VSUMM: An Approach for Automatic Video Summarization and Quantitative Evaluation

Sandra E. F. de Avila[†], Antonio da Luz Jr.[†‡], Arnaldo de A. Araújo[†], and Matthieu Cord[§]
[†]Computer Science Department — Federal University of Minas Gerais
Av. Antônio Carlos, 6627, Pampulha, CEP 31270–901, Belo Horizonte, Minas Gerais, Brazil
[‡]Federal Technical School of Palmas
Setor Agroindustrial, CEP 77600–000, Paraíso, Tocantins, Brazil
[§]Laboratoire d'Informatique de Paris 6 — Université Pierre et Marie Curie
104 Avenue du Président Kennedy, 75016 Paris, France
{sandra, daluz, arnaldo}@dcc.ufmg.br, matthieu.cord@lip6.fr

Abstract

In this paper, we propose an approach for video summarization (VSUMM). The video summaries are generated based on visual features. A factorial experiment is designed to analyze the relative impact of the attributes. We demonstrate the validity of the VSUMM approach by testing it on a collection of videos from Open Video Project. We provide a comparison among results of the proposed summarization technique with Open Video storyboard. A subjective evaluation showed that the summaries are produced with good quality.

1. Introduction

The demand for various multimedia applications is rapidly increasing due to the recent advance in the computing and network infrastructure, together with the widespread use of digital video technology [25]. Among the key elements for the success of these applications is how to effectively and efficiently manage and store a huge amount of audio visual information, while at the same time providing user-friendly access to the stored data. This has fueled a quickly evolving research area known as *video summarization*.

According to [18, 25], there are two fundamentally different kinds of video summaries: *static video storyboard summary*, which involves a set of keyframes extracted from the original video, and *dynamic video skimming*, which collects a set of shots by computing the similarity or relationship of each shot. One advantage of a video skim over a keyframe set is the ability to include audio and motion elements that potentially enhance both the expressiveness and information of the summary. In addition, it is often more entertaining and interesting to watch a skim than a slide show of keyframes. On the other hand, since they are not restricted by any timing or synchronization issues, once keyframes are extracted, there are further possibilities of organizing them for browsing and navigation purposes, rather than the strict sequential display of video skims, as demonstrated in [2, 9, 21, 26, 30]. In this paper, we focus on summarization techniques that produce a collection of static video frames.

Different approaches have been proposed in literature to address the problem of summarizing a video, most of them based on clustering techniques. The solutions are typically based on a two step approach: first identifying video shots from the video sequence, and then selecting keyframes according to some criterion from each video shot [12, 20]. A comprehensive review of past video summarization results can be found in [16, 28]. Some of the main ideas and results among the previously published results are briefly discussed next.

Zhuang et al. [33] proposed an unsupervised clustering method. A video sequence is segmented into video shots by clustering based on color histogram features in the HSV color space. For each video shot, the frame closest to the cluster centroid is chosen as the keyframe for the video shot. And only one frame per shot is selected into the video summary, regardless of the duration or activity of the video shot. As reported in [24], the approach does not guarantee an optimal result since the number of clusters is pre-defined by a density threshold value.

Hanjalic et al. [13] developed a similar approach by dividing the sequence into a number of clusters, and finding the optimal clustering by cluster-validity analysis. Each cluster is then represented in the video summary by a

1530-1834/08 $25.00 © 2008 IEEE
DOI 10.1109/SIBGRAPI.2008.31

IEEE computer society

keyframe.

Gong and Liu [10] proposed a video summarization method to produce a motion video summary that minimizes the visual content redundancy. To create the video summary, the original video sequence is structured into a shot cluster set where any pairs of clusters must be visually different, and all the shots belonging to the same cluster must be visually similar.

Chang and Chen [3] divided the video summarization task into three steps. Firstly, the shot detection process adopts the color and edge information to make the shot boundaries more accurately. Then the clustering process classifies the shots according to their similarity of motion type and scene. Finally, step three selects the important shots of each cluster in the skimming process by adopting shot-importance filter, which determines the importance of each shot by computing the motion energy and color variation.

In general such techniques require two passes and are rather computationally complex [19]. Moreover, earlier approaches based on shot detection return a fixed or variable number of frames per shot. This shot based approach may still contain redundancies because similar content may exist in several shots. For example, in news videos, the anchor person will appear many times in several video shots and those same frames may appear repeatedly in the summary. In contrast, we work on the video frames directly and cluster the frames that have similar content. To reduce the number of frames that will be used in the clustering algorithm, we pre-sample the video frames. As we show later in our experiments, the quality of the summaries is not affected by pre-sampling.

In this paper, we propose a simple and efficient approach for video summarization and a method to quantitatively evaluate the video summaries quality. The approach was applied to a sample of 20 videos selected from the Open Video Project [1]. The obtained results from the users on summaries indicate that the approach proposed is an alternative way to the video summarization problem.

The paper is organized as follows. The video summarization and evaluation method are described in Section 2. The experimental results are discussed in Section 3. Finally, some concluding remarks are derived in Section 4.

2. VSUMM Approach

The approach proposed includes two essential tasks to the video summarization process: *Video Summarization* and *Quantitative Evaluation*. The video summaries are firstly produced in a simple and efficient way. Second, a method to obtain quantitative measures of the quality of summaries is defined, which can even provide a benchmark for evaluating the video summaries.

2.1. Video Summarization

The VSUMM approach for video summarization was designed to be simple and efficient. To implement these characteristics, only color attributes were computed. These attributes are used to identify the similarity among the video frames. Thus, to extract the video frames attributes some simple tools for image analysis as color histogram and line profiles (horizontal, vertical and diagonal) were applied.

Color histograms and line profiles are computationally trivial to compute. In addition, color histogram is also robust to small changes of the camera position and to camera partial occlusion. Line profiles represent the color values of a single line of an image. However, one line is not sufficient to identify the similarity among the video frames. Due to this, VSUMM analyzes more than one line of a video frame. The number of line profiles analyzed is associated with the parameter *interval among line profiles*. For example, if the interval is 10, the line profiles will be analyzed at 10 by 10 lines.

In VSUMM approach the input video is not split into shots. Frame clusters are obtained by video frame analysis, independently of the shot which a frame belongs to. As showed in our experiments the proposed method for video summarization, VSUMM, demonstrated to be a good approach to produce static video summaries in an efficient and rapid way.

After extracting the visual features, VSUMM groups together similar frames and selects the most representative frame per each group, so produces the video summary. This is done with k-means clustering algorithm [22]. It is a staple of clustering methods, because the algorithm is very simple and works well in practice [7].

Figure 1 illustrates our proposed method for video summarization. First, the original video must be segmented into frames, step 1. Then, in step 2, some visual features are extracted from each frame to describe its visual content. We did not consider all the video frames, only a sample of frames. The algorithm used to extract these features should be fast and efficient in the visual content discrimination. We understand that a good approach to discriminate the visual content should be able to identify similar content in different frames using the same description and non similar content with different descriptions. Next, the frames are grouped by an unsupervised clustering method, step 3. The extracted visual features are used to identify frames with similar content. Generally, unsupervised cluster approaches need to know previously the number of clusters that will be generated. In our model, this value indicates the maximal number of keyframes that will be presented in the final results. In step 4, one frame (keyframe) per cluster is selected. The criterion to select a frame is based on its high representativeness. This permits to identify a frame with capabil-

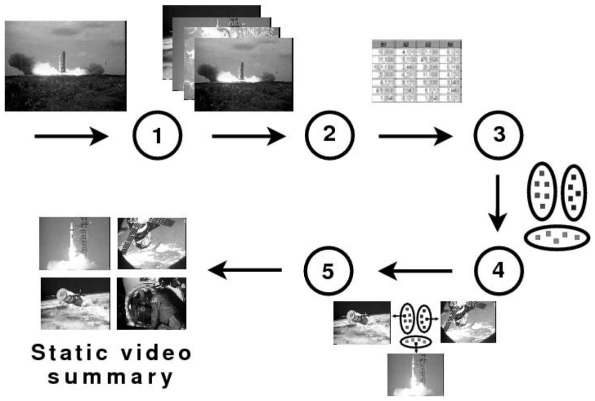


Figure 1. Video summarization architecture.

ity to represent the visual content of all others in its cluster. To create the static video summary, in step 5, keyframes are filtered to eliminate keyframes that are too similar. In some cases, different keyframes with very much similar visual content can be selected. Applying this filtering operation we reduce the number of keyframes maintaining the quality of results. After that, the keyframes are arranged in temporal order to facilitate the comprehension of result.

2.2. Quantitative Evaluation

In order to advance the field, the effectiveness and/or efficiency of a new solution to a particular problem needs to be evaluated, preferably against existing methods. However, a consistent evaluation framework is seriously missing in video abstraction research, resulting in the fact that every work has its own evaluation method, often lacking the performance comparison with existing techniques. This is partly because, unlike other research areas such as object detection and recognition, evaluating the correctness of a video summarization is not a straightforward task due to the lack of an objective ground-truth. It is even difficult for humans to decide if one video summary is better than another, and to make matters worse, the summarization viewpoint and perspective are often application-dependent. The existing evaluation methods for video summarization are grouped into three different categories [25]: result description, objective metrics and user studies.

Result description is the most popular and simple form of evaluation, as it does not involve any comparison with other techniques. This category is also used to discuss the influence of the system parameters or visual dynamics of the video sequence on the keyframe set extracted [13, 31, 32, 33]. Some works may attempt, in descriptive form, to explain and illustrate advantages of the proposed technique compared with some existing methods [15, 27].

In objective metrics, for keyframe extraction techniques, the metric is often the fidelity function computed from the extracted keyframe set and original frame sequence. The metric is used to compare the keyframe set generated by different techniques, or by one underlying technique, but with different parameter sets. However, there is also no experimental justification for whether the metric maps well to human judgement regarding the quality of a keyframe set.

User studies are employed for evaluating keyframe extraction techniques in [4, 6, 8, 17, 21, 29]. These studies involve independent users judging the quality of generated video summaries, and are probably the most useful and realistic form of evaluation (especially when keyframes are extracted for user-based interactive tasks such as content browsing and navigation). Nevertheless, yet not widely employed due to the difficulty in setting them up.

2.2.1. Our Proposal Although the issues of keyframe extraction and video summarization have been intensively investigated, there is not a standard or an optimal method to evaluate their performance. As video summarization assessment is a strong subjective task, it is difficult for any mechanical comparison or simulation methods to obtain accurate evaluations.

The evaluation proposed in this paper is characterized as indirect evaluation, in which each keyframe is evaluated separately, i.e., the relevance of each keyframe is measured independently from the other keyframes that compose the summary. This relevance is determined by the users' perception on a scale of 1 to 5 (1 = bad, 2 = poor, 3 = fair, 4 = good, 5 = excellent), as used in [4, 21]. However, the quality of a summary depends also on whether there is redundant information (e.g. two or more similar keyframes) or whether there is missing information (e.g. parts of the content is not represented with any keyframe). Due to this, to validate the applicability of our evaluation method, the users also evaluated each video summary (the keyframes set).

Figure 2 illustrates our proposed evaluation for video summarization. Different types of summaries from the same video are produced, step 1. In step 2, all keyframes are displayed together to the users. They must assign a score for each keyframe, according to the aforesaid scale. This value represents the users' perception about keyframe significance to identify the video content. Notice that the evaluation process works like a "black box", because the users are not aware of the mechanism used to produce the summaries. Then, in step 3, the mean score for each video summary is computed. This mean gives the quality level of summaries produced and it is calculated as follows:

$$score_M = \frac{\textit{sum of keyframe scores}}{\textit{number of keyframes}} \qquad (1)$$

Video Name	#Frames	Duration
video1 = anni002.mpg	2,494	1:23
video2 = anni008.mpg	2,775	1:32
video3 = NASAWF-AstronautsInSpace.mpg	3,269	1:49
video4 = NASATOAT-AerodynamicForces.mpg	3,302	1:50
video5 = NASAWF-FlyingAPlane.mpg	3,458	1:55
video6 = NASAXPG-ModelTesting.mpg	3,534	1:58
video7 = NASASF-OilCleanUp.mpg	3,537	1:58
video8 = NASAMOAT-SageIIAndPicassoCena.mpg	3,609	2:01
video9 = NASAGWTF-DragActivityPartOne.mpg	3,620	2:00
video10 = NASAMOAT-AerosolMeasurementAndRemoteSensing.mpg	3,630	2:01
video11 = anni004.mpg	3,895	2:10
video12 = anni003.mpg	4,267	2:22
video13 = NASAWF-SpaceSuits.mpg	4,273	2:22
video14 = NASAWF-TheRedPlanet.mpg	4,306	2:23
video15 = NASATOAT-WindTunnels.mpg	4,662	2:35
video16 = NASASF-ImmuneSystem.mpg	5,874	3:16
video17 = NASADT12-FlightPioneers.mpg	6,019	3:20
video18 = NASAAATC-HurricanesAndComputerSimulation.mpg	6,099	3:23
video19 = NASASF-MoonPhases.mpg	6,449	3:35
video20 = NASAFOF-ComputerSimulation.mpg	6,902	3:50
Total	85,974	44:23

Table 1. Videos used in the experiments.

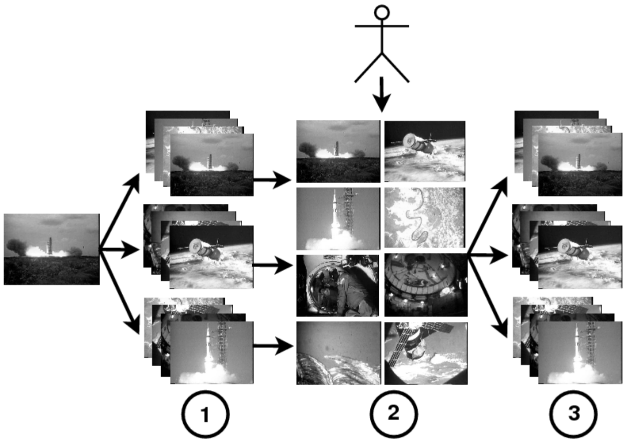


Figure 2. Evaluation architecture.

3. Experimental Results

The experiments were conducted on the Open Video Project files [1]. In order to analyze if the proposed algorithm works properly, a sample of 20 videos was chosen, all in MPEG-1 format (30 fps, 320 x 240 pixels), pertaining to news and documentaries. The total duration of test videos is about 45 minutes, and the duration of individual videos varies from 1 to 4 minutes. Some details of test videos are briefly listed in Table 1. The experiments were done using a Intel® Core$^{\text{TM}}$ Duo 1.83 GHz with 2GB RAM.

3.1. Pre-sample Analysis

A video sequence normally contains a large number of frames. In order to ensure that humans do not perceive any discontinuity in the video stream, a frame rate of at least 25 fps is required, that is, 90,000 images for one hour of video content [11]. It is intuitively obvious that for a frame rate of 25 fps, the 25 frames displayed for each second contain a lot of redundant information. Thus, instead of considering all the video frames, VSUMM takes only a subset of them (the so-called *pre-sampling* approach).

Pre-sampling is a technique largely used to reduce the clustering time (for instance, the mechanism proposed in [24] uses it) and is based on the idea that there are redundancies among the X (e.g, 25) frames per second of the input video. By using a sampling rate, the number of video frames to analyze can be reduced. Needless to say the sampling rate assumes a fundamental importance, as the larger this sampling rate is, the shorter is the clustering time, but the poorer results might be. For this reason, some preliminary tests were made with about eight videos randomly chosen from the 20 videos of the test set. Using different sampling rates, the time necessary for producing the summary was computed to know how many frames have to be analyzed in the video. The sampling rates were defined due to the video genres used in this work for summarization (news and documentaries), which present long scenes.

The average computational time was computed by analyzing all frames (none pre-sampling) and one frame out of 30, 45, 60, 75 and 90 frames (see Table 2). From these tests, it was observed that the quality of the produced summaries showed no significant differences in results, except for sample rate of one frame out of 90. To compare the performance among the results (one frame out of 75 with each other) we applied the student's t-test within 95% confidence interval [14], which proved that the alternatives were statistically different, except between the sample rates one frame out of 60 and 75 frames. As the second alternative was slightly better, the next experiments were executed for this alternative.

3.2. Attributes Analysis and User Study

To analyze the relative impact of the attributes described in Section 2.1 we designed a 2^k factorial experiment [14], with $k = 2$ or $k = 3$ whether attribute was histogram or line profiles, respectively. To keep the analysis simple, the factors that were known to affect the performance of the video summarization were kept fixed at two levels as follows:

1. Number of frame clusters: 15 or 35 clusters.
2. Number of histogram bins: 16 or 256 bins.
3. Interval among line profiles: 10 or 40 lines.

For these new experiments, the same eight videos used previously were taken. The results are shown in Table 3 and Table 4, where one can see that the 5th experiment configuration in Table 4 gives the best performance for video summarization. From these results, we found that, with 95% of confidence, the difference among histogram and line profiles is statistically significant. However, within a 95% confidence interval, the mean difference among line profiles were not statistically significant.

Videos	Computational Time (sec.)					
	All	30	45	60	75	90
video2	59.11	2.07	1.43	1.12	0.90	0.79
video8	85.76	2.92	1.43	1.62	1.26	1.09
video9	87.34	2.95	2.00	1.55	1.27	1.13
video11	83.86	2.93	1.98	1.53	1.20	1.08
video12	96.39	3.15	2.17	1.68	1.28	1.21
video17	162.08	4.71	3.47	2.51	2.04	1.72
video18	160.53	4.87	3.44	2.58	2.09	1.81
video20	172.85	5.43	3.80	3.01	2.26	2.00
mean	113.49	3.63	2.59	1.95	1.54	1.35

Table 2. The average computational time by analyzing all frames in the video and one frame out of 30, 45, 60, 75 and 90 frames.

A	B	Computational Time (sec.)
15	16	**1.50**
35	16	1.56
15	256	1.68
35	256	1.91

Table 3. The average computational time by analyzing the histogram. *A* stands for the number of frame clusters and *B* for the number of histogram bins.

A	B	C	Computational Time (sec.)		
			Horizontal	Vertical	Diagonal
15	16	10	0.87	0.99	1.10
35	16	10	1.26	1.27	1.29
15	256	10	0.81	0.95	1.12
35	256	10	0.96	1.13	1.45
15	16	40	**0.65**	0.90	0.82
35	16	40	0.67	1.05	1.05
15	256	40	0.86	**0.72**	**0.79**
35	256	40	1.11	0.78	0.90

Table 4. The average computational time by analyzing the line profiles. *A* stands for the number of frame clusters, *B* for the number of histogram bins and *C* for interval among line profiles.

As the quality of the video summaries is even more important than the time necessary to produce a video summary, we invited ten individuals to evaluate the quality of the summaries generated from best configuration histogram and each best configuration line profile. These ten evaluators include four graduate students and six undergraduate students (majoring in computer science). Before the tests, they could watch each video sample for as many times as needed till he/she grasped the theme of the sample. Then the evaluators watched summaries generated and answered the following question: what is the relevance of each image (keyframe) to the video summary according to the video content? The average scores computed from Equation 1 are shown in Table 5.

The results of users' evaluation also show that the horizontal line profile gives the best quality for video summarization. Thus, the video summaries were produced for 20 videos chosen (see Table 1) with 16 histogram bins, interval among profiles of 40 lines (i.e., 6 horizontal profiles) and different number of frame clusters (15, 20, 25, 30 and

Videos	Score			
	Histog.	Line Profile		
		Horiz.	Vert.	Diag.
video2	3.6	3.7	3.5	3.7
video8	2.9	3.4	3.1	2.8
video9	2.8	3.3	2.9	2.9
video11	3.3	3.7	3.5	3.3
video12	3.3	3.4	3.5	3.3
video17	3.7	3.4	3.3	3.1
video18	3.4	3.5	3.4	3.3
video20	3.3	3.3	3.2	3.1
mean	3.3	**3.5**	3.3	3.2

Table 5. Results of users' evaluation.

35 clusters). Results of applying the algorithm give for each cluster an average computational time (in seconds) of 0.59, 0.64, 0.63, 0.64 and 0.65, respectively.

We compared our computational times with the results reported in Li et al. [19]. They generated summaries for 15, 28 and 47 clusters, where the execution times were 3.91, 4.36 and 24.77 seconds, respectively. Notice that as the size of the clusters increases, the execution time in [19] increases much more. However, to draw comparisons among different approaches, experimental conditions should be respected.

In order to support the validity of the evaluation method proposed, the summaries produced were evaluated by the same ten users. The best quality level (25 clusters) has an average value of 4.3 and the worst (15 clusters) has 3.6.

After evaluating the summaries according to the proposed approach, the users evaluated each video summary separately (i.e., the keyframes set that represent the video summary) and we confirmed that the qualitative result achieved by the evaluation method describes approximately the measures of users' relevance.

We illustrated our proposed method for video summarization for all 20 test videos[1]. The video summary was created with the best configuration obtained in experiments (6 horizontal line profiles and 16 histogram bins). The number of frame clusters was fixed at 15 clusters. The proposed method is illustrated for each video. First, the 15 frame clusters are presented. Next, the keyframes are shown, where one keyframe per cluster is selected. Finally, the similar keyframes are filtered and the remaining keyframes are arranged in temporal order, to display the video summary.

3.3. Comparison with Open Video Storyboard

The Open Video storyboards are generated using the algorithm from [5] and some manual intervention to refine results so generated. The scheme for generating them is published in [23]. In this section, we provide a comparison[1] of our results with Open Video storyboards for all 20 videos described in Table 1. The length of the produced summary was set in order to match the Open Video summary (i.e., if the Open Video summary was of 5 keyframes, the length of the VSUMM summary was set to 5 keyframes, too). Thus, the maximal number of keyframes will be the same in the final results.

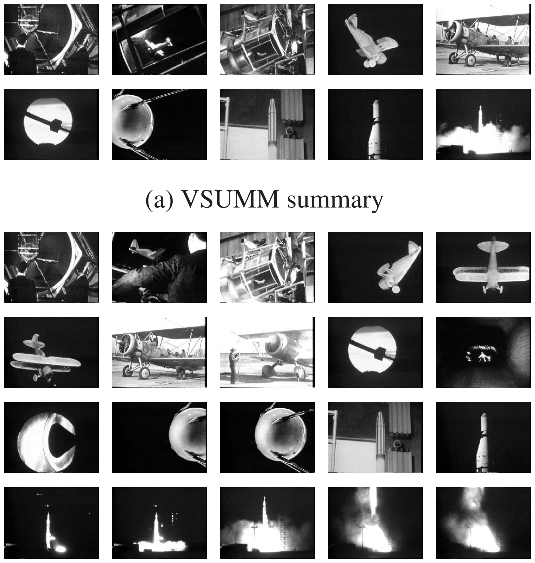
Table 6 shows the quality results and number of keyframes for each video summary. Based on the experimental results, we found out that for nine out of the 20 videos, the VSUMM summaries received a better quality score than the Open Video storyboards did; Five videos received identical summary scores; For six videos, our summaries received lower scores than the Open Video storyboards. We noticed that our highest score was equal to 4.4 against the Open Video best score that was equal to 4.0. Also, our worst results were equal to 3.3. Furthermore, we achieved five videos with scores greater than or equal to 4.0, while the Open Video obtained only one score equal to 4.0.

To illustrate this comparison, we present the results for three videos from the test videos. In Figure 3, the VSUMM summary presented a better quality score than the Open Video storyboard. On the other hand, in Figure 4, the Open Video storyboard showed a better quality score than VSUMM summary. And, in Figure 5, the VSUMM sum-

Videos	Score		#Keyframes	
	OV	VSUMM	OV	VSUMM
video1	4.0	4.4	20	10
video2	3.8	3.8	14	9
video3	3.5	3.5	18	10
video4	3.8	4.1	12	9
video5	3.9	3.5	7	7
video6	3.3	3.3	12	10
video7	3.0	3.6	12	8
video8	3.8	3.7	12	7
video9	3.4	3.3	6	6
video10	3.4	3.7	12	8
video11	3.8	3.8	29	15
video12	3.6	3.8	26	10
video13	3.7	4.0	8	6
video14	3.8	3.5	10	6
video15	3.7	3.6	12	10
video16	3.7	4.0	6	5
video17	3.7	4.0	19	9
video18	3.8	3.8	22	13
video19	3.3	3.6	13	8
video20	3.8	3.6	19	11

Table 6. Comparison of VSUMM summaries with Open Video storyboards.

[1] http://wavelet.dcc.ufmg.br/VSUMM



(a) VSUMM summary

(b) Open Video storyboard

Figure 3. VSUMM summary versus Open Video storyboard for *video1*.

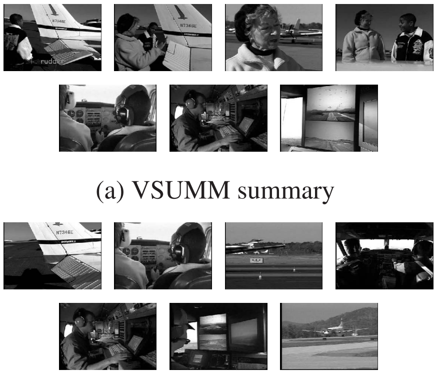
mary and Open Video storyboard exhibited identical scores.

Since no statement is given about the time needed to build the storyboards in the Open Video Project, as well as nothing is said about the running time of the method on which the project is based [5], we did not compare VSUMM computational time with Open Video.

4. Conclusion

In this paper, we presented VSUMM, a simple and efficient approach for video summarization. VSUMM used only color attributes to generate good quality summaries with low computational time. We also presented a method for the quantitative evaluation of the video summaries. The method proposed provided a measure to compare the quality of summaries of different techniques for video summarization. We compared our video summaries to Open Video storyboards. We showed that in most cases VSUMM achieved the best score, it can be said that the quality of the summary produced by VSUMM and Open Video is comparable.

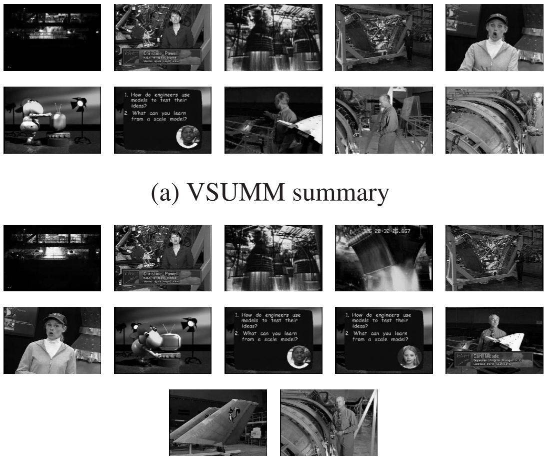
More tests of the evaluating method must be done to confirm its applicability into video summarization evaluation. But, at the moment, it is acceptable that this approach may be a viable alternative to compare the quality of video summaries created by different approaches. We also intend to test VSUMM on different genres of videos (cartoons, sports, tv-shows, talk-show). In addition, VSUMM can be easily used to generate video skims. For this, the video shots should be identified, according to selected keyframes in the static video summary.



(a) VSUMM summary

(b) Open Video storyboard

Figure 4. VSUMM summary versus Open Video storyboard for *video5*.



(a) VSUMM summary

(b) Open Video storyboard

Figure 5. VSUMM summary versus Open Video storyboard for *video6*.

5. Acknowledgments

The authors are grateful to CNPq and CAPES, Brazilian research funding agencies, for the financial support to this work.

References

[1] The Open Video Project. http://www.open-video.org.

[2] J. Ćalić, D. P. Gibson, and N. W. Campbell. Efficient layout of comic-like video summaries. *IEEE Trans. Circuits Syst. Video Techn.*, 17(7):931–936, 2007.

[3] I.-C. Chang and K.-Y. Chen. Content-selection based video summarization. *Int. Conf. on Consumer Electronics (ICCE)*, January 2007.

[4] S. E. F. de Avila, A. da Luz Jr., and A. de A. Araújo. VSUMM: A simple and efficient approach for automatic video summarization. In *Proc. of the Int. Conf. on Systems, Signals and Image Processing (IWSSIP)*, Bratislava, Slovak Republic, June 2008.

[5] D. DeMenthon, V. Kobla, and D. Doermann. Video summarization by curve simplification. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 211–218, NY, USA, 1998.

[6] M. S. Drew and J. Au. Video keyframe production by efficient clustering of compressed chromaticity signatures. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 365–367, New York, NY, USA, 2000.

[7] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*, chapter Unsupervised Learning and Clustering, page 654. Springer-Verlag New York, Inc., 2001.

[8] F. Dufaux. Key frame selection to represent a video. In *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, pages 275 –278. IEEE Computer Society, 2000.

[9] A. Girgensohn, J. S. Boreczky, and L. Wilcox. Keyframe-based user interfaces for digital video. *IEEE Computer*, 34(9):61–67, 2001.

[10] Y. Gong and X. Liu. Video summarization with minimal visual content redundancies. In *Proc. of the IEEE Int. Conf. on Image Processing (ICIP)*, pages 362–365, 2001.

[11] R. I. Hammoud. *Interactive Video Algorithms and Technologies*. Springer Berlin Heidelberg, 2006.

[12] A. Hanjalic. Shot-boundary detection: unraveled and resolved? *IEEE Trans. Circuits Syst. Video Technol.*, 12(2):90–105, 2002.

[13] A. Hanjalic, R. L. Lagendijk, and J. Biemond. *Image Databases and Multi-Media Search*, chapter A New Method for Key Frame based Video Content Representation, page 328. World Scientific, Singapore, January 1998.

[14] R. Jain. *The Art of Computer Systems Performance Analysis*. John Wiley and Sons, Inc., 1992.

[15] A. Joshi, S. Auephanwiriyakul, and R. Krishnapuram. On fuzzy clustering and content based access to networked video databases. In *Proc. of the IEEE Workshop on Research Issues in Database Engineering (RIDE)*, pages 42–43, Washington, DC, USA, 1998. IEEE Computer Society.

[16] Y. Li, S.-H. Lee, C.-H. Yeh, and C.-C. J. Kuo. Techniques for movie content analysis and skimming: tutorial and overview on video abstraction techniques. *IEEE Signal Processing Magazine*, 23(2):79–89, 2006.

[17] Y. Li, S. N. S., and C.-C. J. Kuo. *Video Mining*, chapter Movie content analysis, indexing and skimming via multimodal information, page 352. Springer, 2003.

[18] Y. Li, T. Zhang, and D. Tretter. An overview of video abstraction techniques. Technical report, HP Laboratory, HP-2001-191, July 2001.

[19] Z. Li, G. M. Schuster, and A. K. Katsaggelos. Minmax optimal video summarization. *IEEE Trans. Circuits Syst. Video Technol.*, 15(10):1245–1256, 2005.

[20] R. Lienhart. Reliable transition detection in videos: A survey and practitioner's guide. *Int. Journal of Image and Graphics (IJIG)*, 1(3):469–486, 2001.

[21] X. Liu, T. Mei, X.-S. Hua, B. Yang, and H.-Q. Zhou. Video collage. In *Proc. of the ACM Int. Conf. on Multimedia*, pages 461–462, New York, NY, USA, 2007.

[22] J. B. MacQueen. Some methods for classification and analysis of multivariate observations. In *Proc. of the Berkeley Symposium on Mathematical Statistics and Probability*, pages 281–297. University of California Press, 1967.

[23] G. Marchionini and G. Geisler. The Open Video Digital Library. *D-Lib Magazine*, 8(12), December 2002.

[24] P. Mundur, Y. Rao, and Y. Yesha. Keyframe-based video summarization using delaunay clustering. *Int. Journal on Digital Libraries*, 6(2):219–232, 2006.

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

[26] S. Uchihashi, J. Foote, A. Girgensohn, and J. S. Boreczky. Video manga: generating semantically meaningful video summaries. In *Proc. of the ACM Multimedia Conf. (ACMMM)*, pages 383–392, 1999.

[27] J. Vermaak, P. Pérez, M. Gangnet, and A. Blake. Rapid summarization and browsing of video sequences. In *Proc. of the British Machine Vision Conf. (BMVC)*. British Machine Vision Association, 2002.

[28] Z. Xiong, X. S. Zhou, Q. T., Y. Rui, and T. S. Huangm. Semantic retrieval of video – review of research on video retrieval in meetings, movies and broadcast news, and sports. *IEEE Signal Processing Magazine*, 23(2):18–27, 2006.

[29] I. Yahiaoui, B. Mérialdo, and B. Huet. Automatic video summarization. In *Multimedia Content-Based Indexing and Retrieval (MCBIR)*, 2001.

[30] M. M. Yeung and B.-L. Leo. Video visualization for compact representation and fast browsing of pictorial content. *IEEE Trans. Circuits Syst. Video Technol.*, 7(5):771–785, 1997.

[31] X.-D. Yu, L. Wang, Q. Tian, and P. Xue. Multi-level video representation with application to keyframe extraction. In *Proc. of the IEEE Int. Multimedia Modelling Conf. (MMM)*, pages 117–121, Washington, DC, USA, 2004. IEEE Computer Society.

[32] X.-D. Zhang, T.-Y. Liu, K.-T. Lo, and J. Feng. Dynamic selection and effective compression of key frames for video abstraction. *Patt. Recog. Letters*, 24(9–10):1523–1532, 2003.

[33] Y. Zhuang, R. Yong, T. S. Huang, and S. Mehrotra. Adaptive key frame extraction using unsupervised clustering. volume 1, pages 866–870. IEEE Computer Society, 1998.