Category-specific video summarization

Danila Potapov, Matthijs Douze, Zaid Harchaoui, Cordelia Schmid

Category-specific video summarization

Danila Potapov, Matthijs Douze, Zaid Harchaoui, and Cordelia Schmid

Inria*

Abstract. In large video collections with clusters of typical categories, such as "birthday party" or "flash-mob", category-specific video summarization can produce higher quality video summaries than unsupervised approaches that are blind to the video category.
Given a video from a known category, our approach first efficiently performs a temporal segmentation into semantically-consistent segments, delimited not only by shot boundaries but also general change points. Then, equipped with an SVM classifier, our approach assigns importance scores to each segment. The resulting video assembles the sequence of segments with the highest scores. The obtained video summary is therefore both short and highly informative. Experimental results on videos from the multimedia event detection (MED) dataset of TRECVID'11 show that our approach produces video summaries with higher relevance than the state of the art.

Keywords: video summarization, temporal segmentation, video classification

1 Introduction

Most videos from YouTube or DailyMotion consist of long-running, poorly-filmed and unedited content. Users would like to browse, i.e., to *skim through* the video to quickly get a hint on the semantic content. Video summarization addresses this problem by providing a short video summary of a full-length video. An ideal video summary would include all the important video segments and remain short in length. The problem is extremely challenging in general and has been subject of recent research [1,2,3,4,5,6].

Large collections of videos contain clusters of videos belonging to specific categories with typical visual content and repeating patterns in the temporal structure. Consider a video of a "birthday party" (see Figure 1). It is unclear how an unsupervised approach for video summarization would single out the short segments corresponding to "blow the candles", "applause", etc.

In this paper, we propose a category-specific summarization approach. A first distinctive feature of our approach is the temporal segmentation algorithm. While most previous works relate segment boundaries to shot boundaries, our temporal segmentation algorithm detects general change points. This

* LEAR team, Inria Grenoble Rhône-Alpes, Laboratoire Jean Kuntzmann, CNRS, Univ. Grenoble Alpes, France.

Fig. 1: Original video, and its video summary for the category "birthday party".

includes shot boundaries, but also sub-shot boundaries where the transitions between sub-shots are gradual. A second feature is the category-specific supervised importance-scoring algorithm, which scores the *relative importance* of segments within each category, in contrast to video-specific importance [1,2,7].

Our approach works as follows (see Figure 2). First, we perform an automatic kernel-based temporal segmentation based on state-of-the-art video features that automatically selects the number of segments. Then, equipped with an SVM classifier for importance scoring that was trained on videos for the category at hand, we score each segment in terms of importance. Finally, the approach outputs a video summary composed of the segments with the highest predicted importance scores. Thus, our contributions are three-fold:

- we propose a novel approach, **KVS**, for supervised video summarization of realistic videos, that uses state-of-the-art image and video features
- we introduce a new dataset, **MED-Summaries**[1], along with a clear annotation protocol, to evaluate video summarization
- we obtain excellent experimental results on MED-Summaries, showing that KVS delivers video summaries with higher overall importance, as measured by two performance metrics.

2 Related work

Video summarization. Truong & Venkatesh [7] present a comprehensive overview and classification of video summarization methods. The task is difficult to define and many methods are domain-specific (sports, news, rushes, documentary, etc.).

[1] The annotations and the evaluation codes are available at http://lear.inrialpes.fr/people/potapov/med_summaries.php.

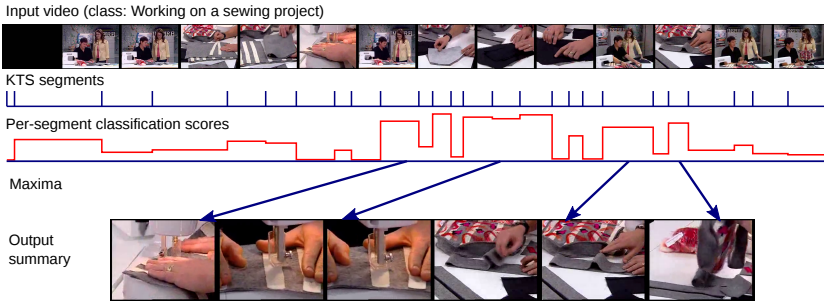


Fig. 2: Overall scheme of our Kernel Video Summarization (KVS) approach.

However, to our knowledge, there are no publicly available implementations or datasets, for eg. sports videos summarization, that could be used for comparison with more recent approaches. Summaries may focus on dominant concepts [8], relate to the video's story [6], the user's preferences, the query context [4], or user attention [9]. A video is either summed up as a sequence of keyframes [3,5,2] or by video excerpts [6].

Video summarization received much attention when NIST was running the Trecvid Rushes summarization task (2006-2008). The evaluation was conducted on a dataset of significant size, with an expensive manual annotation of the ground-truth [8]. However, the methods were mostly specific to the domain, i.e. they focused on detecting redundant shots of a scene, and clapperboards.

For professional and low-dynamic TV broadcast videos (e.g. from [8,4] or Open Video Archive), shot boundaries naturally split a video into "visual sentences". Early summarization methods [7] extract one or more keyframes to represent a shot, often independently from the other shots. Recent works, including this one, focus on user-generated data [3,5,6,10], which typically do not contain shot boundaries.

Without supervision, summarization methods must rely on low-level indices to determine the relevance of parts of a video [9,11,12]. When the video domain is known, summarization can be strongly supervised. For example, soccer games [13,14] or feature films [15] have standard phases that can be manually identified. A few previous works [3,6,5] produced summaries using features crafted for specific visual categories. In contrast to these works, our approach builds short yet highly informative category specific video summaries, using generic state-of-the-art visual features.

In [16,17], the main task is to remove redundant video footage, which is detected as easy to reconstruct based on sparse coding from the rest of the video. A recent work [10] also segments a video at a finer level than shots and relies on supervised mutual information to identify the important segments. The main difference of our work is the use of state-of-the-art video features and the quantitative evaluation of the approach. Leveraging crawled internet photos is another recent trend for video summarization [18,5].

There are several ways of evaluating video summarization methods [7]. Most works [3,6,5,11] conduct *user studies* to compare different summaries of the same video. The *concept coverage* metric evaluates the number of important objects or actions included in the summary [3,8]. Although it requires time-consuming manual annotation of videos, the annotations can be reused to evaluate multiple approaches. When the goal is to simplify video navigation, the time it takes a user to perform some data exploration task can be used as a quality metric [8]. *Automatic comparison to reference summaries* comes from text summarization literature [19]. It relies on a user-generated summary of a video and a metric to compare it to the algorithm's summary [5,2,18]. The protocol used in this paper combines concept coverage with a comparison to multiple reference summaries.

Temporal video segmentation. Computer vision methods often utilize spatial or temporal segmentation to raise the abstraction level of the problem and reduce its dimensionality. Segmentation can help to solve image classification, scene reconstruction [20] and can serve as a basis for semantic segmentation [21]. Similarly, video segmentation usually implies dividing a video into spatio-temporal volumes [22,23]. Temporal video segmentation often means detecting shot or scene boundaries, that are either introduced by the "director" through editing or simply correspond to filming stops.

The proliferation of user-generated videos created a new challenge for semantic temporal segmentation of videos. Lee et al. [3] used clustering of frame color histograms to segment temporal events. In [6] a video is split in sub-shots depending on the activity of the wearer of a head-mounted camera: "static", "moving the head" or "in transit". Similar to these works we focus on the content of the segment rather than its boundaries.

Most shot boundary detection methods focus on differences between consecutive frames [24], relying on image descriptors (pixel color histograms, local or global motion [7], or bag-of-features descriptors [25]). Our temporal segmentation approach takes into account the differences between *all pairs of frames*. Therefore, the approach allows to single out not only shot boundaries but also *change points* in general that correspond to non-abrupt boundaries between two consecutive segments with different semantic content.

3 Kernel video summarization

We start by giving definitions of the main concepts and building blocks of our approach.

Video summary. A video is partitioned into segments. A *segment* is a part of the video enclosed between two timestamps. A *video summary* is a video composed of a subset of the temporal segments of the original video.

A *summary* is a condensed synopsis of the whole video. It conveys the most *important* details of the original video. A segment can be non-informative due to signal-level reasons like abrupt camera shake and dark underexposed segments commonly present in egocentric videos [3,6].

A segment can be considered *important* due to multiple reasons, depending on the video category and application goals: highlights of sport matches, culmination points of movies [7], influential moments of egocentric videos [6].

We make the assumption that the notion of importance can be learned from a set of videos belonging to the video category. This point of view stems from the Multimedia Event Recounting task at Trecvid: selecting segments containing evidence that the video belongs to a certain event category. Similarly, we define importance as a *measure of relevance to the type of event*. Fig. 3 shows an example video together with the importance of its segments.

Our definition of importance spans an ordinal scale, ranging from 0 "no evidence" to 3 "the segment alone could classify the video into the category". More details are given in Sec. 4.1.

Fig. 3: Our definition of importance on the "Changing a vehicle tire" category. These frames come from a 1-minute video where a support car follows a cyclist during a cycle race. The main event — changing a bicycle tire — takes less than one third of the video. The figure shows central frames of user-annotated segments together with their importance score.

The proposed method, **KVS**, decomposes into three steps: i) kernel temporal segmentation; ii) importance-scoring of segments; iii) summary building. Figure 2 summarizes our approach.

3.1 Kernel temporal segmentation

Our Kernel Temporal Segmentation (KTS) method splits the video into a set of non-intersecting temporal segments. The method is fast and accurate when combined with high-dimensional descriptors.

Our temporal segmentation approach is a kernel-based change point detection algorithm. In contrast to shot boundary detection, change point detection is a more general statistical framework [29]. Change point detection usually focuses on piecewise constant one dimensional signals corrupted by noise, and the goal is to detect the jumps in the signal. It is able to statistically discriminate between jumps due to noise and jumps due to the underlying signal. Change-point detection has been subject of intense theoretical and methodological study

in statistics and signal processing; see [29,30] and references therein. Such methods enjoy strong theoretical guarantees, in contrast to shot boundary techniques that are mostly heuristic and tuned to the types of video transitions at hand (cut, fade in/out, etc.). We propose here a retrospective multiple change-point detection approach, based on [31], that considers the whole signal at once.

Given the matrix of frame-to-frame similarities defined through a positive-definite kernel, the algorithm outputs a set of optimal "change points" that correspond to the boundaries of temporal segments. More precisely, let the video be a sequence of descriptors $\mathbf{x}_i \in \mathbf{X}$, $i = 0, \ldots, n-1$. Let $K : \mathbf{X} \times \mathbf{X} \to \mathbb{R}$ be a kernel function between descriptors. Let $\mathcal{H}$ be the feature space of the kernel $K(\cdot, \cdot)$. Denote $\phi : \mathbf{X} \to \mathcal{H}$ the associated feature map, and $\|\cdot\|_{\mathcal{H}}$ the norm in the feature space $\mathcal{H}$. We minimize the following objective

$$\underset{m;\, t_0, \ldots, t_{m-1}}{\text{Minimize}} \quad J_{m,n} := L_{m,n} + Cg(m,n) \tag{1}$$

where m is the number of change points and $g(m,n)$ a penalty term (see below). $L_{m,n}$ is defined from the within-segment kernel variances $v_{t_i,t_{i+1}}$:

$$L_{m,n} = \sum_{i=0}^{m} v_{t_{i-1},t_i}, \quad v_{t_i,t_{i+1}} = \sum_{t=t_i}^{t_{i+1}-1} \|\phi(x_t) - \mu_i\|_{\mathcal{H}}^2, \quad \mu_i = \frac{\sum_{t=t_i}^{t_{i+1}-1} \phi(x_t)}{t_{i+1} - t_i} \tag{2}$$

Automatic calibration. The number of segments could be set proportional to the video duration, but this would be too loose. Therefore, the objective of Equation (1) decomposes into two terms: $L_{m,n}$ which measures the overall within-segment variance, and $g(m,n)$ that penalizes segmentations with too many segments. We consider a BIC-type penalty [32] with the parameterized form $g(m,n) = m(\log(n/m) + 1)$ [33]. Increasing the number of segments decreases $L_{m,n}$ (2), but increases the model complexity. This objective yields a trade-off between under- and over-segmentation. We propose to cross-validate the C parameter using a validation set of annotated videos. Hence we get kernel-based temporal segmentation algorithm where the number of segments is set automatically from data.

Algorithm. The proposed algorithm is described in Algo. 1. First, the kernel is computed for each pair of descriptors in the sequence. Then the segment variances are computed for each possible starting point t and segment duration d. It can be done efficiently by precomputing the cumulative sums of the matrix [34]. Then the dynamic programming algorithm is used to minimize the objective (2). It iteratively computes the best objective value for the first j descriptors and i change points. Finally, the optimal segmentation is reconstructed by backtracking. The total runtime cost of the algorithm is in $O(m_{\max} n^2)$. The penalization introduces a minimal computational overhead because the dynamic programming algorithm already computes $L_{i,n}$ for all possible segment counts.

3.2 Learning to predict importance scores

For each category, we train a linear SVM classifier from a set of videos with video-level labels, assuming that a classifier originally trained to classify the full

Algorithm 1 Kernel temporal segmentation

Input: temporal sequence of descriptors $\mathbf{x}_0, \mathbf{x}_1, \ldots, \mathbf{x}_{n-1}$	Cost
1. Compute the Gram matrix A : $a_{i,j} = K(\mathbf{x}_i, \mathbf{x}_j)$	$dn^2/2$
2. Compute cumulative sums of A	n^2
3. Compute unnormalized variances $v_{t,t+d} = \sum_{i=t}^{t+d-1} a_{i,i} - \frac{1}{d}\sum_{i,j=t}^{t+d-1} a_{i,j}$ $t = 0, \ldots, n-1, \quad d = 1, \ldots, n-t$	$2n^2$
4. Do the forward pass of the dynamic programming algorithm $L_{i,j} = \min_{t=i,\ldots,j-1}\big(L_{i-1,t} + v_{t,j}\big), \quad L_{0,j} = v_{0,j}$ $i = 1, \ldots, m_{\max}, \quad j = 1, \ldots, n$	$2m_{\max}n^2$
5. Select the optimal number of change points $m^\star = \arg\min_{m=0,\ldots,m_{\max}} L_{m,n} + Cg(m,n)$	$2m_{\max}$
6. Find change-point positions by backtracking $t_{m^\star} = n, \quad t_{i-1} = \arg\min_t\big(L_{i-1,t} + v_{t,t_i}\big)$ $i = m^\star, \ldots, 1$	$2m^\star$
Output: Change-point positions $t_0, \ldots, t_{m^\star-1}$	

videos can be used to score importance of small segments. This assumption is reasonable for videos where a significant proportion of segments have high scores. The opposite case, when a very small number of segments allow to classify the video ("needle in a haystack"), is outside the scope of the paper.

At training time, we aggregate frame descriptors of a video as if the whole video was a single segment. In this way a video descriptor has the same dimensionality as a segment descriptor. For each category we use videos of the category as positive examples and the videos from the other categories as negatives. We train one binary SVM classifier per category.

At test time, we segment the video using the KTS algorithm and aggregate Fisher descriptors for each segment. The relevant classifier is then applied to the segment descriptors, producing the *importance map* of the video.

In order to evaluate the summarization separately from the classification, we assume that the category of the video is known in advance. While recent methods specifically targeted at video classification [27,28] are rather mature, depending on them for our evaluation would introduce additional noise.

3.3 Summary building with KVS

Finally, a summary is constructed by concatenating the most important segments of the video. We assume that the duration of the summary is set a priori. Segments are included in the summary by the order of their importance until the duration limit is achieved (we crop the last segment to satisfy the constraint).

4 MED-summaries dataset

Most existing works evaluate summaries based on user studies, which are time-consuming, costly and hard to reproduce.

We introduce a new dataset, called **MED-summaries**. The proposed benchmark simplifies the evaluation by introducing a clear and automatic evaluation procedure, that is tailored to category-specific summarization. Every part of the video is annotated with a category-specific importance value. For example, for the category "birthday party", a segment that contains a scene where someone is blowing the candles is assigned a high importance, whereas a segment just showing children around a table is assigned a lower importance.

We use the training set of the Trecvid 2011 MED dataset (12,249 videos) to train the classifier for importance scoring. Furthermore, we annotate 60 videos from this training set as a validation set. To test our approach we annotate 100 videos from the official test set (10 per class), where most test videos have a duration from 1 to 5 minutes. Annotators mark the temporal segments and their importance; the annotation protocol is described in section 4.1. To take into account the variability due to different annotators, annotations were made by several people. In the experimental section we evaluate our results with respect to the different annotations and average the results. The different metrics for evaluation are described in section 4.2. See the dataset's website for details.

4.1 Annotation protocol

Segment annotation. The annotation interface shows one test video at a time, which can be advanced by steps of 5 frames. First, we ask a user to annotate temporal segments. Temporal segments should be *semantically consistent*, i.e. long enough for a user to grasp what is going on, but it must be possible to describe it in a short sentence. For example it can be "a group of people marching in the street" for a video of the class "Parade", or "putting one slice of bread onto another" for the class "Making a sandwich".

Some actions are repetitive or homogeneous, e.g. running, sewing, etc. In that case we ask to specify the "period" — minimum duration of a sub-segment that fully represents the whole segment. For example, watching 2-3 seconds of a running person is sufficient to describe the segment as "a person is running".

We require all shot boundaries to be annotated as change points, but change points do not necessarily correspond to shot boundaries. Often a shot contains a single action, but the main part is shorter than the whole segment. In this case we ask to localize precisely the main part.

Importance annotation. For each semantic segment we ask a user *"Does the segment contain evidence of the given event category?"*. The possible answers are:

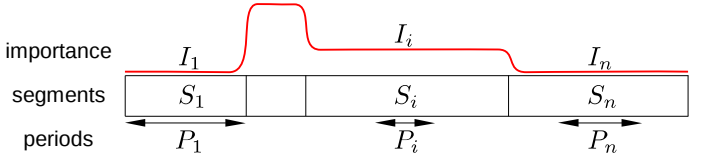
0: No evidence
1: Some hints suggest that the whole video could belong to the category
2: The segment contains significant evidence of the category
3: The segment alone classifies the video to the category

While audio can be used during annotation, we specify that if something is only mentioned in onscreen text or speech, then it should not be labeled as important.

In preliminary experiments we found that annotators tend to give too high importance to very short segments, that often have ambiguous segmentation and importance score. Therefore, we preprocess the ground-truth before the evaluation — we decrease the annotated importance for segments smaller than 4 seconds proportionally to the segment duration.

4.2 Evaluation metrics

We represent the manually annotated ground-truth segments $\mathbf{S} = \{S_1, \ldots, S_n\}$ of a video by:



An automatic temporal segmentation is represented by the sequence of segments $\mathbf{S}' = \{S'_1, \ldots, S'_m\}$.

To evaluate **segmentation** we define a symmetric f-score metric as:

$$f(\mathbf{S}, \mathbf{S}') = \frac{2 \cdot p(\mathbf{S}, \mathbf{S}') \cdot p(\mathbf{S}', \mathbf{S})}{p(\mathbf{S}, \mathbf{S}') + p(\mathbf{S}', \mathbf{S})}, \tag{3}$$

where the similarity of two segmentations $\mathbf{A}$ and $\mathbf{B}$ is

$$p(\mathbf{A}, \mathbf{B}) = \frac{1}{|\mathbf{A}|} |\{A \in \mathbf{A} \text{ st. } \exists B \in \mathbf{B} \text{ matching } A\}| \tag{4}$$

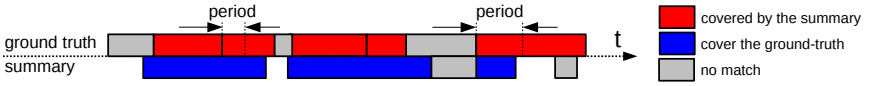
where $|\mathbf{A}|$ is the number of segments in $\mathbf{A}$. We consider segments A and B are matching if the temporal overlap over the union ratio is larger than 0.75, and when a segment has an annotated period, it is reduced to a sub-segment no shorter than the period, that maximizes the overlap over the union.

To evaluate **summarization** we define two metrics: the importance ratio and the meaningful summary duration.

A computed summary is a subset of the segments $\widetilde{\mathbf{S}} = \{\widetilde{S}_1, \cdots, \widetilde{S}_{\tilde{m}}\} \subset \mathbf{S}'$. We say a ground truth segment S_i is *covered* by a detected segment $\widetilde{S}_j$ if

$$\text{duration}\big(S_i \cap \widetilde{S}_j\big) > \alpha P_i \tag{5}$$

When the period equals the segment duration this means that a fraction α of the ground truth segment is covered by the detected segment. We use $\alpha = 80\%$ to enforce visually coherent summaries, which was validated using the ground-truth. Note that this definition allows covering several ground truth segments by a single detected segment, as in the following example:



Let $C(\widetilde{\mathbf{S}}) \subset \mathbf{S}$ be the subset of ground truth segments covered by the summary $\widetilde{\mathbf{S}}$. Given the duration of the summary $\mathcal{T}(\widetilde{\mathbf{S}}) = \sum_{j=1}^{\tilde{m}} \text{duration}(\widetilde{S}_j)$ and its total importance $\mathcal{I}(\widetilde{\mathbf{S}}) = \sum_{i \in C(\widetilde{\mathbf{S}})} I_i$, we define the *importance ratio* as

$$\mathcal{I}^*(\widetilde{\mathbf{S}}) = \frac{\mathcal{I}(\widetilde{\mathbf{S}})}{\mathcal{I}^{\max}(\mathcal{T}(\widetilde{\mathbf{S}}))}, \quad \text{with} \quad \mathcal{I}^{\max}(T) = \max_{\substack{\mathbf{A} \subset \mathbf{S} \\ \mathcal{T}(\mathbf{A}) \leq T}}^{\text{s.t.}} \mathcal{I}(\mathbf{A}) \tag{6}$$

We use the *maximum possible summary importance* $\mathcal{I}^{\max}(T)$ as a normalization factor. This normalization takes into account the duration and the redundancy of the video and ensures that $\mathcal{I}^*(\widetilde{\mathbf{S}}) \in [0, 1]$.

It turns out that maximizing the summary importance given the ground-truth segmentation and importance is NP-hard, as it is a form of knapsack problem. Therefore we use a greedy approximate summarization: we reduce each segment to its period, sort the segments by decreasing importance (resolving ties by favoring shorter segments), and constructing the optimal summary from the top-ranked segments that fit in the duration constraint.

A second measure is the *meaningful summary duration*, **MSD**. A meaningful summary is obtained as follows. We build it by adding segments by order of classification scores until it covers a segment of importance 3, as defined by the ground-truth annotation. This guarantees that the gist of the input video is represented at this length and measures how relevant the importance scoring is. Summaries assembling a large number of low-importance segments first are mediocre summaries and get a low MSD score. Summaries assembling high-importance segments first get a high MSD score. In our experiments we report the median MSD score over all test videos as a performance measure.

5 Results

5.1 Baselines

As the videos are annotated by several users, we can evaluate their annotations with respect to each other in a leave-one-out manner (**Users**). This quantifies the task's ambiguity and gives an upper bound on the expected performance.

For segmentation we use a shot detector (**SD**) of Massoudi et al. [24] as a baseline. *For classification* we use two baselines: one with the shot detector, where shots are classified with an SVM (**SD+SVM**) and one where the segments are selected by clustering instead of SVM scores (**KTS+Cluster**).

The SD+SVM baseline is close to an event detection setup, where a temporal window slides over the video, and an SVM score is computed for every position of the window [27,35]. However, we pre-select promising windows with the SD segmentation.

Table 1: Evaluation of segmentation and summarization methods on the test set of 100 videos. The performance measures are average f-measure for segmentation (higher is better) and median Meaningful Summary Duration for summarization (lower is better).

Method	Segmentation Avg. f-score	Summarization MSD (s)
Users	49.1	10.6
SD + SVM	30.9	16.7
KTS + Cluster	**41.0**	13.8
KVS	**41.0**	**12.5**

Clustering descriptors produces a representative set of images or segments of the video, where long static shots are given the same importance as short shots [5]. We use a simple k-means clustering, as the Fisher Vectors representing segments (see next section) can be compared with the L2 distance [26]. The summary is built by adding one segment from each cluster in turn. First we add segments nearest to each centroid, ordered by increasing duration, then second nearest, etc.

Our KVS method combines the KTS segmentation with a SVM classifier.

5.2 Implementation details

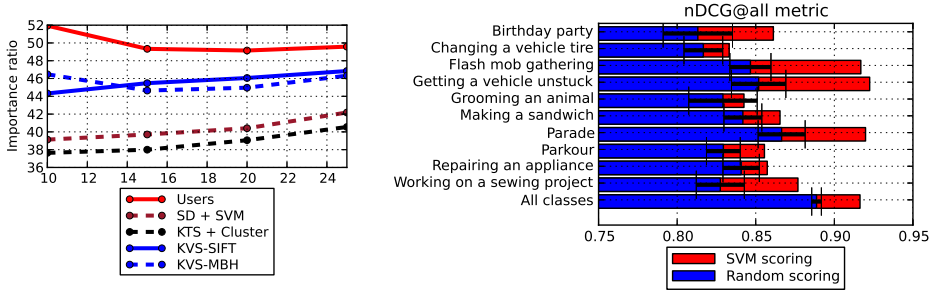
Video descriptors & classifier. We process every 5-th frame of the video. We extract SIFT descriptors on a dense grid at multiple scales. The local descriptors are reduced to 64 dimensions with PCA. Then a video frame is encoded with a Fisher Vector [26] based on a GMM of 128 Gaussians, producing a d=16512 dimension vector.

For segmentation we normalize frame descriptors as follows. Each dimension is standardized within a video to have zero mean and unit variance. Then we apply signed square-rooting and L_2 normalization. We use dot products to compare Fisher vectors and produce the kernel matrix.

For classification, the frame descriptors from a segment are whitened under the diagonal covariance assumption as in [26]. Then we apply signed square-rooting and L_2-normalization. The segment descriptor is the average of the frame descriptors. This was shown to be the best pooling method for frame descriptors [27,28].

The linear SVM classifier for each class is built from about 150 positive and 12000 negative training videos from the MED 2011 training dataset. The C parameter of the classifier is optimized using cross-validation.

We use grid-search on the 60-video validation set to optimize the parameters of the different methods. The shot detector (**SD**) has a single threshold T. Our



(a) (b)

Fig. 4: Summarization of the 100-video test dataset. (a) Importance ration of Equation (6) for different durations of the summary. (b) Correlation of SVM scores and scores assigned by users.

KVS method relies on a single parameter C that controls the number of segments (equation 1). For the clustering method, the optimal ratio of the number of clusters over the number of segments was found to be $1/5^{\text{th}}$.

On average, the annotated segments are 3.5 s long, and so are **SD** segments. The **KTS** method produces segments of 4.5 s on average.

5.3 Segmentation

Table 1 shows the segmentation quality of users and algorithms. For algorithms we average the f-scores of Equation (3) over segmentations from different users. For users we report the average f-score of the leave-one-out evaluation, i.e. we assume each user in turn to be the ground truth. The proposed approach KTS outperforms the competing method SD in terms of temporal segmentation performance. Surely, human segmentations are better than the algorithms', which means that the annotation protocol is consistent. Yet, the average f-score of users is not close to 100%, which suggests that the segment annotation task is somewhat subjective.

5.4 Summarization

The MSD metric in Table 1 shows that the temporal segmentation output by KTS has a significant impact on the summary's quality. Indeed, the SD+SVM method generally produces longer summaries than KTS+Cluster.

Fig. 4a shows the summarization quality for different summary durations. The user curve gives an upper bound on what can be achieved, by evaluating the consensus between annotators, following the leave-one-out procedure as before. The proposed approach, KVS, is the closest to the user curve. Again, KVS

clearly outperforms the competing methods KTS+Cluster and SD+SVM. We also run an experiment where the SIFT low-level descriptor is replaced by the MBH motion descriptor [36]. We get 2% improvement for 10 sec. and 1% drop for longer summaries compared to SIFT. A recent work [27] also reports little difference between SIFT and MBH on the MED 2011 dataset.

We also investigate how well SVM scores correlate with user importance, irrespective of the segmentation mismatches. We score ground truth segments from all videos of a class with SVM, and order the segments by descending score. Ideally segments with importance 3 should be in the top of the list, and non-relevant segments at the bottom. Since ground truth scores are discrete (from 0 to 3), we use the nDCG ranking metric [37], $\text{nDCG} = Z_p^{-1} \sum_{i=1}^{p} I^{(i)} (\log_2 i)^{-1}$, where $I^{(i)}$ is the annotated importance score of the i^{th} segment in the ranked list; p is the total number of segments over all videos of the class; Z_p is the normalization factor such that a perfect ranking's nDCG is 1.

Fig. 4b shows that, for 9 out of 10 classes, the SVM ranking is stronger than the random ranking.

Note that our approach does not require a ground-truth segmentation nor importance annotation for the training set. Therefore there can be some information loss due to unrelated clutter. To quantify this loss, we run an experiment by cross-validation on the test set where we use as positives during training the segments with the highest-importance scores, and observe an increase of 3 points in performance with respect to learning from full videos. Thus, a multiple instance learning (MIL) approach might give some improvement and is a possible extension of our approach.

Figure 5 illustrates our approach.

6 Conclusion

We proposed a novel approach to video summarization, called KVS, that delivers short and highly-informative summaries, that assemble the most important segments for a given video category.

KVS requires a set of training videos for a given category so that the method can be trained in a supervised fashion, but does not rely on segment annotations in the training set. We also introduced a new dataset for category-specific video summarization, MED-Summaries, that is publicly available, along with the annotations and the evaluation code that computes the performance metrics introduced in this paper.

Acknowledgements. This work was supported by the European integrated project AXES, the MSR/INRIA joint project, the LabEx PERSYVAL-Lab (ANR-11-LABX-0025), and the ERC advanced grant ALLEGRO. We thank the LEAR team members and Hyun Oh Song who helped with the annotation.

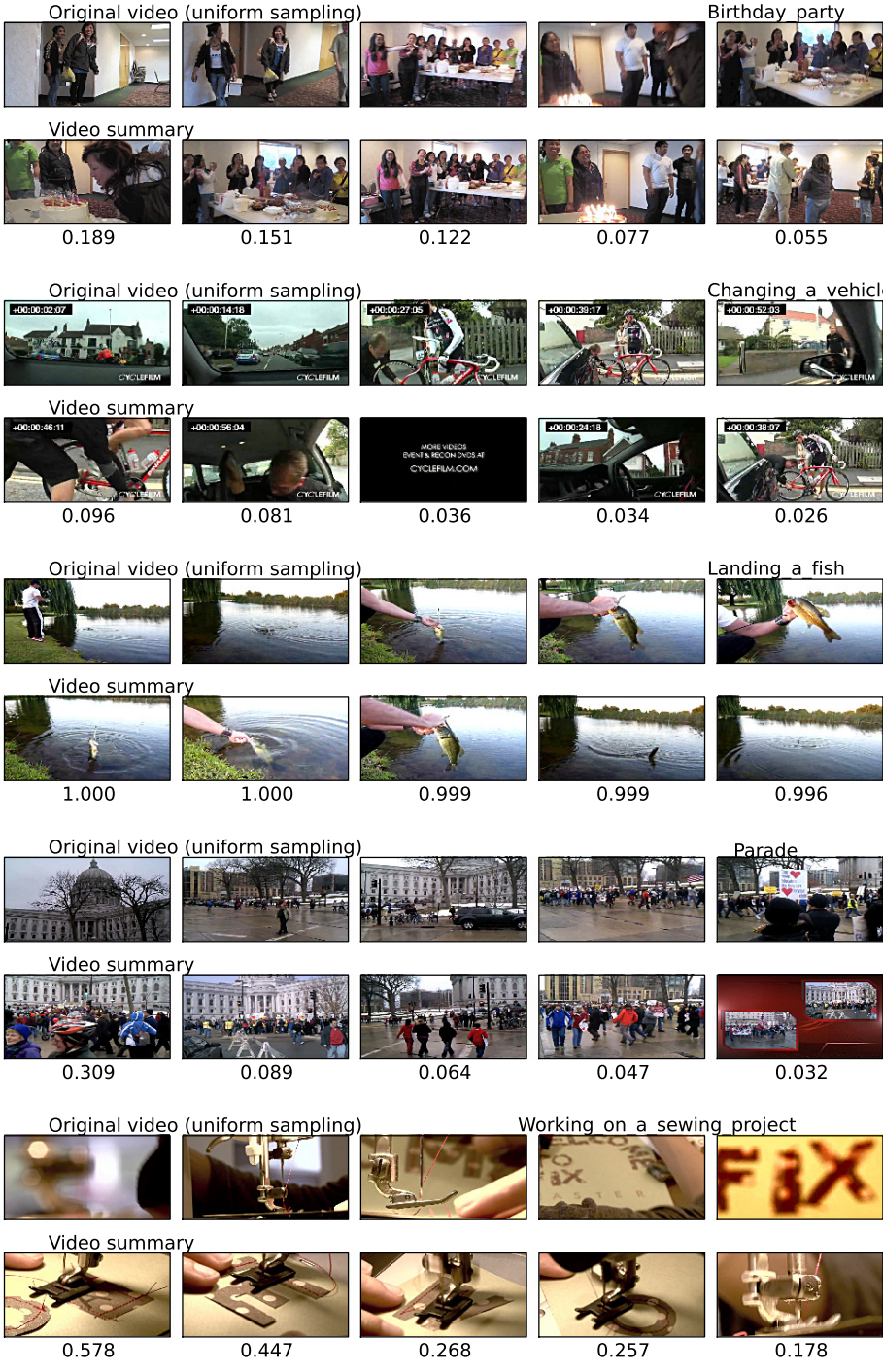


Fig. 5: Illustrations of summaries constructed with our method. We show the central frame in each segment with the SVM score below.

References

1. Liu, Y., Zhou, F., Liu, W., De la Torre, F., Liu, Y.: Unsupervised summarization of rushes videos. In: ACM Multimedia. (2010) 1, 2
2. de Avila, S., Lopes, A., et al.: VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters **32**(1) (2011) 56–68 1, 2, 3, 4
3. Lee, Y.J., Ghosh, J., Grauman, K.: Discovering important people and objects for egocentric video summarization. In: CVPR. (2012) 1, 3, 4
4. Wang, M., Hong, R., Li, G., Zha, Z.J., Yan, S., Chua, T.S.: Event driven web video summarization by tag localization and key-shot identification. Transactions on Multimedia **14**(4) (2012) 975–985 1, 3
5. Khosla, A., Hamid, R., Lin, C.J., Sundaresan, N.: Large-scale video summarization using web-image priors. In: CVPR. (2013) 1, 3, 4, 11
6. Lu, Z., Grauman, K.: Story-driven summarization for egocentric video. In: CVPR. (2013) 1, 3, 4, 5
7. Truong, B.T., Venkatesh, S.: Video abstraction: A systematic review and classification. ACM Transactions on Multimedia Computing, Communications, and Applications **3**(1) (2007) 3 2, 3, 4, 5
8. Over, P., Smeaton, A.F., Awad, G.: The Trecvid 2008 BBC rushes summarization evaluation. In: 2nd ACM TRECVID Video Summarization Workshop. (2008) 3, 4
9. Ma, Y.F., Hua, X.S., Lu, L., Zhang, H.J.: A generic framework of user attention model and its application in video summarization. Transactions on Multimedia (2005) 3
10. Li, K., Oh, S., Perera, A.G.A., Fu, Y.: A videography analysis framework for video retrieval and summarization. In: BMVC. (2012) 3
11. Ngo, C.W., Ma, Y.F., Zhang, H.J.: Video summarization and scene detection by graph modeling. Circuits and Systems for Video Technology **15**(2) (2005) 3, 4
12. Divakaran, A., Peker, K., Radhakrishnan, R., Xiong, Z., Cabasson, R.: Video summarization using Mpeg-7 motion activity and audio descriptors. In: Video Mining. Volume 6. Springer (2003) 3
13. Xie, L., Xu, P., Chang, S.F., Divakaran, A., Sun, H.: Structure analysis of soccer video with domain knowledge and hidden markov models. Pattern Recognition Letters **25**(7) (2004) 3
14. Rui, Y., Gupta, A., Acero, A.: Automatically extracting highlights for TV baseball programs. In: ACM Multimedia. (2000) 3
15. Sundaram, H., Xie, L., Chang, S.F.: A utility framework for the automatic generation of audio-visual skims. In: ACM Multimedia. (2002) 3
16. Zhao, B., Xing, E.P.: Quasi real-time summarization for consumer videos. In: CVPR. (2014) 3
17. Cong, Y., Yuan, J., Luo, J.: Towards scalable summarization of consumer videos via sparse dictionary selection. Transactions on Multimedia (2012) 3
18. Kim, G., Sigal, L., Xing, E.P.: Joint summarization of large-scale collections of web images and videos for storyline reconstruction. In: CVPR. (2014) 3, 4
19. Lin, C.Y.: Rouge: A package for automatic evaluation of summaries. In: Text Summarization Branches, ACL Workshop. (2004) 74–81 4
20. Hoiem, D., Efros, A.A., Hebert, M.: Automatic photo pop-up. ACM Transactions on Graphics **24**(3) (2005) 577–584 4
21. Tighe, J., Lazebnik, S.: Superparsing: scalable nonparametric image parsing with superpixels. In: ECCV. (2010) 4

22. Lezama, J., Alahari, K., Sivic, J., Laptev, I.: Track to the future: Spatio-temporal video segmentation with long-range motion cues. In: CVPR. (2011) 4
23. Grundmann, M., Kwatra, V., Han, M., Essa, I.: Efficient hierarchical graph-based video segmentation. In: CVPR. (2010) 4
24. Massoudi, A., Lefebvre, F., Demarty, C.H., Oisel, L., Chupeau, B.: A video fingerprint based on visual digest and local fingerprints. In: ICIP. (2006) 4, 10
25. Chasanis, V., Kalogeratos, A., Likas, A.: Movie segmentation into scenes and chapters using locally weighted bag of visual words. In: CIVR. (2009) 4
26. Perronnin, F., Sánchez, J., Mensink, T.: Improving the fisher kernel for large-scale image classification. In: ECCV. (2010) 11
27. Oneata, D., Verbeek, J., Schmid, C.: Action and Event Recognition with Fisher Vectors on a Compact Feature Set. In: ICCV. (2013) 7, 10, 11, 13
28. Cao, L., Mu, Y., Natsev, A., Chang, S.F., Hua, G., Smith, J.: Scene aligned pooling for complex video recognition. In: ECCV. (2012) 7, 11
29. Kay, S.M.: Fundamentals of Statistical signal processing, Volume 2: Detection theory. Prentice Hall PTR (1998) 5, 6
30. Harchaoui, Z., Bach, F., Moulines, E.: Kernel change-point analysis. In: NIPS. (2008) 6
31. Harchaoui, Z., Cappé, O.: Retrospective mutiple change-point estimation with kernels. In: Workshop on Statistical Signal Processing, IEEE (2007) 768–772 6
32. Hastie, T., Tibshirani, R., Friedman, J.: The elements of statistical learning: data mining, inference and prediction. 2 edn. Springer (2009) 6
33. Arlot, S., Celisse, A., Harchaoui, Z.: Kernel change-point detection. arXiv:1202.3878 (2012) 6
34. Crow, F.C.: Summed-area tables for texture mapping. In: ACM SIGGRAPH Computer Graphics. Volume 18. (1984) 207–212 6
35. Gaidon, A., Harchaoui, Z., Schmid, C.: Temporal localization with actoms. PAMI (2013) 10
36. Wang, H., Kläser, A., Schmid, C., Liu, C.L.: Dense trajectories and motion boundary descriptors for action recognition. IJCV (2013) 13
37. Manning, C.D., Raghavan, P., Schütze, H.: Introduction to information retrieval. Volume 1. Cambridge (2008) 13