2013 12th International Conference on Machine Learning and Applications

Unsupervised Video Summarization via Dynamic Modeling-based Hierarchical Clustering

Karim M. Mahmoud*†, Nagia M. Ghanem*, and Mohamed A. Ismail*
* *Computer and Systems Engineering Department*
Faculty of Engineering, Alexandria University
Alexandria 21544, Egypt
† *IBM, Egypt Branch*
Email: kmahmoud@eg.ibm.com

***Abstract*—Mining the video data using unsupervised learning techniques can reveal important information regarding the internal visual content of large video databases. One of these information is the video summary which is a sequence of still pictures that represent the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original video is preserved. In this paper, an enhanced method for generating static video summaries is presented. This method utilizes a modified dynamic modeling-based hierarchical clustering algorithm that depends on the temporal order and sequential nature of the video to fasten the clustering process. Video summaries generated by our method are compared with summaries generated by others found in the literature and the ground truth summaries. Experimental results indicate that the video summaries generated by the proposed method have a higher quality than others.**

***Keywords*-Video Summarization, Key Frames Extraction, Dynamic Modeling, Clustering.**

I. Introduction

A tremendous amount of multimedia information including digital video is becoming prevailing as a result of the advances in multimedia computing technologies and high-speed networks. These advances and revolution in multimedia present new challenges for accessing and representing large visual collections that aims at improving the effectiveness and efficiency of video acquisition, archiving, and indexing as well as increasing the usability of stored videos. As a result, many research techniques have been proposed, including video summarization to generate important key frames regarding the internal visual content of large video databases.

Video summary is a set of static video key frames and it is defined as a sequence of still pictures that represent the content of a video in such a way that the respective target group is rapidly provided with concise information about the content, while the essential message of the original video is preserved [1].

Over the past years, various approaches and techniques have been proposed towards the summarization of video content. Many of these methods utilize unsupervised learning and clustering techniques including hierarchical clustering which do not require the number of clusters as an input, for example in [2]. However, these clustering algorithms have many drawbacks as they use a static model of the clusters and do not use information about the nature of individual clusters as they are merged. Moreover, some algorithms ignore the information about the aggregate interconnectivity of items in two clusters; while others ignore the information about the closeness of two clusters as defined by the similarity of the closest items across two clusters. By only considering either interconnectivity or closeness, these algorithms can easily select and merge the wrong pair of clusters [3].

Unlike other clustering algorithms, Chameleon [3] is an agglomerative hierarchical clustering that uses a dynamic modeling framework which overcomes the limitations of the other clustering algorithms. The key feature of Chameleon algorithm is that it utilizes both interconnectivity and closeness to identify the most similar pair of clusters [3].

In this paper, we present an enhanced method for generating static video summaries utilizing a modified agglomerative hierarchical clustering algorithm. This algorithm is based on the dynamic modeling framework used in Chameleon clustering algorithm [3]. The main modification proposed to Chameleon clustering algorithm is utilizing the temporal and sequential structure of the video stream in order to partition the video frames and to fasten the clustering process as it will be shown later in this paper.

The rest of this paper is organized as follows. Section 2 introduces some related work. Section 3 presents the proposed video summarization method used to extract video key frames and shows how to apply it to summarize a video sequence. Section 4 illustrates the evaluation method and reports the results of our experiments. Finally, we offer our conclusions and directions for future work in Section 5.

II. Related Work

In this section, some of the main recent video summarization methods which can be found in the literature are briefly discussed.

978-0-7695-5144-9/13 $31.00 © 2013 IEEE
DOI 10.1109/ICMLA.2013.140

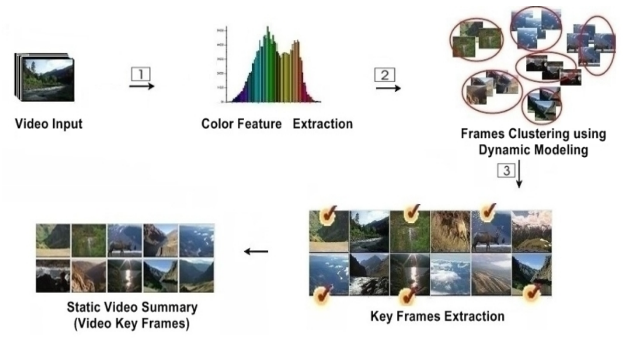
Video Input
1
Color Feature Extraction
2
Frames Clustering using Dynamic Modeling
3
Key Frames Extraction
Static Video Summary (Video Key Frames)

Figure 1. Proposed Video Summarization Method

In [4], a video summarization method called STIMO (STIll and MOving Video Storyboard) is introduced. This method is designed to produce on-the-fly video storyboards and it is composed of three phases. In the first phase, the video frames are pre-sampled and then feature vectors are extracted from the selected video frames by computing a color histogram in the HSV color space. In the second phase, a clustering method based on the Furthest-Point-First (FPF) algorithm is applied. To estimate the number of clusters, the pairwise distance of consecutive frames is computed using Generalized Jaccard Distance (GJD). Finally, a post-processing step is performed for removing noise video frames.

In [5], a method called VSUMM (Video SUMMarization) is presented. In the first step, the video frames are pre-sampled by selecting one frame per second. In the second step, the color features of video frames are extracted from Hue component only in the HSV color space. In the third step, the meaningless frames are eliminated. In the fourth step, the frames are clustered using k-means algorithm where the number of clusters is estimated by computing the pairwise Euclidean distances between video frames and a key frame is extracted from each cluster. Finally, another extra step occurs in which the key frames are compared among themselves through color histogram to eliminate that similar key frames in the produced summaries.

In [6], a video summarization method based on clustering the video frames using the Delaunay Triangulation (DT) is developed. The first step is pre-sampling the frames of the input video. Then, the video frames are represented by a color histogram in the HSV color space and the Principal Component Analysis (PCA) is applied on the color feature matrix. After that, the Delaunay diagram is built and clusters are formed by separating edges in the Delaunay diagram. Finally, a frame is selected from each cluster.

III. Proposed Video Summarization Method

Figure 1 shows the steps of the proposed video summarization method. First, the color features of video frames are extracted (Step1). Second, the modified dynamic modeling-based hierarchical clustering algorithm is applied (Step 2). Then, in step 3, one frame per cluster is selected as a key frame. Finally, the extracted key frames are arranged in the original order of appearance in the video to facilitate the visual understanding of the result. These steps are explained in more details in the following subsections.

A. Color Feature Extraction

In this proposed video summarization method, color histogram [7] is applied to describe the visual content of the video frames. This technique is computationally trivial and can detect small changes of the camera position. Moreover, color histograms tend to be unique for different objects. For these reasons, this technique is widely used in automatic video summarization.

In video summarization systems, the color space selected for histogram extraction should reflect the way in which humans perceive color. This can be achieved by using user-oriented color spaces as they employ the characteristics used by humans to distinguish one color from another [5], [8]. A popular choice is the HSV color space, the HSV color space was developed to provide an intuitive representation of color and to be near to the way in which humans perceive color [5].

The color histogram used in our proposed method is computed from the HSV color space using 32 bins of H, 4 bins of S, and 2 bins of V. The quantization of the color histogram aims at reducing the amount of data without losing important information (bins values are established through experimental tests [9]).

B. Video Frames Clustering using Dynamic Modeling Framework

After extracting the color features of the original video frames, the task of grouping similar video frames is executed by clustering the extracted color features. Most of the existing clustering algorithms use a static model of the clusters and do not use information about the nature of individual clusters as they are merged. Some algorithms ignore the information about the aggregate interconnectivity of items in two clusters; while others ignore the information about

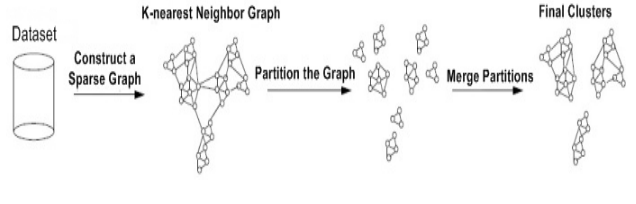

Figure 2. Overall framework of Chameleon clustering algorithm [3]. Chameleon uses a two-phase algorithm, which first partitions the data items into subclusters and then repeatedly combines these subclusters to obtain the final clusters

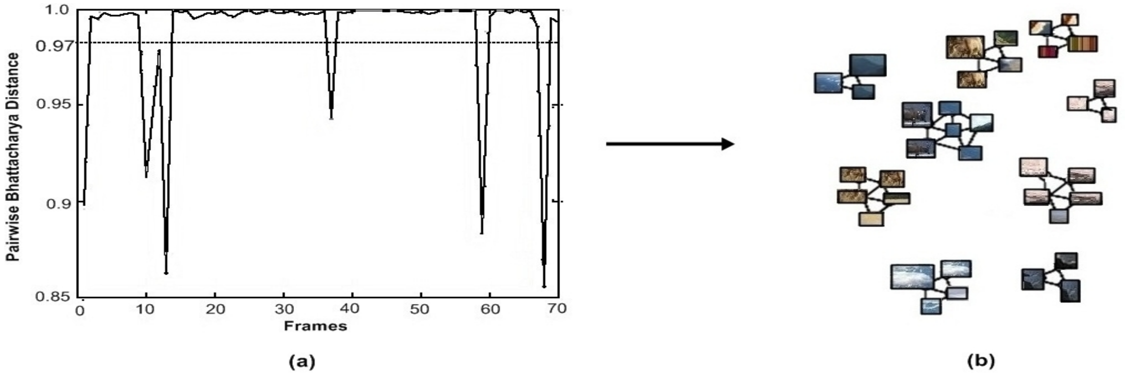

Figure 3. Video Temporal Partitioning. (a) Pairwise Bhattacharyya distances of consecutive frames of video "The Great Web of Water,segment 02", available at Open Video Project [10]. (b) Extracted video shots represented as undirected graphs(sub-clusters)

the closeness of two clusters as defined by the similarity of the closest items across two clusters. By only considering either interconnectivity or closeness, these algorithms can easily select and merge the wrong pair of clusters [3].

Unlike other clustering algorithms, Chameleon [3] is an agglomerative hierarchical clustering that uses a dynamic modeling framework which overcomes the limitations of the other clustering algorithms. The key feature of Chameleon algorithm is that it utilizes both interconnectivity and closeness to identify the most similar pair of clusters. Furthermore, Chameleon uses a novel approach to model the degree of interconnectivity and closeness between each pair of clusters. This approach considers the internal characteristics of the clusters themselves. Thus, it does not depend on a static, user-supplied model and can automatically adapt to the internal characteristics of the merged clusters [3]. Figure 2 shows the overall framework of Chameleon where the clusters in the data set are detected by using a two-phase algorithm. In the first phase, Chameleon uses a graph-partitioning algorithm [11] to cluster the data items into several relatively small sub-clusters. In the second phase, it utilizes an algorithm to find the genuine clusters by repeatedly combining these sub-clusters [3].

Inspired by the dynamic modeling framework used in Chameleon [3], we propose a modified hierarchical clustering algorithm for summarizing the video; this algorithm depends on the temporal order and sequential nature of the video input to fasten the clustering process and it utilizes the dynamic modeling framework to cluster the video frames and to extract the Key frames. In this proposed video clustering algorithm, clustering the video frames is achieved using two phases. The first phase is considered the main proposed modification; instead of building a graph and using a graph-partitioning algorithm as done in Chameleon algorithm; the partitioning process is executed using a simple video temporal segmentation process as it will be illustrated later. The second phase of the proposed algorithm utilizes dynamic modeling framework similar to that used in Chameleon but with Bhattacharya distance [12] as a dissimilarity measure.

In the following subsections, each of the two phases of the video frames clustering algorithm is discussed in details.

*1) **Phase I: Video Temporal Partitioning**:* Video stream has a temporal structure, as it consists of a set of consecutive frames arranged by the time of appearance. The proposed video summarization method makes use of this temporal structure in order to divide the video into partitions. This process is also known as video segmentation.

In this phase, video segmentation process is executed using a simple shot boundary detection method similar to the one used in [5], [13]. In this method the pairwise distances of consecutive frames are computed from color features in the extracted sample. The color features are extracted as shown in the previous section. Instead of using Euclidean distance, the Bhattacharyya distance [12] is used to compute the pairwise distances between the consecutive frames.

The Bhattacharyya distance between two histograms P and Q of size n; is defined as:

$$\textit{Bhattacharyya Distance} = \sum_{i=0}^{n} \sqrt{\sum Pi \bullet \sum Qi} \qquad (1)$$

In this method, every time the Bhattacharyya distance between two consecutive frames is less than threshold T, a shot is created. The threshold value applied in this work is equal to 0.97 (threshold value established through experimental tests). Figure 3(a) shows the pairwise Bhattacharyya distances of sampled frames of the video the Great Web of Water, segment 02 (video is available at Open Video Project [10]).

For every detected video shot, a weighted undirected graph is built using the video frames as nodes. In this graph, every two video frames are connected only if the Bhattacharya distance between them is equal or greater than the predetermined threshold 0.97. Also, it is well to notice that this graph can be constructed in parallel while detecting the video shot boundaries. This mechanism is actually building Eps-Farthest Neighbor graph, where the Eps value is the threshold of the Bhattacharya distance which is set to 0.97 according to our experimental tests. The output of this phase consists of a set of video shots or sub-clusters, where each one is represented by a weighted undirected graph. Figure 3 shows the process illustrated in this phase.

Using the Bhattacharyya distance as a dissimilarity measure has many advantages [14]. First, the Bhattacharyya measure has self-consistency Property as all poisson errors are forced to be constant therefore ensuring the minimum distance between two observations points is indeed a straight line. Second advantage is the independence between Bhattacharyya measure and bin widths, as the Bhattacharyya metric the contribution to the measure is the same irrespective of how the quantities are divided between bins. Therefore the Bhattacharyya statistic is unaffected by the distribution of data across the histogram and is the only form of sum-of-product functions with this property. Finally, the Bhattacharyya measure is dimensionless, as it is not affected by the measurement scale used, when Bhattacharyya measure is used to compare two identical distributions, it has been proven that the term is maximized to a value of one [14].

*2) **Phase II: Merging Video Shots**:* This phase aims at finding the genuine video clusters using a dynamic modeling framework similar to the one used in Chameleon [3]. The input to this phase are the video shots generated from the previous phase which are considered as primary sub-clusters. In this phase, the dynamic modeling framework is used to determine the similarity between pairs of primary clusters by looking at their relative interconnectivity (RI) and relative closeness (RC). The pairs are selected to merge for which both RI and RC are high; as it selects clusters that are well interconnected as well as close together.

The relative interconnectivity (RI) between a pair of video clusters C_i and C_j is given by

$$RI(C_i, C_j) = \frac{\mid EC_{(C_i,C_j)} \mid}{\frac{|EC_{(C_i)}|+|EC_{(C_j)}|}{2}} \tag{2}$$

where $EC_{(C_i,C_j)}$ is absolute inter-connectivity between C_i and C_j, and $EC_{(C_i)}$, $EC_{(C_j)}$ are internal interconnectivity of the two video clusters C_i and C_j respectively. The absolute interconnectivity between a pair of video clusters C_i and C_j is the sum of the weight of the edges that connect video frames in C_i to video frames in C_j. This is the edge cut of the cluster containing both C_i and C_j such that the cluster is broken into C_i and C_j. The internal interconnectivity of a video cluster C_i is calculated by the size of its min-cut bisector, as done in [3] (i.e., the weighted sum of edges that partition the graph into two roughly equal parts).

The relative closeness (RC) between a pair of video clusters C_i and C_j is given by

$$RC(C_i, C_j) = \frac{\overline{S}EC_{(C_i,C_j)}}{\frac{|C_i|}{|C_i|+|C_j|}\overline{S}EC_{(C_i)} + \frac{|C_j|}{|C_i|+|C_j|}\overline{S}EC_{(C_j)}} \tag{3}$$

Where $\overline{S}EC_{(C_i)}$ and $\overline{S}EC_{(C_j)}$ are the average weights of the edges that belong in the min-cut bisector of clusters C_i and C_j respectively. While $\overline{S}EC_{(C_i,C_j)}$ is the average weight of the edges that connect video frames in C_i and C_j. Terms $\mid C_i \mid$ and $\mid C_j \mid$ are the number of data points in each cluster. This equation normalizes the absolute closeness of the two clusters by the weighted average of the internal closeness of C_i and C_j which discourages merging small sparse clusters into large dense clusters [3].

It is worth to note that the dynamic framework for modeling cluster similarity is applicable only when each cluster contains a sufficiently large number of data items (video frames in this case). The reason is that to compute the relative interconnectivity and closeness of clusters, it is required to compute each cluster's internal interconnectivity and closeness, neither of which can be accurately calculated for clusters containing a few data items [3]. For that reason, in this work, the original input video frames are used for segmentation phase (first phase), instead of pre-sampling the video as done in other video summarization approaches.

The next step in this phase is merging the video clusters, the pairs of video clusters are merged if the relative interconnectivity and relative closeness exceeds user-specified thresholds for T_{RI} and T_{RC} respectively. Each video cluster C_i is visited and merged with adjacent video cluster C_j with RI and RC that exceed these thresholds. According to our experimental tests, the thresholds values are set to T_{RI} = 0.9 and T_{RC} = 0.9

C. Key Frames Extraction

After clustering the video frames, the final step is selecting the key frames from the genuine video clusters. For each cluster the middle frame in the ordered frames sequence is selected to construct the static video summary. According to our experiments, we found that this middle frame is the best representative of the cluster to which it belongs. Figure 4 shows the extracted video frames of the video Senses and Sensitivity, Introduction to Lecture 2, video available at Open Video Project [10].

IV. Experimental Evaluation

Evaluation of static video summaries is a challenging problem and many methods have been proposed to assess the quality of the video summaries [15]. In [5], an evaluation method called Comparison of User Summaries (CUS) is used to evaluate the quality of video summaries. In CUS method, the video summary is built manually by a number

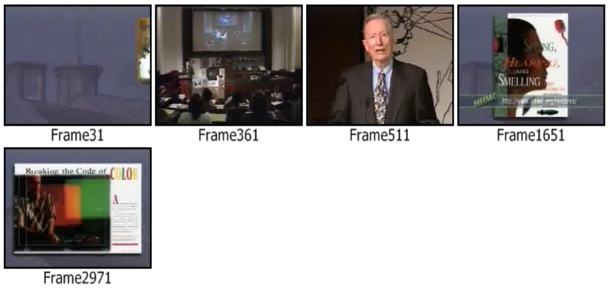

Figure 4. Static video summary of the video Senses and Sensitivity, Introduction to Lecture 2, video available at Open Video Project [10]

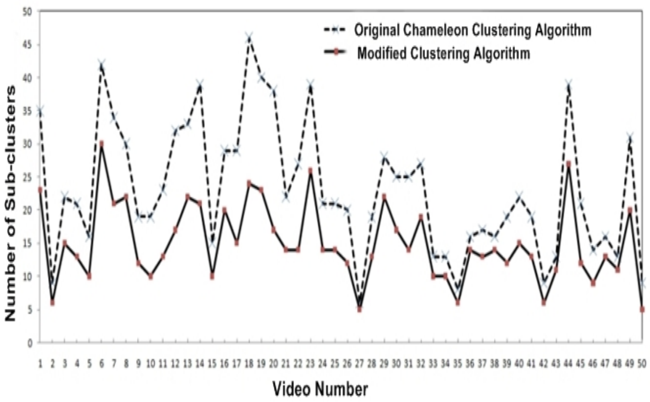

Figure 5. Comparing number of generated sub-clusters (primary clusters) in first phase using the original Chameleon clustering algorithm and the proposed modified clustering algorithm, the x-axis shows the video numbers which corresponds to the 50 videos used in the experimental evaluation

Video Summarization Method	Mean F-Measure
OV [17]	0.65
DT [6]	0.59
STIMO [4]	0.64
VSUMM [5]	0.71
Proposed Method	**0.78**

Table I
MEAN F-MEASURE ACHIEVED BY DIFFERENT VIDEO SUMMARIZATION METHODS.

of users from the sampled frames and the user summaries are taken as reference (i.e. ground truth) to be compared with the automatic summaries obtained by different methods using the color features and Euclidean distance as dissimilarity measure.

In this paper, an evaluation method similar to CUS method [5] is used to assess the quality of the video summaries. Each key frame in the automatic video summary is compared with frames in the ground truth video summary using the extracted color features, these features are extracted as previously mentioned in section 3. Instead of using Euclidean distance as dissimilarity measure as in [5], the Bhattacharya distance is used due to its advantages [14], which are discussed in the previous section. In the comparison process, if the calculated Bhattacharya distance between the compared frames is greater than or equal to a threshold value of 0.97 (established through experimental tests), these two frames are considered similar, and they are excluded from the next iteration. This process continues until all of the automatic video summary frames or user summary frames are processed.

In order to evaluate the automatic video summary, the F-measure is used as a metric. The F-measure consolidates both Precision and Recall values into one value using the harmonic mean [16], and it is defined as:

$$F\text{-}measure = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4}$$

The Precision measure of video summary is defined as the ratio of the total number of similar frames to the total number of frames in the automatic summary; and the Recall measure is defined as the ratio of the total number of similar frames to the total number of frames in the user summary. Our proposed video summarization method is evaluated on a set of 50 videos selected from the Open Video Project [10]. All videos are in MPEG-1 format (30 fps, 352 X 240 pixels). They are distributed among several genres (documentary, historical, lecture, educational) and their duration varies from 1 to 4 min. Also, we use the user summaries used in [5] as a ground-truth data. These user summaries were created by 50 users, each one dealing with 5 videos, meaning that each video has 5 summaries created by five different users. So, the total number of video summaries created by the users is 250 summaries and each user may create different summary.

For comparing the proposed method with others, we used the results reported by three video summarization methods: VSUMM [5], STIMO [4], and DT [6]. In addition to that, the automatic video summaries generated by our method were compared with the OV summaries generated by the algorithm in [17]. Table I shows the mean F-measure achieved by the different video summarization methods. The results indicate that our proposed video summarization method performs better than all other methods. All the videos, user summaries, and automatic summaries in addition to the detailed results for each method are available publicly [1] .

In addition to the previous results, our experiments shows that the modified clustering algorithm that has been proposed in this video summarization method allowed us to generate a faster video summaries than using the original Chameleon clustering algorithm. As a result of using the video temporal partitioning phase that is proposed in this work, the number of the sub-clusters (primary clusters) generated in the first phase were less than the number of sub-clusters generated

[1] https://sites.google.com/site/vchamsite/

by the original Chameleon clustering algorithm in which a graph-partitioning algorithm [11] is used to generate the sub-clusters. As a consequence, the second phase which is merging sub-clusters will take much less time if compared to the time taken by the original Chameleon algorithm's merging process. This improvement is mainly due to utilizing the temporal order and sequential nature of thge video by using a shot boundary detection algorithm to partition the video which is discussed in section 3. Figure 5 shows the number of the generated sub-clusters (primary clusters) in the first phase using the original Chameleon clustering algorithm and the proposed modified clustering algorithm.

V. Conclusion

In this paper, an enhanced static video summarization method is presented. This method utilizes a modified dynamic modeling-based hierarchical clustering algorithm that depends on the temporal order and sequential nature of the video to fasten the clustering process. In this method, the color histogram on HSV color space is used to represent the video frames; and the Bhattacharyya distance is used as a dissimilarity measure.

The video summaries generated by our method are compared with summaries generated by others found in the literature and the ground truth summaries. Experimental results indicate that using the dynamic modeling-based clustering algorithm in the proposed video summarization method gives a higher quality video summaries. Also, experiments shows that the proposed modification of the first phase in the Chameleon clustering algorithm fastens the video clustering process.

Future work includes combining other features to the proposed approach like texture, edge and motion descriptors. Also, another interesting future work could be generating video skims (dynamic key frames, e.g. movie trailers) from the extracted key frames. Since the video summarization step is usually considered as a prerequisite for video skimming [15], the extracted key frames from the proposed method can be used to develop an enhanced video skimming system.

References

[1] S. Pfeiffer, R. Lienhart, S. Fischer, and W. Effelsberg, "Abstracting digital movies automatically," *Journal of Visual Communication and Image Representation*, vol. 7, no. 4, pp. 345–353, 1996.

[2] S. J. F. Guimaraes and W. Gomes, "A static video summarization method based on hierarchical clustering," *Pattern Recognition, Image Analysis, Computer Vision, and Applications*, pp. 46–54, 2010.

[3] G. Karypis, E.-H. Han, and V. Kumar, "Chameleon: Hierarchical clustering using dynamic modeling," *Computer*, vol. 32, no. 8, pp. 68–75, 1999.

[4] M. Furini, F. Geraci, M. Montangero, and M. Pellegrini, "STIMO: STIll and MOving video storyboard for the web scenario," *Multimedia Tools and Applications*, vol. 46, no. 1, pp. 47–69, 2010.

[5] S. E. F. de Avila, A. P. B. Lopes *et al.*, "VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method," *Pattern Recognition Letters*, vol. 32, no. 1, pp. 56–68, 2011.

[6] P. Mundur, Y. Rao, and Y. Yesha, "Keyframe-based video summarization using Delaunay clustering," *International Journal on Digital Libraries*, vol. 6, no. 2, pp. 219–232, 2006.

[7] M. J. Swain and D. H. Ballard, "Color indexing," *International journal of computer vision*, vol. 7, no. 1, pp. 11–32, 1991.

[8] R. O. Stehling, M. A. Nascimento, and A. X. Falcao, "Techniques for color-based image retrieval," *Multimedia Mining*, pp. 61–82, 2002.

[9] K. M. Mahmoud, M. A. Ismail, and N. M. Ghanem, "VS-CAN: An Enhanced Video Summarization Using Density-Based Spatial Clustering," in *Image Analysis and Processing Conference-ICIAP 2013*, vol. 1. Springer Berlin Heidelberg, 2013, pp. 733–742.

[10] Open video project. [online]. available: http://www.open-video.org. [Online]. Available: http://www.open-video.org

[11] G. Karypis, E.-H. Han, and V. Kumar, "hMETIS: A Hypergraph Partitioning Package," *Tech. Report, Dept. of Computer Science, Univ. of Minnesota, 1998; http://winter.cs. umn.edu/ karypis/metis.*

[12] T. Kailath, "The divergence and Bhattacharyya distance measures in signal selection," *Communication Technology, IEEE Transactions on*, vol. 15, no. 1, pp. 52–60, 1967.

[13] Guimaraes, S. J. F., M. Couprie, A. de Albuquerque Araujo, and N. Jeronimo Leite, "Video segmentation based on 2D image analysis," *Pattern Recognition Letters*, vol. 24, no. 7, pp. 947–957, 2003.

[14] F. J. Aherne, N. A. Thacker, and P. I. Rockett, "The Bhattacharyya metric as an absolute similarity measure for frequency coded data," *Kybernetika*, vol. 34, no. 4, pp. 363–368, 1998.

[15] B. T. Truong and S. Venkatesh, "Video abstraction: A systematic review and classification," *ACM Transactions on Multimedia Computing, Communications, and Applications (TOMCCAP)*, vol. 3, no. 1, p. 3, 2007.

[16] H. M. Blanken, A. De Vries, H. E. Blok, and L. Feng, *Multimedia retrieval*. Springer Verlag, 2007.

[17] D. DeMenthon, V. Kobla, and D. Doermann, "Video summarization by curve simplification," in *Proceedings of the sixth ACM international conference on Multimedia*. ACM, 1998, pp. 211–218.