Local Descriptors in Temporal Video Segmentation: A Performance Evaluation

Im Sio Kei
Macao Polytechnic Institute
Macao, China
marcusim@ipm.edu.mo

Pedro Martins
CISUC, DEI
University of Coimbra
Coimbra, Portugal
pjmm@dei.uc.pt

***Abstract*—Local image descriptors are widely used and play an important role in temporal video segmentation tasks. However, the study of descriptors performance under a temporal video segmentation context has been overlooked. In this paper, we present a qualitative and quantitative performance evaluation of state-of-the-art local image descriptors in temporal video segmentation tasks. Additionally, we complement our study with an analysis of the applicability of local descriptor-based temporal segmentation to video summarization**

Keywords—local descriptors; video segmentation; video summarization

I. INTRODUCTION

Many computer vision and image analysis tasks rely on spatial or temporal segmentation to obtain a more compact representation of the image/video and highlight meaningful information. In the particular case of video sequences, segmentation usually corresponds to the division of the whole sequence into disjoint spatio-temporal segments. Image descriptors, i.e., algorithms aimed at providing a summarized description of image patches, are often used in temporal video segmentation. Descriptors provide compact representations of the frames, allowing a more efficient comparison of frames. Despite being widely used in video segmentation and playing an important role in this task, as they serve as a basis for subsequent actions, the study of their performance in this context has been overlooked.

In this paper, we qualitatively and quantitatively assess the performance of descriptors in video segmentation tasks. This work is part of a study on video summarization, which is a technique aimed to build a concise yet informative representation of a full-length video.

In most summarization frameworks, temporal segmentation is the first step in the whole process. Segmentation often consists in detecting transitions (boundaries) between shots, i.e., a sequence of frames depicting a continuous action captured from a single operation of a single camera [1]. Shot detection can be based on different techniques (e.g., color histograms [2], global motion [3], local motion [4], or even bags-of-words [5]). The typical shot boundary segmentation assumes the existence of edited videos such as movies or documentaries. User generated videos are usually unedited, containing often one single shot [6]. As such, it is important to detect not only shot boundaries but also less abrupt changes. The kernel temporal segmentation proposed by Potapov et al. [7] is especially designed to deal with unedited videos. The method is based on change point detection, which is a statistical framework to detect changes in time series, and it takes into account the differences between all pairs of frames.

While our evaluation is not based on the construction of video summaries, it allows to infer about the applicability of descriptors in summarization tasks. We are primarily interested in finding answers and clues to the following questions: How similar are segmentations when they are based on different descriptors? Can a local descriptor-based segmentation easily identify the most relevant segments (as classified by humans)? To our knowledge, this type of evaluation was only done in [8]. However, it was a comparison between local and global descriptors. Here, we focus on several local descriptors and on a particular type of segmentation, kernel temporal segmentation [7], to build our evaluation framework.

II. PERFORMANCE EVALUATION

Our performance evaluation is mainly based on the segmentations provided by different descriptors. Since the segmentation step is crucial to the whole summarization process and it immediately reflects the performance of the descriptors, we do not focus on the construction of video summaries. To offer a more insightful analysis, we analyze the propensity of the different descriptors to present segmentations in which relevant segments, as classified by humans, are properly segmented.

We follow the kernel temporal segmentation proposed in [7]. The rationale behind the choice of this approach is basically two-fold: first, as it detects change points that correspond to shot boundaries as well as less abrupt boundaries between frames, it is an appropriate choice for summarization; second, it can be used with different types of descriptors. We opted for three types of state-of-the-art descriptors: SIFT [9], SURF [10] and ORB [11]. As for SIFT, we considered two variants: one based on a dense grid of points computed over the frames [12], which we coined as dSIFT (dense SIFT), and a second one using the locations provided by the Difference of Gaussians detector (DoG), i.e, the standard SIFT keypoint

detection. The remaining descriptors use keypoints as in their original implementations. This particular choice of methods allows us to compare state-of-the-art solutions based on different approaches: histogram-based (SIFT and SURF) and binary string-based (ORB). In the case of SIFT, we can also compare dense to sparse representations.

A. Dataset

All the evaluation was performed using videos from the SumMe dataset [6]. This collection consists of 25 sequences covering topics such as sports, holiday and events. The editing in these videos is minimal or inexistent, which makes them good candidates for summarizations, as they contain redundant and uninformative parts. Fig. 1 presents frames for each video. The SumMe dataset contains three categories of videos: moving, static, and egocentric. Our selection contains a video from each category.

B. Parameter Settings

As in [7], we processed every 5-th frame of the video. The local descriptors are reduced to half of their dimensions through Principal Component Analysis (PCA). Then, video frames were encoded with Fisher Vector using a Gaussian Mixture Model of 64 Gaussians. Finally, to construct the kernel matrix, each dimension was normalized to have zero mean and unit variance. Then a signed square-rooting and an *L2*-normalization were applied. The dot product was used to compute the kernel matrix. As for the descriptors, we ran the OpenCV implementations with default parameter values. Concerning the kernel-based temporal segmentation, we used the implementation provided and maintained by the authors. The segmentation was performed either with a fixed number of segments, n, or with an automatic selection of this number. In the latter case, a maximum number of points, n_{max}, was defined.

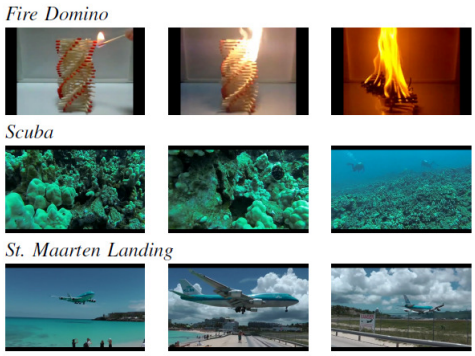

Fig. 1. Videos used in the experiments. *Fire Domino* (static), *Scuba* (egocentric), and *St. Maarten Landing* (moving).

III. RESULTS AND DISCUSSION

For each video, we present the segmentation produced by each one of the descriptors. The change points are depicted as vertical lines along the time line. In addition, we add the relevance scores given by humans to the video/frames segments [6]. As previously mentioned, we are not constructing video summaries. Our intent is to assess to what extent these low-level temporal segmentations can provide estimates of a human segmentation aimed at summarizing the content.

We also show the similarity between segmentations. Instead of computing the percentage of overlap between segments, we look for common change points. Two change points are considered the same if they are 5 frames within each other. Given two sets of change points, T_1 and T_2, the similarity between them is given by $sim(T_1, T_2) = \frac{|T_1 \cap T_2|}{|T_1 \cup T_2|}$.

Fig. 2 depicts the segmentation for Fire.Domino using a fixed number of segments. We verify that parts that were voted as the most relevant ones contain change points regardless of the descriptor that is being used. In fact, the four segmentations tend to present similar results. It should also be noted that the ORB-based segmentation does not produce change points in the initial frames (0-200), which is one of the least informative parts of the sequence.

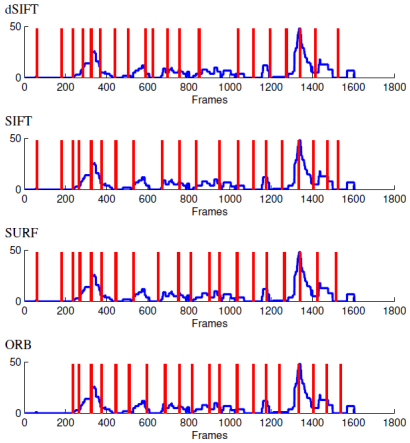

Fig. 2. Segmentation for the *Fire Domino* sequence using a fixed number of change points ($n = 20$).

Table I contains the similarity values between the different segmentations. There is not a considerable discrepancy among these values: they range between 0.29 (dSIFT, SURF) and 0.5 (SURF, ORB).

TABLE I. SIMILARITY BETWEEN SEGMENTATIONS OF THE *FIRE DOMINO* SEQUENCE.

sim(.,.)	dSIFT	SIFT	SURF	ORB
dSIFT	-	0.38	0.29	0.3
SIFT		-	0.48	0.4
SURF			-	**0.5**

In Fig. 3, we present the segmentations for *Fire Domino* using an optimal number of segments. Here, we observe that a keypoint-based description tends to produce an over-segmentation, while dSIFT provides a more balanced segmentation. In addition, dSIFT produces a segmentation in which the change points tend to coincide with either peaks or valleys of the relevance score.

Fig. 4 presents the segmentation results for the *Scuba* sequence with a fixed number of segments. Here, we observe that several change points occur in less relevant image parts. Table II summarizes the similarity between segmentations. SIFT and SURF share more change points, whereas dSIFT and SURF have a reduced number of common change points. Nonetheless, all the segmentations show similar change points at the most informative parts.

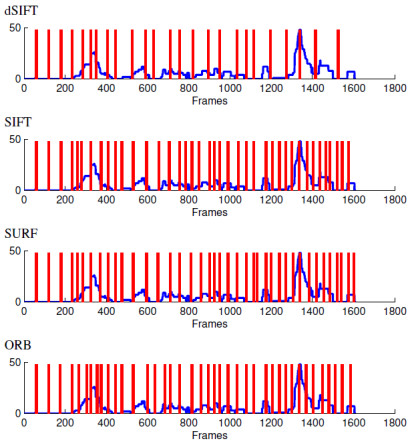

Fig. 3. Segmentation for the *Fire Domino* sequence using an automatic number of change points ($n_{max} = 40$).

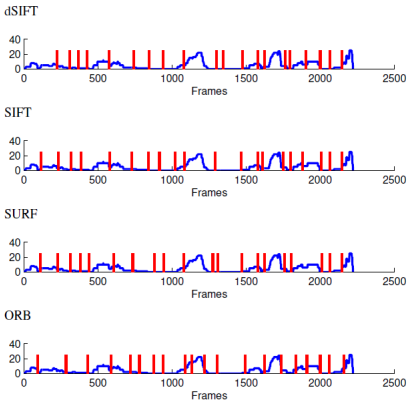

Fig. 4. Segmentation for the *Scuba* sequence using a fixed number of change points ($n = 20$).

TABLE II. SIMILARITY BETWEEN SEGMENTATIONS OF THE SCUBA SEQUENCE.

sim(.,.)	dSIFT	SIFT	SURF	ORB
dSIFT	-	0.25	0.38	0.22
SIFT		-	**0.43**	0.08
SURF			-	0.15

Fig. 5 presents the segmentation results for the *Scuba* sequence with an automatic selection of the number of change points. Here, we observe again that keypoint-based descriptions provide an over-segmentation with several change points occurring at least informative parts.

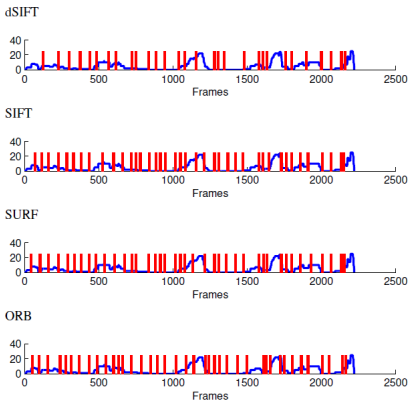

Fig. 5. Segmentation for the *Scuba* sequence using an automatic number of change points ($n_{max} = 40$).

Fig. 6 depicts the segmentation for *St. Maarten Landing* with a fixed number of segments. For this particular sequence, one interesting observation is that dSIFT provides a clear segmentation of the part voted as the most relevant one (around frame 1300). The similarity between segmentations is shown in Table III. ORB and SIFT provide the highest number of common change points. It is also worth noting that dSIFT and SIFT do not share many change points.

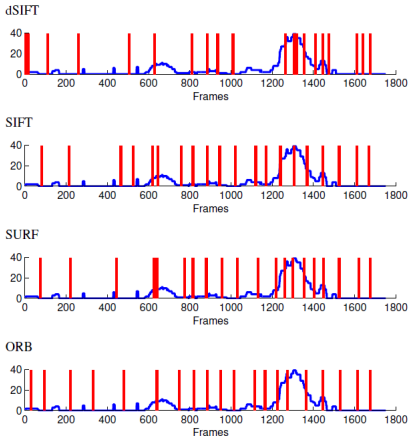

Fig. 6. Segmentation for the *St. Maarten Landing* sequence using a fixed number of change points ($n = 20$).

TABLE III. SIMILARITY BETWEEN SEGMENTATIONS OF THE ST. MAARTEN LANDING SEQUENCE.

sim(.,.)	dSIFT	SIFT	SURF	ORB
dSIFT	-	0.18	0.21	0.08
SIFT		-	0.33	**0.48**
SURF			-	0.33

Fig. 7 depicts the segmentations for the *St. Maarten Landing* sequence with an automatic selection of the number of change points. Here, we observe two extreme situations: dSIFT produces very few segments and the keypoint-based descriptions provide over-segmentations. However, in the latter case, we observe a tendency to properly segment informative parts.

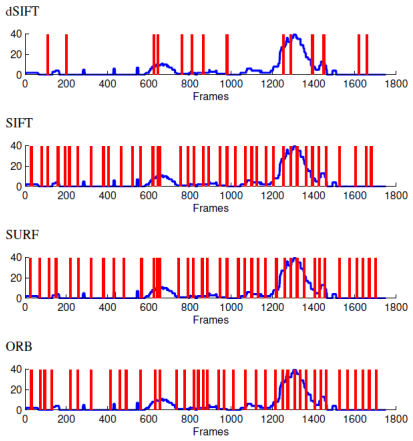

Fig. 7. Segmentation for the *St. Maarten Landing* sequence using an automatic number of change points (n_{max} = 40).

Table IV summarizes the average running times for the different descriptions and segmentations of the *Fire Domino* sequence after 10 runs. Since SIFT, SURF and ORB include a keypoint detection, this step was considered as part of the description process. Although ORB description includes keypoint detection, this algorithm requires less time to compute the descriptor vectors. On the other hand, SIFT and SURF running times are affected by keypoint detection stages. SIFT running time is twice of dSIFT time. It is also worth noting that SURF takes longer than SIFT to compute the feature vectors. This is not an unexpected result. Even though SURF tends to be faster than SURF per keypoint, the number of detected features is not the same and we used the extended 128 dimensions SURF as opposed to the standard 64 dimensions version. dSIFT takes on average approximately 5 seconds more than ORB to compute the description. As for the segmentation running times, the histogram-based descriptors show similar times, whereas ORB has a significantly lower time. The reduced number of dimensions (32 dimensions before PCA) of ORB vectors contribute to lower running times

TABLE IV. AVERAGE RUNNING TIMES FOR THE DIFFERENT DESCRIPTIONS AND SEGMENTATIONS OF THE FIRE DOMINO SEQUENCE.

	Description (secs.)	Segmentation (secs.)
dSIFT	32.2±0.2	12.8±1.1
SIFT	64.4±1.2	10.5±0.7
SURF	80.5±1.5	10.6±0.3
ORB	27.4±0.4	6.9±0.3

IV. CONCLUSIONS AND FUTURE WORK

In this paper, we presented a small-scale test aimed at analyzing the performance of three different state-of-the-art photometric descriptors, SIFT, SURF and ORB, in video segmentation tasks. As for SIFT, we used dense (based on a regular grid of points) and sparse (based on the detection of keypoints) representations. Our goal was to analyze not only the segmentation results per se, but to examine the applicability of the different descriptor-based segmentations to video summarization. All the experiments were performed using kernel-based temporal segmentation, which is a segmentation technique based on the change point detection approach. We opted for this technique because it has shown interesting and promising results in the video summarization problems and it can be used with different high-dimensional descriptors.

Dense representations are a good option for kernel-based segmentation. There are two major advantages in using dense representations: the feature detection step is not required and more image information is provided to perform the segmentation. If there are strong temporal constraints, we believe that binary-based approaches, such as ORB, can be a valid option due to their efficiency. Our experiments also showed that there are no significantly discrepant results among the different segmentations and relevant segments, as classified by humans, tend to be properly segmented. As future work, we intend to analyze the performance of descriptors in a larger-scale test that will take into account the specificities of different video categories.

REFERENCES

[1] E. Cahuina and G. C. Chavez, "A new method for static video summarization using local descriptors and video temporal segmentation," in Proc. of the 2013 26th SIBGRAPI - Conference on Graphics, Patterns and Images (SIBGRAPI), pp. 226–233.

[2] M. A. Smith and T. Kanade, "Video skimming and characterization through the combination of image and language understanding techniques," in Proc. of the 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'97), Jun 1997, pp. 775–781.

[3] Y. Murai and H. Fujiyoshi, "Shot boundary detection using cooccurrence of global motion in video stream," in Proc. of the 19th International Conference on Pattern Recognition (ICPR'08), Dec 2008, pp. 1–4 .

[4] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," ACM Trans. Multimedia Comput. Commun. Appl., vol. 3, no. 1, 2007.

[5] V. Chasanis, A. Kalogeratos, and A. Likas, "Movie segmentation into scenes and chapters using locally weighted bag of visual words," in Proc. of the ACM International Conference on Image and Video Retrieval (CIVR'09), 2009, pp. 35:1–35:7.

[6] M. Gygli1, H. Grabner, H. Riemenschneider, and L. V. Gool, "Creating summaries from user videos," in Proc. of the 13th European Conference on Computer Vision (ECCV'13), 2014.

[7] D. Potapov, M. Douze, Z. Harchaoui, and C. Schmid, "Categoryspecific video summarization," in Proc. of the 13th European Conference on Computer Vision (ECCV'13), 2014.

[8] M. Kogler, M. D. Fabro, M. Lux, K. Schoeffmann, and L. B ¨ osz ¨ ormenyi, "Global vs. local feature in video summarization: Experimental results," in Proc. of the 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies (SeMuDaTe'09) in conjunction with the 4th International Conference on Semantic and Digital Media Technologies (SAMT 2009), 2009.

[9] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," International Journal of Computer Vision, vol. 60, pp. 91–110, 2004.

[10] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," Computer Vision and Image Understanding, vol. 100, no. 3, pp. 346–359, 2008.

[11] E. Rublee, V. Rabaud, K. Konolige, and G. Bradski, "ORB: An Efficient Alternative to SIFT or SURF," in Proc. of the 13th IEEE International Conference on Computer Vision (ICCV'11), 2011, pp. 2564–2571.

[12] L. Fei-Fei and P. Perona, "A Bayesian Hierarchial Model for Learning Natural Scene Categories," in Proc. of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, 2005, pp. 524–531.