [13]) is a method for matching two time series that takes temporal ordering into account but is tolerant to temporal deformations. Let us consider two time series, Q and X, each being an ordered list $Q = \mathbf{q}_1, \mathbf{q}_2, ...\mathbf{q}_M$, $X = \mathbf{x}_1, \mathbf{x}_2, ...\mathbf{x}_N$ of n-dimensional vectors $\mathbf{q}_i$ and $\mathbf{x}_j$. DTW finds the best warping path $W^* \in \mathcal{A}$ (the set of all possible warping paths), $W^* = w_1, w_2, ..., w_K$ where $w_k = (i, j)$ between Q and X such that the total cost of the match is minimized, with respect to a cost function c: $DTW(Q, X) = \min_W \sum_{k=1}^{|W|} c(w_k)$. While DTW cannot be directly employed to build a positive definite kernel for time series classification with SVM, Cuturi *et al.*[3] propose to remove the $\min$ calculation and replace it with a summation over all the paths. They show that the exponentiation of the *soft-minimum* over the summation gives a positive definite kernel, the Global Alignment (GA) kernel (eq. 1), when the ground cost function c is the L_2 distance.

$$k_{GA}(Q, X) = \sum_{W \in \mathcal{A}} \exp \left\{ - \sum_{k=1}^{|W|} c(w_k) \right\} \qquad (1)$$

3. CASCADED DETECTION APPROACH

We consider that order at the frame level is important for the detection and localisation of the intermediate level actions we focus on. Comparisons between the time series describing videos should be robust to variations in speed while being discriminant with respect to the order of movements. This naturally leads us to use the GA kernel, which, however, is computationally expensive. Since we are interested in both the quality and the scalability of action detection, we propose a two level cascade where the inexpensive first level serves to filter out a maximum of irrelevant video segments. The more expensive second level, using the GA kernel, only processes what the first level considers as potentially relevant. To further improve the quality and speed of detection with the GA kernel, we also introduce a feature selection method adapted to sparse multidimensional time series.

For both stages of the cascade we employ sliding windows. Detectors are applied to fixed-size windows of $L = 30$ frames sampled every $s = 5$ frames. The cascade classifies the windows as Positive (containing the action) or Negative. In a final post-processing stage, overlapping positive detections are merged and their final score is computed. We provide below a more detailed description of the approach. Note that in [4] and [14] multiple window lengths of up to 120 frames are employed, which increases computation time. We found that with our method the use of multiple window sizes did not improve results.

3.1. Video time series description

To describe video content for action detection and localisation we follow [15]: points are sampled on a regular grid in each frame and tracked across 15 frames. Tracking is done by motion estimation between consecutive frames, based on the optical flow. The trajectory of a point consists of the coordinates of the point in consecutive frames. A “tracklet” description is built by concatenating three features, HoG, HoF and MBH, computed in patches around the trajectory points, which results in 396-dimensional descriptors. We quantize these descriptors using K-means into a visual dictionary $\mathcal{W}$ of 4,000 words and compute a L_1 normalized BoVW histogram for each frame. These sparse vectors, one for each video frame, constitute a high-dimensional time series description of the video. Below, *sequence* refers to any segment, or subseries, of a tracklet BoVW time series, while *window* refers to a sequence of $L = 30$ frames.

3.2. Model learning

We employ SVM-based detectors at both levels of the cascade. SVM parameters are optimized through adaptive grid search, using either a dedicated validation set or cross-validation, with Average Precision as the performance criterion. Detectors at both stages of the cascade are trained following a One-vs-All scheme. For each action class, an equal number of positive and negative windows are extracted from the training set. Negative examples are sampled from background windows and from the windows corresponding to the other classes.

3.3. Cascade construction and optimization

First stage. The first stage of the cascade has to filter out a maximum of windows that are not likely to contain the action of interest. It is directly applied on all the windows extracted from the videos in which we aim to locate actions. For the first stage, a window is described by the renormalised sum of the BOVW histograms of all its frames (denoted below $X^{(agg)}$). The first stage has to decide whether a window is relevant (to be passed to the second level) or should be ignored. The decision is taken according to the value of the SVM decision function: relevant iff $f_1(X^{(agg)}) > \tau_1$. For the first stage, the SVM classifiers employ the Histogram Intersection (HI) kernel: $k_{HI}(Q, X) = \sum_{i=1}^{|\mathcal{W}|} min(Q_i^{(agg)}, X_i^{(agg)})$. The decision threshold τ_1 is adjusted so as to minimise the number of false negatives.

Second stage. The second stage classifies windows as Positive or Negative. It is only applied to the windows that are considered relevant by the first stage of the cascade. For the second stage, a window is described as the time series of BOVW histograms of all its frames (denoted below $X^{(ts)}$). The second stage employs SVM with the GA kernel (eq. 1). A sequence $X^{(ts)}$ is Positive iff $f_2(X^{(ts)}) > \tau_2$, where f_2 is the decision function of the second stage SVM. We employed $\tau_2 = 0$. The final score of positive windows is f_2.

Cascade optimization. The first stage should filter out a maximum of irrelevant windows in order to reduce overall detection cost, while keeping recall as high as possible. In the following we call *coverage* the ratio of windows that are found relevant by the first stage and, thus, sent to the second stage. Higher values of the decision threshold τ_1 reduce the coverage (thus diminishing computation time). But the number of false negatives also increases with τ_1. In this work, recall maximisation was the only criterion employed for selecting τ_1. When the scalability requirements are important an optimal trade-off between maximal recall and minimal value of coverage has to be found. Note that, for each class, τ_1 is optimized on a drawn out validation set.

Post-processing. A sliding window approach for detection can lead to multiple overlapping positive windows. To obtain the final detection sequences all positive windows overlapping by more than $\tau_{merge} = 50\%$ are merged by using the union of their bounds. The resulting detection sequence is assigned the sum of scores of the windows.

3.4. Feature selection for frame alignment

Videos usually contain background motion in addition to the actions of interest, tracklet features are noisy and spurious trajectories are abundant. Many of the features (visual words) describing video frames will then act as noise. Furthermore, the GA kernel employs a summation over all the warping paths, using the L_2 distance between frames as an atomic dissimilarity measure (eq. 1). Consequently, the use of all the features may strongly impair the discrimination ability of the GA kernel. It is then important to use feature selection in the second stage prior to the application of the GA kernel. This also has a positive impact on the computation cost.

Feature selection methods following the filter approach, like mRMR [16], aim to maximise the mutual information between the selected features and the classes, while minimising feature redundancy. Such methods retain just as well features (visual words) that are present in the positive examples and absent from the negative ones as features that are present in the negative examples and absent from the positive ones. However, the negative examples for one class include not only examples from the other classes but also 'background' sequences, i.e. video sequences not showing any of the actions. Such background sequences from the training data are not representative for other videos. Consequently, features that are present in such sequences and absent from the positive examples will act as noise.

We introduce here a feature selection method that takes this issue into account and also considers the fact that we compare sequences of vectors rather than simple vectors. More formally, a feature set $\mathcal{F}$ is considered present in a sequence $x = x_1, \ldots, x_L$ if *all* the frames in the sequence contain at least one feature of the set. A feature set $\mathcal{F}$ is considered absent if at least one frame *does not* contain *any* feature from the set. We aim to find a set of features (visual words) that is (1) maximally present in the positive examples $\mathcal{P}$, and (2) maximally absent from the negative examples $\mathcal{N}$. Presence in the positive examples is measured by $P^+(\mathcal{F})$ in eq. 2 and absence from the negative examples by $A^-(\mathcal{F})$ in eq. 3. We use greedy search to jointly maximize these two criteria. We call this method Presence in Positives and Absence from Negatives (PPAN).

$$P^+(\mathcal{F}) = \frac{1}{|\mathcal{P}|} \sum_{x \in \mathcal{P}} \prod_{i=1}^{L} \mathbb{1}\left(\sum_{w \in \mathcal{F}} x_{iw} \right) \quad (2)$$

$$A^-(\mathcal{F}) = \frac{1}{|\mathcal{N}|} \sum_{x \in \mathcal{N}} \left[1 - \mathbb{1}\left(\prod_{i=1}^{L} \sum_{w \in \mathcal{F}} x_{iw} \right) \right] \quad (3)$$

In eq. 2 and 3, x_{iw} is the value of word w at time i in the BoVW sequence x and $\mathbb{1}(x) = \{1 \text{ if } x > 0, 0 \text{ otherwise}\}$. These criteria can be extended to other feature representations. The algorithm stops when the selected features are

	D	**S**	**C**	**W**	**B**
Windows with action	5%	7%	9%	12%	8%
Coverage	21%	70%	48%	35%	46%
Nb. 2nd stage features	150	150	50	30	30
Windows with features	93%	89%	83%	37%	72%
Windows for 2nd stage	20%	62%	40%	13%	33%

Table 1. Key figures for the cascade on the test sets of the Smoking and Drinking and MSR Action II datasets

present in 80% of the positive examples or an upper limit is reached for this presence. Preliminary experiments allowed to validate this criterion.

4. EXPERIMENTAL EVALUATION

Datasets. The "Smoking and Drinking" dataset [4] (3 hours) consists of three videos : 2 feature films, "Coffee and Cigarettes" (2002) and "Sea of Love" (1989), and one video consisting solely of drinking sequences. It is split into a training set, a validation set and a testing set. For the **D**rinking action there are 106 training, 16 validation and 38 test sequences. For the **S**moking action there are 78 training, 12 validation and 42 test sequences. The "MSR Action Dataset II" [5] has one hour of footage split into 54 videos with cluttered background. It contains three actions selected among those of the KTH dataset [6]: **B**oxing (81 instances), [hand]**C**lapping (51) and [hand]**W**aving (71). Training is done using sequences extracted from KTH (100 videos per action).

Evaluation metrics. Following the recent literature, action localisation is evaluated like a retrieval problem: we consider all detection windows with positive scores as results and we sort the windows by their score. This allows to obtain precision/recall curves and to compute the Average Precision (AP) in order to characterize the detection performance.

A detection X is considered to be a "true positive" if the Jaccard coefficient $\mathcal{J}(X, G) = |X \cap G|/|X \cup G|$ between it and a positive ground truth window G is greater than 20%.

We first discuss the impact of the first stage and of feature selection, giving figures in Table 1. The first row of Table 1 characterizes the datasets: we can say they are relatively 'dense', i.e. the proportion of ground truth windows containing the action of interest is high. The percentage of windows needing further examination after the first stage (coverage) is given in the second row of the table. We note that the filtering power of the first stage was highest on Drinking (20%) and is on average of 44%. On Smoking and Drinking the first stage decision threshold τ_1 was selected to maximise recall. On MSRII, since no validation split is provided, τ_1 was set to the average first stage decision value. On less 'dense' datasets and with a joint optimisation of coverage and recall, we expect coverage to be much lower. Feature selection strongly reduces the number of features, from 4,000 to 30-150. This significantly accelerates the second stage of the cascade since the computation of the GA kernel has a complexity of $O(L^2)$ (L is the window length). As a side effect of our feature selection method, in part of the windows sent to the second stage none of the selected features is present. These windows can thus be discarded. Row 'Windows with features' in Table 1 shows the ratio of windows still containing the selected features. Row 'Windows for 2nd stage' (product between 'Coverage' and 'Windows with features') shows how many windows are actually processed by the second stage of the cascade. In the experiments reported here, filtering and feature selection together make the cost of the second stage roughly equal to the cost of the first one.

Method	Drinking	Smoking
Gaidon *et al.*[1]	57%	31%
Klaser *et al.*[9]	59%	24%
Oneata *et al.*[10]	64%	**50%**
AP of our method	**65.5%**	45.1%
Search time (ours)	178 s	240 s

Table 2. Performance comparison on Smoking and Drinking

Method	Clapping	Waving	Boxing
Cao *et al.*[17]	13.1%	36.7%	17.5%
B&B Search [12]	23.9%	43.0%	30.3%
Max Subarray [18]	36.1%	54.1%	31.7%
AP of our method	**39.7%**	**55.0%**	**39.6%**
Search time (ours)	78s average		

Table 3. Performance comparison on MSR Action II

Table 2 and Table 3 show how our results compare to the state of the art on Drinking and Smoking and respectively on MSR Action II. We improve AP for all actions except Smoking. The better results obtained in [10] for Smoking can probably be explained by the better description quality of Fisher Vectors (FV) compared to sparse BoVW.

5. CONCLUSION

In this paper we presented an approach for localising intermediate level actions in videos. The method is designed for processing large volumes of data. We improve upon the state of the art in two ways: by using a GA kernel for sequence comparison, considering that frame order is important for action detection, and by devising a two-stage cascaded approach that allows to achieve fast and accurate retrieval of actions due to pre-filtering and feature selection. We have also shown that the cascade provides better results than both the use of the GA kernel alone or of temporal summarisation alone.

6. REFERENCES

[1] A. Gaidon, Z. Harchaoui, and C. Schmid, "Actom Sequence Models for Efficient Action Detection," *CVPR 2011 - IEEE Conference on Computer Vision & Pattern Recognition*, pp. 3201–3208, June 2011.

[2] A. Gaidon, Z. Harchaoui, and C. Schmid, "Recognizing activities with cluster-trees of tracklets," *British Machine Vision conf. 2012*, pp. 30.1–30.13, 2012.

[3] M. Cuturi, "Fast global alignment kernels," *Proceedings of the 28th International Conference on Machine Learning (ICML-11)*, pp. 929–936, June 2011.

[4] I. Laptev and P. Pérez, "Retrieving actions in movies," *Proc. Int. Conf. on Computer Vision (ICCV'07)*, pp. 1–8, Oct 2007.

[5] J. Yuan, Z. Liu, and Y. Wu, "Discriminative video pattern search for efficient action detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 9, pp. 1728–1743, 2011.

[6] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: a local svm approach," in *Proceedings of the 17th International Conference on Pattern Recognition ICPR 2004*. IEEE, 2004, vol. 3, pp. 32–36.

[7] M. Raptis and S. Soatto, "Tracklet descriptors for action modeling and video analysis," *ECCV'10 Proc. of the 11th European conf. on Computer vision*, pp. 577–590, 2010.

[8] N. Dalal, B. Triggs, and C. Schmid, "Human detection using oriented histograms of flow and appearance," *ECCV'06 Proc. of the 9th European conf. on Computer Vision - Volume Part II*, pp. 428–441, 2006.

[9] A. Klaser, M. Marszalek, C. Schmid, and A. Zisserman, "Human focused action localization in video," *Proc. of the 11th European Conf. on Trends and Topics in Computer Vision - Volume Part I*, pp. 219–233, 2012.

[10] D. Oneata, J. Verbeek, and C. Schmid, "Action and Event Recognition with Fisher Vectors on a Compact Feature Set," in *ICCV 2013 - IEEE Intenational Conference on Computer Vision*, Sydney, Australia, Dec. 2013, pp. 1817–1824, IEEE.

[11] J. Yuan, Z. Liu, Y. Wu, and Z. Zhang, "Speeding up spatio-temporal sliding-window search for efficient event detection in crowded videos," in *Proceedings of the 1st ACM International Workshop on Events in Multimedia*, New York, NY, USA, 2009, EiMM '09, pp. 3–8.

[12] G. Yu, J. Yuan, and Z. Liu, "Unsupervised random forest indexing for fast action search," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*, June 2011, pp. 865–872.

[13] H. Sakoe and S. Chiba, "Dynamic programming algorithm optimization for spoken word recognition," *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 26, no. 1, pp. 43–49, 1978.

[14] A. Gaidon, Z. Harchoui, and C. Schmid, "A time series kernel for action recognition," *Proc. of the British Machine Vision conf. 2011*, pp. 63.1–63.11, 2011.

[15] H. Wang, A. Kläser, C. Schmid, and C.L. Liu, "Dense trajectories and motion boundary descriptors for action recognition," *International Journal of Computer Vision*, vol. 103, no. 1, pp. 60–79, May 2013.

[16] H. Peng, F. Long, and C. Ding, "Feature selection based on mutual information criteria of max-dependency, max-relevance, and min-redundancy," *IEEE Trans. on PAMI*, vol. 27, no. 8, pp. 1226–1238, 2005.

[17] L. Cao, Z. Liu, and T.S. Huang, "Cross-dataset action detection," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, June 2010, pp. 1998–2005.

[18] G. Yu, J. Yuan, and Z. Liu, "Real-time human action search using random forest based hough voting," in *Proceedings of the 19th ACM International Conference on Multimedia*, New York, NY, USA, 2011, MM '11, pp. 1149–1152, ACM.