A New Method for Static Video Summarization Using Local Descriptors and Video Temporal Segmentation

Edward J. Y. Cayllahua Cahuina
Computer Research Center
San Pablo Catholic University
Arequipa, Peru
Email: ecayllahua1@gmail.com

Guillermo Camara Chavez
Department of Computer Science
Federal university of Ouro Preto
Ouro Preto, Brazil
Email: gcamarac@gmail.com

***Abstract*—The continuous creation of digital video has caused an exponential growth of digital video content. To increase the usability of such large volume of videos, a lot of research has been made. Video summarization has been proposed to rapidly browse large video collections. To summarize any type of video, researchers have relied on visual features contained in frames. In order to extract these features, different techniques have used local or global descriptors. In this paper, we propose a method for static video summarization that can produce meaningful and informative video summaries. We perform an evaluation using over 100 videos in order to achieve a stronger position about the performance of local descriptors in semantic video summarization. Our experimental results show, with a confidence level of 99%, that our proposed method using local descriptors and temporal video segmentation produces better summaries than state of the art methods. We also demonstrate the importance of a more elaborate method for temporal video segmentation, improving the generation of summaries, achieving 10% improvement in accuracy. We also acknowledge a marginal importance of color information when using local descriptors to produce video summaries.**

***Keywords*-Local descriptors; Temporal segmentation;Video summarization**

I. Introduction

Due to the increased use of video and the human effort taken to process it, new technologies need to be researched in order to manage effectively and efficiently such an enormous quantity of information. Video summarization has recently been of interest for many researchers due to its importance in several applications such as information browsing and retrieval [1], [2]. A video summary is a short version of an entire video sequence and aims to give to a user a synthetic and useful visual abstract of a video sequence.

The most important goal is to provide users with a concise video representation so that the user have a quick idea about the content of the video [3]. Generally speaking, the task of video summarization has been approached by using different methods to cluster the video content and therefore, detect the redundancy of the video content in order to summarize it [4]. Figure 1 shows a generic approach for video summarization. The general steps involved are: video segmentation, then the feature extraction process is performed, afterwards a redundancy detection based on the features is applied and finally the video summary is generated.

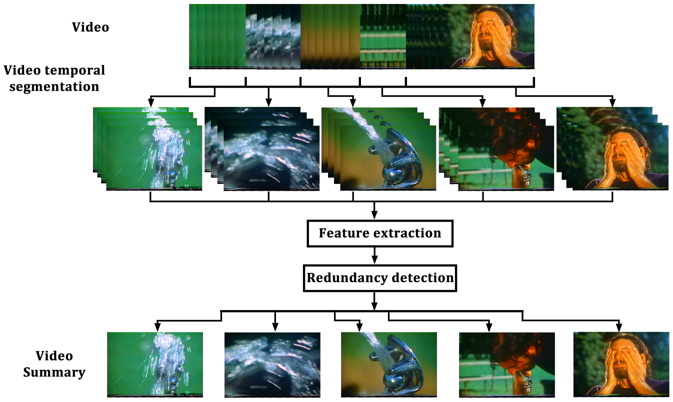

Fig. 1. A general approach for video summarization.

According to [5] and [6], the video summary can be represented into two fashions: a static video summary (storyboard) and a dynamic video skimming. Dynamic video skimming, consists in selecting the most relevant small dynamic portions (video skims) of audio and video in order to generate the video summary. On the other hand, a static video summary selects the most relevant frames (*keyframes*) of a video sequence [4]. A static summary is more appropriate for indexing, browsing and retrieval.

In order to summarize a generic video, most of the methods [7], [8], [9] have heavily relied on visual features computed from video frames. Visual features can be used to describe the global or local characteristics of an image. Many methods have based their analysis on global or local image descriptors. However, there has not been a wide evaluation about the performance of these two types of descriptors applied to video summarization. In [10], an evaluation was performed but their data set was limited to only 4 short videos, and therefore it is not possible to achieve strong conclusions about the performance of local descriptors for video summarization.

Another important consideration in video summarization is the temporal segmentation of the video. This task is usually performed by detecting transitions between shots and is often applied as the first step in video summarization. A shot is defined as an image sequence that presents continuous action which is captured from a single operation of a single camera. Shots are joined together in the editing stage of video production to form the final video, using different transitions. There are two different types of transitions that can occur between shots: abrupt shot transitions (cuts) or gradual transitions (dissolves, fade-in and fade-out).

A successful video temporal segmentation can lead into a better shot identification. Shots can be considered as the smallest indexing unit where no changes in scene content can be perceived and higher level concepts are often constructed by combining and analyzing the inter and intra shot relationships [11]. Therefore, we must outline the importance of an effective video temporal segmentation.

However, in video summarization, most of the methods [7], [12], [3] perform a simple video segmentation. This means that no visual effects, such as dissolves, are considered. Their approach is usually simple, as they only try to identify the cuts transitions in order to detect the video shots. Temporal video segmentation applied to video summarization has not been widely explored by most of the researchers. And consequently, its importance has not been evaluated.

A. Contributions

During the last years, several approaches for video summarization have been proposed. Most of these approaches base their analysis on local or global visual features computed by descriptors. Nonetheless, there has not been an evaluation about what type of descriptor is better for video summarization. The main contribution of this paper is to present a method for static video summarization using semantic information and video temporal segmentation. We also perform a wide evaluation in order to achieve a stronger position about the performance of local descriptors in a semantic video summarization. Furthermore, we evaluate the robustness of local descriptors compared to global descriptors.

Additionally, another important thing to outline is that some local descriptors use color information and others do not. So far there has not been any evaluation on whether local descriptors using color information give more meaningful summaries compared to other local descriptors. Therefore, we also inspect if color information can help local descriptors to produce better video summaries.

One essential operation in video summarization is temporal segmentation. However, most of the approaches use a simple temporal segmentation where only cuts transitions are detected. We propose a more elaborated video temporal segmentation to detect abrupt and gradual transitions (dissolves, fade-in and fade-out). Furthermore, we evaluate how local descriptors are affected by temporal segmentation and its importance in giving better video summaries.

II. Related work

We now present some of the existing methods for static video summarization. Since our work is more related to visual features, special attention to these methods will be taken.

A. Methods Based on Image Descriptors

Image descriptors are probably one of the most popular resource used in computer vision. One of the most common image descriptors used in video summarization are color histograms. Color histograms have often been used to measure the similarity between two frames, which is useful when the method's goal is to summarize the video based on redundancy elimination.

In [13], the color histogram is used as the main descriptor. The idea is to segment the video in shots and form groups using an unsupervised clustering algorithm. Every frame will belong to a certain cluster based on its color histogram. Then, the closest frame to each centroid is marked as a *keyframe* and extracted to build the storyboard. Recent research, still use the simplicity and power of color histograms. Such as the methods proposed by [3] and [7].

Color histograms are usually vectors of high dimensionality. In order to overcome this problem, several methods have proposed to apply mathematical procedures on these feature vectors in order to reduce their dimensionality. In [14], the singular value decomposition (SVD) is proposed. Later, in [15] and [16] the principal component analysis (PCA) is also used. Also, in [17], a static video summarization approach is presented using both color histograms and dimension reduction using PCA. Thus far, no comparisons have been performed between the two approaches. Furthermore, there has been no evaluation about the cost-benefit between the results obtained and the computational cost implied in performing these mathematical procedures.

In [8], not only color descriptors but other visual descriptors are used. They propose to use the Compact Composite Descriptors (CCDs), which consist of four descriptors: the Color and Edge Directivity Descriptor (CEDD) and the the Fuzzy Color and Texture Histogram (FCTH) proposed by [18], the Brightness and Texture Directionality Histogram (BTDH) descriptor [19]. They state that their method gives satisfactory results compared to four other methods, unfortunately they only use five videos to make the comparison.

III. Methods Based on Mid-level Semantics

A more informative summary can be obtained if the method considers the semantic meaning implied in the video. To summarize generic videos taking into account the semantic information, the methods have relied on object detection.

In [20], a static video summarization based on object recognition is proposed. The idea is to eliminate redundancy of information from the temporal and spatial domain and also from the content domain by doing object recognition. The shot boundaries are detected and video objects are extracted using a 3D graph-based algorithm. Then, a K-means [21] clustering algorithm is applied to detect the key objects.

In [9], a method based on concept preservation is proposed. They use the Bag of Words (BoW) model. They first segment the video into samples/shots, then for each shot the SIFT descriptor (Scale-invariant feature transform) [22] is used to extract the local features from detected *keypoints*. Later these features are clustered to produce a visual word dictionary. In addition, for each shot they produce a histogram of occurrences of visual words using a visual word dictionary. Then, the histograms are grouped, meaning that similar visual entities will be grouped together. Finally, the video summary is produced by extracting the frames that contain the important visual entities. In [23], the video content is analyzed using the SIFT features to produce a static summary. The idea is to summarize the video based on the content complexity and the difference between frames. In order to do this, a video segmentation is applied based on the content complexity. Once the video is segmented, the detected shots are merged based on their similarity. Finally, the *keyframes* are extracted from the detected merged shots. The SIFT descriptor has been widely used in computer vision for its ability to handle intensity, rotation and scale variations; this makes it a good descriptor but one disadvantage is its high computational cost. In order to overcome the computational cost, the SURF (Speeded Up Robust Features) [24] descriptor is used by [8]. Unfortunately, no evaluation has been performed between these descriptors applied to video summarization.

Temporal information has also been used for video summarization. In [25], a method that first extracts the *keyframes* considering the temporal information is proposed. Then, a Region of Interest (ROI) is estimated for the extracted frames. Finally an image is created by arranging the ROI's according to their time of appearance and their size. The result is a static summary of the video.

Probabilistic models can also be used. In [26], a method is proposed to extract *keyframes* by using a *maximum a posteriori* estimation and therefor produce the video summary. The method uses three probabilistic components: the prior of the *keyframes*, the conditional probability of shot boundaries and the conditional probability of each video frame. Then, the Gibbs sampling algorithm is applied for *keyframe* extraction and finally producing a static video summary.

IV. Technical background

The production of videos usually involves two important operations, which are: shooting and edition operations [11]. The shooting operation consist in the generation of the different shots that compose the video. The second operation involves the creation of a structured final video. To achieve this, different visual effects have been added to provide smooth transitions between the shots.

A. Visual Effects in Videos

The visual effects usually used on the videos consist of abrupt transitions (cuts) and gradual transitions (dissolves).

1) Cut: A cut is defined as a sharp transition, it is characterized by the abrupt change between consecutive shots.

2) Dissolve: The dissolve is characterized by a progressive change of a shot S_i into a shot S_{i+1} with non-null duration. This means that final frames of Shot S_i are combined with the first frames of Shot S_{i+1} to create the transition. Fade-in and fade-out transitions are special cases of dissolves, instead of combining two shots, a shot is combined with a monochrome frame. An example of a dissolve effect is shown in Figure 2

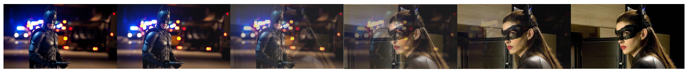

Fig. 2. A dissolve transition.

V. Proposed Method

In Figure 3, we present a general overview of our proposed method. Initially, the temporal segmentation procedure detects the shot boundaries. Then, each shot is clustered. This is done to detect frame samples from each shot. Later, for each of the frames previously detected, we apply the feature description procedure. Afterward, a Bag-of-Visual-Words approach is adopted. The detected local features are clustered to generate our Visual Word Vocabulary. Next, we compute the histograms of occurrences of visual words from each frame of the detected frame samples. The histograms of occurrence are clustered, the method finds the frames that are closer to each cluster's centroid. The frames that represent the centroids are considered as *keyframes*. The method filters the results to eliminate possible redundant *keyframes*. Finally, the *keyframes* are ordered in chronological order. The final result is the video summary.

A. Temporal Video Segmentation

In order to perform a temporal video segmentation, our method first computes the color histograms for each pair of adjacent frames in the video. Then, the cosine dissimilarity measure is computed between two consecutive histograms, the resulting information is a dissimilarity vector. The method detects abrupt changes in the dissimilarity vector when two adjacent frames have a high dissimilarity. This implies that a cut transition has been detected. Empirically, we found that using a threshold th of value 0.4 satisfactorily detects the abrupt changes.

In Figure 4, a dissimilarity vector is shown for a video. As we can see, using $th = 0.4$ will detect several false shot cuts. In order to overcome this problem, a filter procedure is executed to refine the dissimilarity vector.

The procedure works as follows: each of the values in the dissimilarity vector is analyzed. Each value becomes a pivot when evaluating the dissimilarity vector. Then, a neighborhood of values around the pivot i is taken, excluding the value of

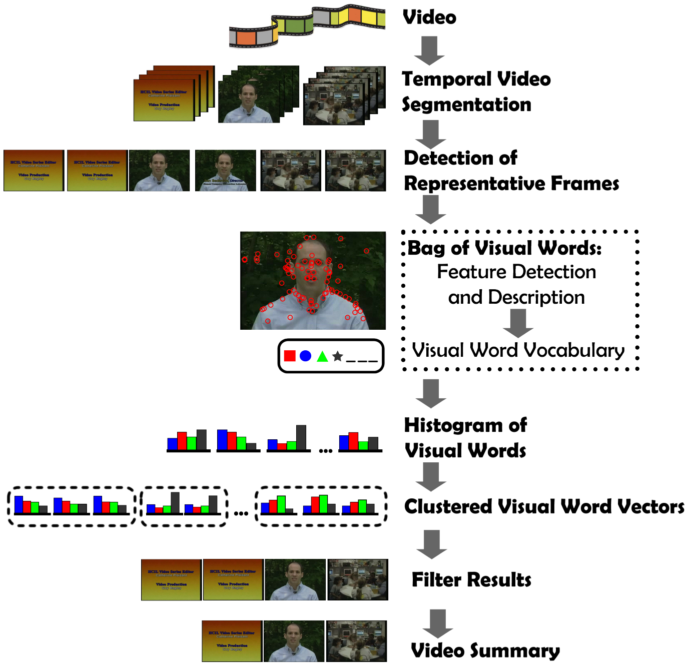
Video
Temporal Video Segmentation
Detection of Representative Frames
Bag of Visual Words:
Feature Detection and Description
Visual Word Vocabulary
■●▲★___
Histogram of Visual Words
Clustered Visual Word Vectors
Filter Results
Video Summary

Fig. 3. Proposed method for summarizing videos.

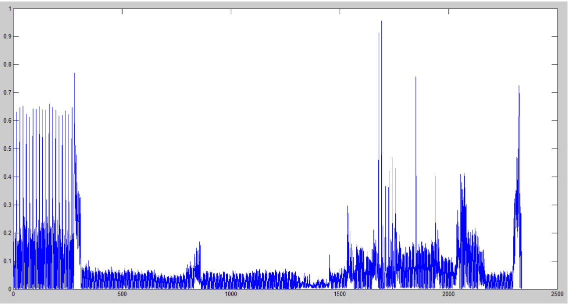

Fig. 4. A dissimilarity vector computed from video *HCIL Symposium 2002 - Introduction, segment 01*.

the pivot. The maximum value mv_i of the neighborhood is calculated. If the value of the pivot v_i is greater than mv_i, the value of the pivot is modified by applying Equation 1. This is done to give importance to the high values that are not around noisy areas.

$$r_i = \begin{cases} \frac{v_i - mv_i}{v_i} & \text{if } v_i > mv_i \\ v_i & \text{otherwise} \end{cases} \quad (1)$$

where i is a position in the dissimilarity vector, r_i is the computed value, v_i is the value of the pivot and mv_i is the maximum value of the neighborhood around the pivot.

After applying Equation 1 the values in the dissimilarity vector are refined. Figure 5 shows the refined dissimilarity vector for a video, the marked rounds in the peaks are the recognized abrupt changes. To know the number of segments present on the video, the method simply counts the number of abrupt changes.

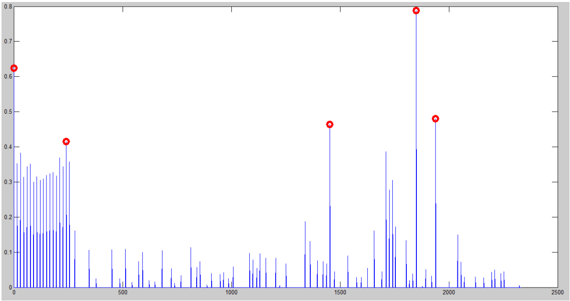

Fig. 5. A refined dissimilarity vector computed from video *HCIL Symposium 2002 - Introduction, segment 01*.

This type of segmentation is very effective and not computationally expensive. However, there is one shortcoming. Videos usually have visual effects such as dissolves, as shown in Figure 2.

A dissolve effect can produce distorted images, which have a great impact in the discriminative power of a descriptor. A frame that is produced during the dissolve transition will produce a lot of spurious key points. These false key points will eventually affect the summarization process. To overcome this problem, the method identifies the portions of the video where this effect happens. Once these portions are detected, they are excluded of any posterior analysis.

To detect the possible dissolve effect in the video, our method first computes the variance in each frame of the video and put it in a vector. Figure 6 shows the resulting variance vector for a video. Dissolve effects are located in the valley areas in the variance vector. To detect the valleys areas in the variance vector we created a procedure based on [27], [11] to find the portion of video where a dissolve effect occurs. As a fade is a special case of a dissolve, we can explore some of the features used for dissolve detection. The existence of monochrome frames is a very good clue for detecting all potential fades. The variance of the frame is used for detecting the monochrome frames. Then, we use the descriptor that characterizes the dissolve effect.

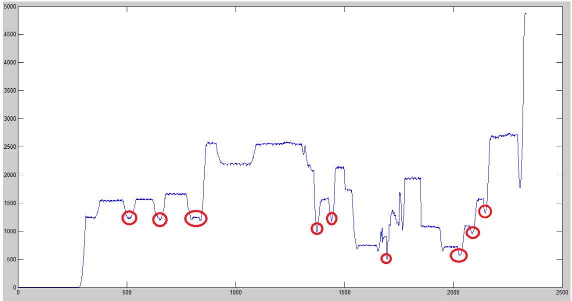

Fig. 6. Variance vector for video *HCIL Symposium 2002 - Introduction, segment 01*. The circles are the detected dissolve effects.

At this point, the method has detected the shot boundaries and therefore segmented the video. Furthermore, by detecting the gradual transitions the method can exclude non relevant portions of video that can actually affect the performance of the summarization method.

B. Detection of Representative Frames

Once the video has been segmented, the method uses the valid parts of the video for analysis. Our method applies the X-means [28] algorithm for each portion of the previously segmented video using the color histograms information. This is done to detect the most representative frames for each portion of the analyzed video. All the posterior analysis will be performed using these frames.

C. Bag of Visual Words

Our method uses the Bag-of-Words (BoW) approach to summarize videos. To adopt the approach, an image can be considered as a document.

The "words" are the visual entities found in the image. They will describe the object and therefore represent the semantic entities. Using their information we can perform a semantic summarization based on the objects of the video.

The Bag-of-Visual-Words approach consists of three operations: feature detection, feature description and visual word vocabulary generation. A visual word vocabulary is generated from the feature vectors obtained during the feature detection process, each visual word (codeword) represents a group of several similar features. The visual word vocabulary defines a space of all entities occurring in the video, it can be used to semantically summarize the video based on the entities present on it.

D. Histogram of Visual Words

A histogram of visual words is created by counting the occurrence of the visual words. For each representative frame, the local image features are used to find the visual words that occur in the image. These occurrences are counted and arranged in a vector. Consequently, each representative frame will have an associated vector of visual word occurrences (visual word vector).

E. Visual Word Vectors Clustering

Finally, the method uses all the visual word vectors recently obtained and applies the X-means algorithm. Frames with similar visual entities are grouped together. Then, for each cluster, the nearest frame to the centroid is chosen as the *keyframe*. All the detected *keyframes* are ordered according to their time of appearance and they represent the video summary or storyboard. By doing it, we have grouped together the most representative frames of the video taking into consideration the semantic information (visual entities) contained in them. This ensures that the final video summary contains the most important visual entities present in the video.

F. Filter Results

This final operation tries to eliminate possible duplicated *keyframes*. A pairwise Manhattan distance is calculated from color histograms of consecutive *keyframes*. The method uses a threshold of value 0.5 to define a high similarity between two color histograms, this value was also used and discussed by [7].

VI. Experiments

In this paper, a total of 100 videos were used. The data set consists of: 50 videos from the Open Video data set [29] and another 50 videos from websites like Youtube. The videos belong to different genres such as cartoons, news, sports, commercials, tv-shows, home videos, etc. The whole data set can be downloaded in [30].

In order to discover the values of the parameters, we have used 10 videos from the O.V. data set. These values were used for the experiments executed using the O.V. data set and the Youtube data set. We have discovered and used the following parameters for our experiments: The X-means algorithm is used two times. The first time, it is used to extract the representative frames for each part of our segmented video. The X-means has two important parameters: the initial number of clusters K_{min} and the maximum number of possible clusters K_{max}. During our experiments we discovered that using $K_{min} = 5$ and $K_{max} = 10$, we obtained the best results. This means that each segment will have at least 5 and at most 10 representative frames. According to our experiments, no shot produced more than 10 representative frames. The second time the X-means algorithm is used, is to generate the video summary by grouping the similar visual entities. To perform this, we have set $K_{min} = ns$, where the number of detected shots are defined by (ns) and $K_{max} = 3 \times ns$.

To adopt the Bag-of-Visual-Words approach we need to set the size of the "codebook" (number of words). The number of visual words used for the experiments is 400. In order to obtain this value, we have performed experiments with 12 videos from the O.V. data set. The best results were obtained using 400 visual words.

We have used several local descriptors for our experiments, such as SIFT: this paper uses the implementation of [31], the non-edge selection threshold value is set to 30. The peak selection threshold is set to 2.5. The number of levels per octave of the DoG scale space is set to 2. SURF: we use the implementation of [32]. Color descriptors such as: CSIFT, HUESIFT. We use the implementation of [33]. Finally, the HOG descriptor is also applied using the implementation of [34]. For all these descriptors, the default parameters were used.

A. Video Summary Evaluation

The evaluation method used was the Comparison of User Summaries (CUS) proposed by [7]. Such method allowed us to reduce the subjectivity of the evaluation task and we can also compare our results to other approaches.

CUS makes a comparison between the user summary and the automatic summary. The idea is to take a *keyframe* from the user summary and a *keyframe* from the automatic video summary. Then, they are both converted to the HSV color space. Afterwards, a 16 bins color histogram using only the Hue component is computed for both images. A similarity distance is calculated between the two histograms. If the result is superior than a predetermined threshold d (high similarity), then both are considered as matched frames. If both frames are matched, then both are removed from future iterations. The threshold value used is $d = 0.5$, it was the same threshold used by [7]. Then, the relevance of the summary is measured by the metric CUS_A.

CUS_A measures the accuracy of the summary, defined in Equation 2.

$$CUS_A = \frac{n_{mAS}}{n_{US}} \quad (2)$$

where n_{mAS} is the number of matching *keyframes* from the automatic summary and n_{US} is the number of *keyframes* from user summary.

CUS_A values vary in a range of 0 to 1. The lowest value (zero), means that no *keyframes* between the automatic summary and the user summary were matched, while the highest value one, means that all the *keyframes* were matched (best case scenario).

B. Experiments

All the tables show the mean accuracy rate for CUS_A. Furthermore, to measure the similarity between summaries we have used 2 different distances: Manhattan and cosine similarity.

The following local descriptors have been used: HoG[35], SURF[24], SIFT[22], CSIFT[36] and HUESIFT[37].

We have compared our results to other methods that use global descriptors, such as: VSUMM [7], DT [15], STIMO [3] and the summaries provided by the Open Video Data Set.

C. Results for the Open Video Data Set

In Table I, we show the performance of the visual words using local descriptors and temporal segmentation for the OV dataset. Table II shows the results obtained with visual words using local descriptors and without using temporal segmentation, and we also show the results obtained using global descriptors.

As we can see in Table I, local descriptors had a better performance compared to the models using global descriptors shown in Table II. This means that our semantic analysis considering the visual words described by local descriptors produced video summaries more in accordance to the summaries expected by the users. Between the local descriptors, HUESIFT performed better than the traditional SIFT and got the best results. Meaning that the color information was useful, but the improvement is not significant. The rest of the local descriptors had a similar performance showing the robustness of local descriptors in video summarization.

In Table II, the video segmentation proposed by [7] is used in our method, instead of our proposed temporal segmentation. As we can see, the visual words using local descriptors decreased their performance. This shows that local descriptors can be sensitive to visual video effects, such as dissolves. We can see that VSUMM performed better, but not by much, compared to local descriptors using the Manhattan distance. But when we observe the results obtained with the other distances, there is no significant difference between them. The reason of this, is that VSUMM was proposed using the Manhattan distance. Using other distances, VSUMM obtained better results, but the differences between the performance of VSUMM and the rest of local descriptors decreased to the point of being fairly similar. Another thing that we can observe is that, taking aside VSUMM, the local descriptors had better results than the rest of global descriptors. We must also note that despite decreasing their performance compared to the results with temporal segmentation, the visual words using local descriptors still produce relevant summaries and even better than most of the global descriptors used in our experiments, excluding VSUMM. This helps us to corroborate the importance of temporal segmentation in video summarization when using local descriptors. Additionally, it is noted that HoG and CSIFT produced the best results among the local descriptors without video temporal segmentation.

TABLE I
(Open Video Data Set) Experiments using temporal segmentation

Descriptor	Manhattan	Cosine
	CUS_A	CUS_A
HueSIFT	**0.967**	**0.987**
CSIFT	0.959	0.981
HoG	0.956	0.985
SURF	0.955	0.982
SIFT	0.954	0.985

TABLE II
(Open Video Data Set) Experiments without temporal segmentation

Descriptor	Manhattan	Cosine
Local	CUS_A	CUS_A
HueSIFT	0.829	0.896
CSIFT	0.858	0.919
HoG	0.869	0.925
SURF	0.808	0.883
SIFT	0.829	0.909
Global	CUS_A	CUS_A
DT	0.635	0.706
STIMO	0.827	0.886
OV	0.795	0.831
VSUMM	**0.901**	**0.940**

D. Results for the Youtube Database

Using the Youtube database, we observe that the evaluation values decreased for both global and local descriptors, although they have the same tendency of the results obtained using the Open Video data set. The reason is that the database has another type of videos, such as sports, games and news. In this type of videos, users are more interested in the events rather than the objects. Consequently, a domain-specific summarization is more ideal. Nonetheless, according to Table III, local descriptors still got better results compared to global descriptor VSUMM shown in Table IV. Consequently, the

TABLE III
(Youtube Data Set) Experiments using temporal segmentation

Descriptor	Manhattan	Cosine
Local	CUS_A	CUS_A
HueSIFT	**0.870**	0.949
CSIFT	0.865	0.932
HoG	0.865	0.943
SURF	0.869	0.948
SIFT	0.859	**0.956**

TABLE IV
(Youtube Data Set) Experiments without temporal segmentation

Descriptor	Manhattan	Cosine
Local	CUS_A	CUS_A
HueSIFT	0.749	0.865
CSIFT	0.735	0.863
HoG	0.726	0.860
SURF	0.725	0.852
SIFT	0.746	**0.875**
Global	CUS_A	CUS_A
VSUMM	**0.759**	0.868

TABLE V
Difference between mean accuracy rates CUS_A at a confidence of 99%, using the O.V. data set.

	Confidence Interval 99%	
Difference	Min	Max
HueSIFT - VSUMM1	0.031	0.062
HueSIFT - DT	0.243	0.319
HueSIFT - VISTO	0.077	0.125
HueSIFT - OV	0.123	0.188
CSIFT - VSUMM1	0.025	0.056
CSIFT - DT	0.238	0.313
CSIFT - VISTO	0.069	0.121
CSIFT - OV	0.115	0.183
HoG - VSUMM1	0.029	0.060
HoG - DT	0.241	0.318
HoG - visto	0.073	0.125
HoG - OV	0.120	0.187
SURF - VSUMM1	0.025	0.058
SURF - DT	0.238	0.313
SURF - VISTO	0.068	0.123
SURF - OV	0.116	0.185
SIFT - VSUMM1	0.031	0.059
SIFT - DT	0.242	0.318
SIFT - VISTO	0.074	0.125
SIFT - OV	0.121	0.187

overall better performance of local descriptors shows that the proposed semantic analysis and temporal segmentation helps to produce more meaningful summaries.

In Table IV, it is also noted that the local descriptors decreased their performance without temporal segmentation, this behavior was also noted in Table II. Compared to VSUMM (Global descriptor), local descriptors still produced good summaries. We can conclude from the two tables that local descriptor's performance is greatly benefited from temporal segmentation. And also, even without temporal segmentation, local descriptors still provide promising summaries.

In order to compare every pair of approaches and to validate the statistical significance of our results, the confidence intervals have been computed for the differences between paired means. If the confidence interval includes zero, the difference between the two methods is not significant at that confidence level. If the confidence interval does not include zero, then the sign of the mean difference indicates which alternative is better [38]. The positive sign means that our new method is better, negative sign means otherwise.

Table V shows the results of such comparisons between the different local descriptors and the other considered approaches. Since the confidence intervals with a confidence of 99% do not include zero in any case and we are also positive signs, the results presented in Table V confirms that our approach provides results with superior quality (highest accuracy rate) relative to the approaches to which it was compared. Furthermore, it is possible to say that our summaries are closer to the summaries created by users.

E. Final Results Analysis

According to our experiments, the color information used by the HUESIFT descriptor got the best results in the Open Video data set and also in the Youtube data set using the Manhattan distance. Nonetheless, this performance is not drastically superior to others local descriptors that do not use color information, such as SIFT, HoG or SURF. Due to the mixed good and bad results of HUESIFT and CSIFT, it is inconclusive whether color leads to more relevant video summaries.

We must outline that the SURF and HoG descriptors also produced promising results. Because the SURF and HoG descriptors are not as computationally expensive to compute as the SIFT, CSIFT and HUESIFT, they are a good option for a faster video summarization considering the semantic information.

VII. Conclusion

In this paper, we approached the task of video summarization by considering the semantic information expressed by the video's visual entities. The proposed method elaborates static video summaries and our core approach is to use temporal video segmentation and visual words obtained by local descriptors. The proposed method has taken advantage of previous techniques in video summarization and segmentation. We show how this approach leads to a successful summarization of generic videos.

Additionally, we evaluate the importance of local descriptors and temporal segmentation in automatic video summarization. We compare our results to other models that use global descriptors and simple video segmentation. According to our experiments, the color information used by some local descriptors did not lead into a greater performance compared to other local descriptors that do not use color information. In addition, video summaries created with our semantic analysis were the most similar to the user's summaries (ground-truth).

Nonetheless, they decrease their performance when no temporal segmentation is used.

VIII. ACKNOWLEDGMENT

The authors are thankful to CNPq, CAPES and FAPEMIG, Brazilian funding agencies and the San Pablo Catholic University for the support to this work.

REFERENCES

[1] Z. Xiong, X. S. Zhou, Q. Tian, Y. Rui, and H. TS, "Semantic retrieval of video - review of research on video retrieval in meetings, movies and broadcast news, and sports," *IEEE Signal Processing Magazine*, vol. 23, no. 2, pp. 18–27, march 2006.

[2] V. Valdes and J. Martinez, "Efficient video summarization and retrieval tools," in *2011 9th International Workshop on Content-Based Multimedia Indexing (CBMI)*, june 2011, pp. 43–48.

[3] M. Furini, F. Geraci, M. Montangero, M. Pellegrini, a1, a2, and a3, "Stimo: Still and moving video storyboard for the web scenario," *Multimedia Tools Appl.*, vol. 46, no. 1, pp. 47–69, Jan. 2010.

[4] Y. Gao, W.-B. Wang, J.-H. Yong, and H.-J. Gu, "Dynamic video summarization using two-level redundancy detection," *Multimedia Tools and Applications*, pp. 233–250, 2009.

[5] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Transactions on Multimedia Computing, Communications and Applications*, 2007.

[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*, pp. 121–143, 2008.

[7] S. E. F. de Avila, A. P. B. ao Lopes, J. A. da Luz, and A. de Albuquerque Araújo, "Vsumm: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, Jan. 2011.

[8] D. P. Papadopoulos, S. A. Chatzichristofis, and N. Papamarkos, "Video summarization using a self-growing and self-organized neural gas network," in *Proceedings of the 5th international conference on Computer vision/computer graphics collaboration techniques*, ser. MIRAGE'11. Berlin, Heidelberg: Springer-Verlag, 2011, pp. 216–226.

[9] Z. Yuan, T. Lu, D. Wu, Y. Huang, and H. Yu, "Video summarization with semantic concept preservation," in *Proceedings of the 10th International Conference on Mobile and Ubiquitous Multimedia*, ser. MUM '11. New York, NY, USA: ACM, 2011, pp. 109–112.

[10] M. Kogler, M. del Fabro, M. Lux, K. Schoeffmann, and L. Boeszoermenyi, "Global vs. local feature in video summarization: Experimental results," in *Proceedings of the 10th International Workshop of the Multimedia Metadata Community on Semantic Multimedia Database Technologies in conjunction with the 4th International Conference on Semantic and Digital Media Technologies (SAMT 2009)*. Germany: http://ceur-ws.org, December 2009.

[11] G. Camara Chavez, F. Precioso, M. Cord, S. Phillip Foliguet, and A. de A. Araujo, "Shot boundary detection by a hierarchical supervised approach," in *14th International Workshop on Systems, Signals and Image Processing, 6th EURASIP Conference focused on Speech and Image Processing, Multimedia Communications and Services*, 2007, pp. 197–200.

[12] G. Guan, Z. Wang, K. Yu, S. Mei, M. He, and D. Feng, "Video summarization with global and local features," in *Proceedings of the 2012 IEEE International Conference on Multimedia and Expo Workshops*. Washington, DC, USA: IEEE Computer Society, 2012, pp. 570–575.

[13] Y. Zhuang, Y. Rui, T. Huang, and S. Mehrotra, "Adaptive key frame extraction using unsupervised clustering," in *1998 International Conference on Image Processing, 1998. ICIP 98. Proceedings*, vol. 1, oct 1998, pp. 866 –870.

[14] Y. Gong and X. Liu, "Video summarization and retrieval using singular value decomposition," *Multimedia Systems*, vol. 9, no. 2, pp. 157–168, Aug. 2003.

[15] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using delaunay clustering," *Int. J. Digit. Libr.*, vol. 6, no. 2, pp. 219–232, Apr. 2006.

[16] T. Wan and Z. Qin, "A new technique for summarizing video sequences through histogram evolution," in *International Conference on Signal Processing and Communications (SPCOM)*, 2010, pp. 1–5.

[17] E. C. Cahuina, G. C. Chavez, and D. Menotti, "A static video summarization approach with automatic shot detection using color histograms," in *Proceedings of the 2012 International conference on Image Processing, Computer vision, Pattern recognition IPCV 2012, Las Vegas, USA*, 2012, pp. 92–99.

[18] K. Zagoris, K. Ergina, and N. Papamarkos, "Accurate image retrieval based on compact composite descriptors and relevance feedback information," in *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 24, no. 2, 2010, pp. 207–244.

[19] S. A. Chatzichristofis and Y. S. Boutalis, "Content based radiology image retrieval using a fuzzy rule based scalable composite descriptor," *Multimedia Tools Application*, vol. 46, no. 2-3, pp. 493–519, Jan. 2010.

[20] Z. Tian, J. Xue, X. Lan, C. li, and N. Zheng, "Key object-based static video summarization," in *Proceeding of the 19th ACM International conference on Multimedia*, 2011, pp. 1301–1304.

[21] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. of the fifth Berkeley Symposium on Mathematical Statistics and Probability*, L. M. L. Cam and J. Neyman, Eds., vol. 1, 1967, pp. 281–297.

[22] D. Lowe, "Object recognition from local scale-invariant features," in *IEEE International Conference on Computer Vision*, vol. 2, 1999, pp. 1150–1157.

[23] J. Li, "Video shot segmentation and key frame extraction based on sift feature," in *2012 International Conference on Image Analysis and Signal Processing (IASP)*, 2012, pp. 1–8.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (surf)," *Computer Vision and Image Understanding —*, vol. 110, no. 3, pp. 346–359, 2008.

[25] Y. Lim, Y. Uh, and H. Byun, "Plot preservation approach for video summarization," in *2011 IEEE International Conference on Systems, Man, and Cybernetics (SMC)*, 2011, pp. 67–71.

[26] X. Liu, M. Song, L. Zhang, J. Bu, C. Chen, and D. Tao, "Joint shot boundary detection and key frame extraction," in *2012 21st International Conference con Pattern Recognition*, 2012, pp. 2565 – 2568.

[27] J.-U. Won, Y.-S. Chung, I.-S. Kim, J.-G. Choi, and K.-H. Park, "Correlation based video-dissolve detection," in *International Conference on Information Technology: Research and Education, 2003. Proceedings. ITRE2003.*, 2003, pp. 104–107.

[28] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML '00, 2000, pp. 727–734.

[29] Open Video Project, "Open Video Data Set," http://www.open-video.org, 2011, [Online; accessed on November 23, 2011].

[30] Complete Data Set, "Open Video and Youtube Data Set," https://sites.google.com/site/vsummsite/download, 2013, [Online; accessed on July 11, 2013].

[31] A. Vedaldi and B. Fulkerson, "VLFeat: An open and portable library of computer vision algorithms," http://www.vlfeat.org/, 2008.

[32] P. Strandmark, "SURFmex: A matlab surf interface," http://www.maths.lth.se/matematiklth/personal/petter/surfmex.php, 2010.

[33] K. V. de Sande, "ColorDescriptor: a color descriptor binary software," http://koen.me/research/colordescriptors/, 2010, [Online; accessed 20-February-2013].

[34] C. Xiankai, "smartlearning: Focus on machine learning and pattern recognition," http://code.google.com/p/smartlearning/source/browse/trunk/c%2B%2B/feature/hog/HOGDescriptor_/mexproc.cpp?r=29, 2012.

[35] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR05́)*, vol. 1, 2005, pp. 886–893.

[36] A. Abdel-Hakim and A. Farag, "Csift: A sift descriptor with color invariant characteristics," in *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, vol. 2, 2006, pp. 1978 – 1983.

[37] K. V. de Sande, T. Gevers, and C. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 9, pp. 1582–1596, Sep. 2010.

[38] R. Jain, *The Art Of Computer Systems Performance Analysis : Techniques For Experimental Design, Measurement, Simulation, And Modeling*. Wiley New York, 1992.