Speeded-up Video Summarization Based on Local Features

Javier Iparraguirre*†
*Electronics Engineering Department
Universidad Tecnológica Nacional, Facultad Regional Bahía Blanca
Bahía Blanca, Argentina
j.iparraguirre@computer.org

Claudio Delrieux†
†Imaging Sciences Lab
Universidad Nacional del Sur
Bahía Blanca, Argentina
cad@uns.edu.ar

***Abstract*—Digital video has become a very popular media in several contexts, with an ever expanding horizon of applications and uses. Thus, the amount of available video data is growing almost limitless. For this reason, video summarization continues to attract the attention of a wide spectrum of research efforts. In this work we present a novel video summarization technique based on tracking local features among consecutive frames. Our approach operates on the uncompressed domain, and requires only a small set of consecutive frames to perform, thus being able to process the video stream directly and produce results on the fly. We tested our implementation on standard available datasets, and compared the results with the most recent published work in the field. The results achieved show that our proposal produces summarizations that have similar quality than the best published proposals, with the additional advantage of being able to process the stream directly in the uncompressed domain.**

***Keywords*-Video summarization; video processing; video skimming; keyframe selection; local features**

I. Introduction

In recent years, digital video became a vastly used media in several contexts and applications. The generation and availability of digital video from different sources is growing at an exponential rate [6]. This fact poses several challenges for users, in order to manipulate a vast collection of videos. Video summarization is aimed to provide condensed versions in a consistent and predictable way. There are multiple aspects to consider in the manipulation of digital video. The information to consider includes the image sequences, the sound, the illumination of the scene, and the occluded parts of the relevant objects. An effective summarization also must take into account the psychological features of the human perceptual system for an adequate processing and manipulation. These requirements makes video summarization a very complex task.

There are several ways to characterize video summarization strategies. One of the most popular classification is based on the output of the summary [7]. A *keyframe-based* summarization consist on obtaining a storyboard or a static summarization. By contrast, a *sequence-based* summarization, produces a short version of the original material (called video skim in the literature). Another relevant classification of the summarization technique consist on determining if the method requires the complete video sequence to proceed, or if it may perform directly over the video stream, which makes it adequate for online processing. Finally, there is a third way to classify a summarization system, depending on the way the video information is accessed. It is possible to apply the processing in the compressed or in the uncompressed domain. The compressed domain techniques use the features that are provided by the existing video encoders. By contrast, uncompressed summarization uses all the information available in the frames.

In this paper we present a novel technique that can produce either keyframe or sequence summarizations. We work only with the local features of a small set of consecutive (uncompressed) frames. Our method operates directly with the video stream, and since it performs at an adequate speed, it can be applied to summarize video in real time. Our technique is based on the Speeded-up Robust Features (SURF) algorithm [3], thus we call our method Speeded-up Video Summarization. The performance of our approach was tested with a set of standard datasets. The resulting summarizations are similar in terms of quality to the ones produced by the best published results in the field. In addition, our processing was performed directly on the uncompressed video stream, which makes it useful for live applications.

In the next Section we introduce the previous works that are related to our approach to video summarization. Section III documents our method and presents the most important implementation details. The results are presented in Section IV. Finally, we discuss the conclusions and propose further work in Section V.

II. Related Work and Concepts

Research in video summarization has been very active recently, and therefore there are many proposed techniques. The earliest proposals followed variations of a similar approach, which starts creating a large matrix that contains the features extracted from the individual frames, and then the frames that best represent the dataset are selected based on a particular criterion. Most of these methods operate in the uncompressed domain. Even though these methods provide adequate results, they require huge amounts of space and they are not suitable for online summarization.

978-0-7695-5140-1/13 $26.00 © 2013 IEEE
DOI 10.1109/ISM.2013.70

Guan et al. [5] proposed a video summarization method based on global and local features. Their approach involves two steps. In first place, they represent each video frame with the a feature called CEED (Color and Edge Directivity Descriptor). After the feature representation, they perform frame clustering using k-means algorithm. This classification allows them to divide the entire video into separarte takes using global features. The second step consists on selecting keyframes within each take using Lowe's SIFT descriptor (local features). Afterwards, they build a global pool with the keypoints detected in a particular frame. After the pool is constructed, they select a set of keyframes that covers the global feature pool at a selected percentage. Finally, they create the summary reordering the keyframes in chronological order. In a later work Guan et al. [4] presented another summarization method. In the later thechnique they discard the use of global features. However, their method still requires the entire video stream to make a decision about the keyframes.

Summarization methods in the compressed domain, on the other hand, take advantage of the video compression algorithms. Most of the summarization approaches in this domain are focused on particular applications. The main advantage of applying summarization to the compressed video is that it requires less computational power. One relevant limitation of working in the compressed domain is that the algorithms are constrained to the features of the compression methods.

Almeida et al. [1] [2] presented a video summarization technique that is adequate for online operation. Their proposal relies on the MPEG compression standard. Their method consist on three steps: feature extraction, content selection, and noise filtering. For each frame, they obtain features building a color histogram using the HSV color space as reference. They use ZNCC (Zero-mean Normalized Cross Correlation) as a metric to determine the distance between two frames. In order to decide if a frame is relevant, dissimilarity among successive frames is computed, and then a threshold is applied to determine relevance. After contents selection, they compute a color distribution histogram and a gradient orientation histogram for each relevant frame. Finally, they filter noisy frames if one of the two histograms has a normalized variance that is greater than a predetermined threshold.

III. Our Proposal

In general terms, our method can be divided into a sequence of steps to be applied per-frame. Figure 1 shows the algorithm behind our approach. The first step consist on local feature detection. We apply SURF to every frame. For each frame we obtain a set of keypoints (or *ipoints*) and a set of descriptors.

Afterwards, we match the features of the actual frame with the features of the previous frame. In this way, we obtain the number of features that are common to the actual and the previous frame. We define this parameter as the current *common feature number* (CFN). On every frame we calculate the average CFN of the latest N frames. Empirical tests show that, in all cases, averaging the prior 10 CFNs is enough for an adequate performance on keyframe detection.

To detect a potential keyframe we analyze the percentual change (in absolute value) of the current CFN versus the average CFN among the previous pairs of frames. If this absolute value is above a given threshold, then we assume that the change among the two frames being considered is unusual in the given take. Therefore the second frame should be considered to be a potential keyframe. The threshold values are therefore between 0 and 1, 0 means that *any* change among frames will trigger a keyframe detection, and 1 means that no change at all will trigger a keyframe detection. This threshold value, then, allows to control the *sensitivity* of the system.

After we having detected a potential keyframe, we match the feature descriptors of the candidate against the descriptors of the last detected keyframe. We evaluate a "noise amount", obtaining a percentual value between the amount of feature points on the last keyframe and the number of matches detected in the new candidate keyframe. In a similar way as we did with the sensitivity threshold, we defined *noise tolerance threshold*, where 0 means that any difference between the candidate frame and the latest detected keyframe will turn the candidate into a new keyframe. By contrast, 1 means that no candidate will ever become a new keyframe.

A. Creating Online Skims and Storyboards

In order to produce a sequence-based summarization (video skim), once the algorithm described before detects a keyframe, we add N consecutive frames after the selected keyframe. This allows human viewers to follow what happened the moment after the keyframe was detected. As a result it is possible to obtain a comprehensible compressed version of the most relevant moments of the original stream. After a series of tests with human users, we discovered that if the take after the keyframe elapses less than half a second, the summarization is perceived too "jumpy". However, longer takes obviously make the summarization longer. For this reason, adding 30 consecutive frames after the keyframe appears to be the best compromise solution. In case that a new keyframe appears before the 30 frames, we reset the frame counter to the new situation.

It is important to highlight two features of our skim generation process. The first advantage of our method is that the skim is performed on the fly over the video stream. As a result, our method can produce either a keyframe storyboard or a short movie as it processes the video. The other advantage of our approach is that the viewer can follow the events inside the video skim.

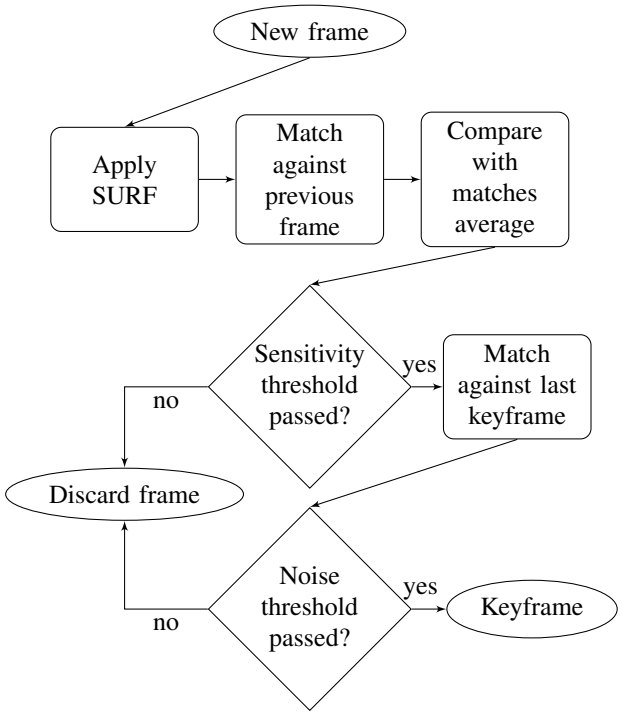

Figure 1. Flowchart of our algorithm

B. Comparison with Previous Works

Our method can be considered similar to Guan et. al. [5]. However, they need to process the entire dataset before they choose the keyframes and they use global features, thus making their proposal unfeasible for online video processing. Another differentiation is that they downsample datasets to 5 FPS before running experiments, while our method does not require previous downsamplig. In a later work, Guan et. al. [4] proposed a summarization method based only on keypoints. However, they still use a global pool of keypoints before they choose the keyframes.

We also share some common aspects with Almeida et. al. [1], [2]. In their work, they use a frame-by-frame dissimilarity approach that allows the method to proceed online. However, our method works on the uncompressed domain and with local features, which makes it able to trigger keyframe detection even when tiny details change among frames. Another benefit of our approach is that we are not constrained to the block division of the MPEG compression method.

Table I presents together the most relevant features of the related work and how our proposal fits in the general landscape. It is important to highlight that our proposal is the only one that works in the uncompressed domain and requires only a fraction of the dataset. In contrast to other methods that work in the compressed domain, our method is suitable for online summarization. These characteristics make our method unique.

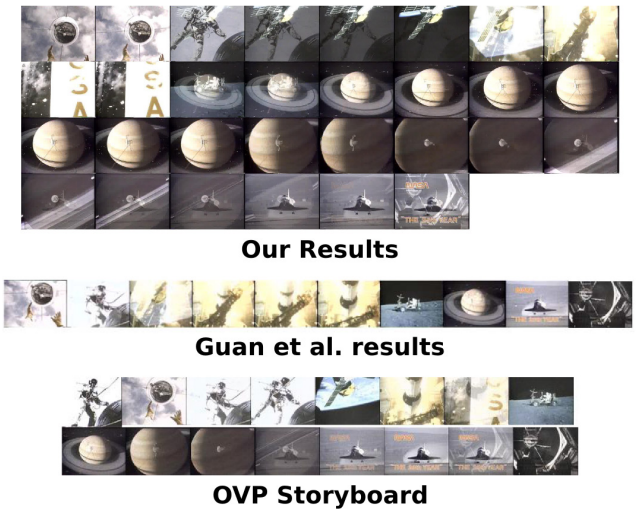

Figure 2. Keyframes of the video NASA 25th Anniversary Show, Segment 01

IV. Results

Evaluating results in video summarization is not a straightforward task. In this work, we apply the same comparison criterion as in Guan et. al. [5]. We run our algorithm over the Open Video Project (OVP)[1] dataset. As a performance evaluation, we compared our results against the OVP ground truth storyboard, and against results of other published proposals. For all the results shown in this work we selected a 0.25 sensitivity threshold and a 0.5 noise threshold. The rest of the mentioned parameters were left as stated previously. All the outcomes presented in this work and the corresponding skims are available at a public website[2].

Figure 2 shows the output keyframes of the video *NASA 25th Anniversary Show Segment 1*. In the figure it is possible to observe our results, the OVP storyboard, and Guan et al. [5] results. In general terms it is possible to say that our approach produces results that are comparable to state of the art summarization techniques. In the case of the sequence that involves the Saturn planet we observe that our method is sensitive to a zoom scene. This is a reasonable behavior since we are computing only the local features.

Figure 3 shows the output keyframes of the video *A New Horizon Segment 2*. In the figure it is possible to observe our summarization results, the OVP storyboard, and Almeida et al. [1] results. Our method detects most of the relevant frames, including the one that contains the divisions of the map, which seems to be missed by the other methods.

[1] http://www.open-video.org/
[2] http://www.javieriparraguirre.net/video-summarization/

Table I
SUMMARY OF THE MAIN FEATURES OF PREVIOUS WORK

Method	Domain	Dataset requirement	Brief description
Guan et al. [5]	Uncompressed	Complete	Global features, k-means clustering, local features, frame selection based on pool
Guan et al. [4]	Uncompressed	Complete	Local features, keypoints pool construction, coverage and redundancy to pick frames
Almeida et al. [1] [2]	Compressed	Partial	DC image, color histograms, clustering, and filtering
Our Proposal	Uncompressed	Partial	SURF, match consecutive frames, average matches, match to last keyframe to filter

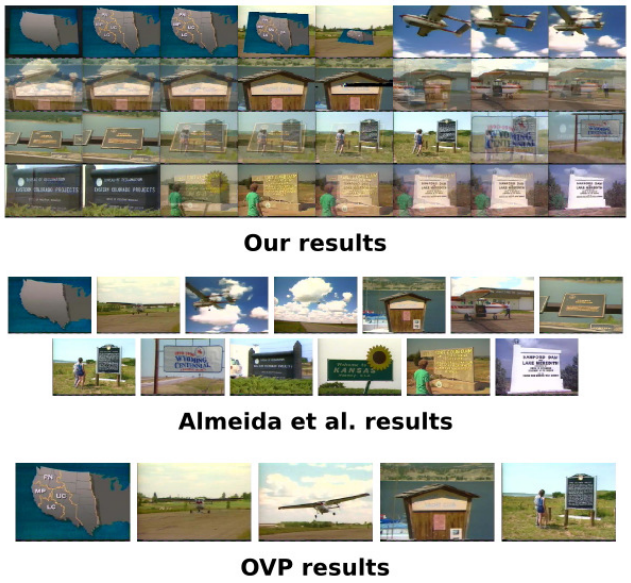

Figure 3. Keyframes of the video A New Horizon, Segment 2

A. Advantages and Limitations of our Approach

Our approach to summarization produces comparable results to the state of the art methods. Using local features allows us to detect even tiny changes among frames. One of the strongest features of our contribution is that we do not require the entire video to produce a summarization. This fact allow us to create an online summarization, working directly with the video stream. The use of the SURF algorithm contributes to an on-the-fly processing solution that produces plain keyframes of a video skim as required by viewers.

In cases where there are zoom-in or object translations within the take, our method may detect spurious keyframes. In skimming or storyboarding movies, homemade videos, or similar sources, this may be an undesired feature, but in video surveillance being able to detect movement or size change in objects is obviously a requirement. This feature of our proposal can be easily changed by disregarding the position tags in the ipoints vector provided by the SURF algorithm.

V. CONCLUSIONS AND FUTURE WORK

In this work we introduced a novel video summarization technique that uses local features, works in the uncompressed domain, and produces an online output. To the best of our knowledge, this is the first proposal to be able to perform over the uncompressed video stream. We tested our result on standard datasets, where our method produces results comparable in quality to the available proposals in the literature. Also, as compared to those proposals, our technique requires to set only a few working parameters, making it much easier and flexible to use. We presented the advantages and limitations of our proposal.

As future work we are planning to implement the same algorithm on GPU architectures. Because of the nature of the application, we expect to obtain a significant acceleration using GPUs. Finally, we are currently working on applying these results to provide unsupervised video tagging and annotation functions.

REFERENCES

[1] J. Almeida, N. J. Leite, and R. d. S. Torres. Online video summarization on compressed domain. *Journal of Visual Communication and Image Representation*, 2012.

[2] J. Almeida, R. Torres, and N. J. Leite. Rapid video summarization on compressed video. In *Multimedia (ISM), 2010 IEEE International Symposium on*, pages 113–120. IEEE, 2010.

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.

[4] G. Guan, Z. Wang, S. Lu, J. Deng, and D. Feng. Keypoint-based keyframe selection. *Circuits and Systems for Video Technology, IEEE Transactions on*, 23(4):729–734, 2013.

[5] G. Guan, Z. Wang, K. Yu, S. Mei, M. He, and D. Feng. Video summarization with global and local features. In *Multimedia and Expo Workshops (ICMEW), 2012 IEEE International Conference on*, pages 570–575. IEEE, 2012.

[6] A. G. Money and H. Agius. Video summarisation: A conceptual framework and survey of the state of the art. *Journal of Visual Communication and Image Representation*, 19(2):121–143, 2008.

[7] B. T. Truong and S. Venkatesh. Video abstraction: A systematic review and classification. *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, 3(1):3, 2007.