# Video Key Frame Extraction through Canonical Correlation Analysis and Graph Modularity

Rameswar Panda, Sanjay K. Kuanar, and Ananda S. Chowdhury

Department of Electronics and Telecommunication Engineering
Jadavpur University, Kolkata - 700032, India
{rameswar183,sanjay.kuanar}@gmail.com,
{aschowdhury@etce.jdvu.ac.in}

**Abstract.** Key frame based video summarization has emerged as an important area of multimedia research in recent times. In this paper, we propose a novel automated approach for video key frame extraction in compressed domain using canonical correlation analysis (CCA) and graph modularity. We prune certain edges from the Video Similarity Graph (VSG) using an iterative strategy until there is no improvement in graph modularity. Resulting connected components in the final VSG correspond to separate clusters. The proposed algorithm also uses multi-feature fusion using canonical correlation analysis to achieve higher semantic dependency between different video frames. Experimental results on some standard videos of different genre clearly indicate the superiority of the proposed method in terms of the $F_1$ measure.

**Keywords:** Key frames, Video Summarization, Canonical Correlation Analysis, Graph modularity.

## 1 Introduction

Recently, there has been a drastic increase in creation and storage of multimedia contents on the web. For example, as of now, one of the most primary video sharing web sites like YouTube reported that more than 1 billion unique users visit YouTube each month and 72 hours of video are uploaded every minute [13]. The evegrowing number of videos has motivated researchers to design efficient and effective video management schemes in order to provide a better overall multimedia experience to the consumer [1]. Summarization is mainly used to provide a condensed version of a full-length video through the identification of most important content within it [2,4]. Video skimming and key frame extraction are the two basic methods for summarizing videos [1-4]. A video of much shorter duration than the actual one is produced in the later case whereas the former deals with extracting some salient frames from the videos which preserves the overall content of a video with minimum data. Though the technique of skimming provides more information because of the audio and motion contents, key frames summarize the video content in a more rapid and compact manner. Moreover, key frame extraction can be used as a pre-processing step in various video analytics

P. Maji et al. (Eds.): PReMI 2013, LNCS 8251, pp. 410–416, 2013.

applications which suffer from the problem of processing large number of video frames [3]. The focus of this paper is extraction of key frames from the videos.

Different clustering techniques have been proposed in the literature to address the problem of extracting key frames from a video sequence [4-5, 10]. The performance of these methods heavily depends on user inputs and/or certain threshold parameters [5]. Moreover, many of those research works have focused on the uncompressed domain which makes the system unsuitable for online usage. Although the work presented in [6] is in compressed domain it utilizes the notion of similarity between successive frames. However, choice of similarity measures greatly influences the effective content representation of the key frame set. Most of the approaches use a single visual feature, *i.e.*, color to represent a video frame [5-6]. However, color alone cannot in general capture all the pictorial details needed to estimate the changes in the visual content of frames. Recently, Naveed et al. [3] proposed a key frame extraction technique based on a weighted combination of several features but their approach is too sensitive to selection of many control parameters.

In this paper, we present a novel approach for key frame extraction using multi-feature fusion via canonical correlation analysis (CCA) [7] and graph modularity clustering in compressed domain. Firstly, several features extracted from compressed domain I-frames are combined using CCA to form a single feature representing a video frame. Secondly, video similarity graph (VSG) is constructed over the video frames. Finally, an edge pruning strategy is repeated until there is no improvement in graph modularity.

## 2 Proposed Methodology

The proposed method consists of four main steps: (1) Feature extraction from compressed domain I-frames; (2) Multi-feature fusion with CCA; (3) Graph Modularity clustering; (4) Key frame extraction.

### 2.1 Feature Extraction

Since video data are usually available in compressed form, it is desirable to directly process the compressed video without decoding. Specifically, most video codecs (i.e., MPEG-1/2/4) are based on group of pictures(GOPs) as basic units [6]. The content of a GOP is represented using I-frames. The compression of the I-frames of a MPEG video is carried out by dividing the original image into 8 x 8 pixel blocks and transforming the pixels values of each block into 64 DCT coefficients [6]. Finally, on extracting the DC term of all the pixel blocks, a reduced version of the original image (known as DC image) is produced whose size is only 1/64 of original image.

In general, a single visual descriptor cannot capture all the pictorial details needed to estimate the changes in the visual content of frames. Hence we have extracted three features such as color (256 bins) [4], texture (i.e., edge, 80 bins) [4] and wavelet feature (20 bins) [9] from DC images.

### 2.2 Multi-feature Fusion

Since the feature extraction step produces three distinct feature vectors of different dimensions, we apply CCA to combine them [7]. In recent years, CCA has been applied to several fields as one of the most valuable multi-data processing methods [7] . CCA is a statistical method that finds a pair of directions which maximizes the correlation between projections of two random vectors. These projections are called canonical variates. This process of finding projections or directions is repeated until all the correlation features of the two random vectors are obtained. Algorithm 1 summarizes the steps involved in CCA fusion of the three features.

**Algorithm 1. Multi-feature fusion**
**INPUT**: Given three feature vectors: Color (C), Edge (E) and Wavelet (W)
**OUTPUT**: Canonically Correlated Feature Vector (CCFV)
1: **Procedure** (C, E, W, CCFV)
2: Assign X=C (256 elements), Y=E+W (100 elements).
3: Compute Covariance matrices $S_{xx}, S_{yy}$ and between set covariance $S_{xy}$.
4: Compute $G_1$ and $G_2$.
$G_1 = S_{xx}^{-1/2} S_{xy} S_{yy}^{-1} S_{yx} S_{xx}^{-1/2}$,
$G_2 = S_{yy}^{-1/2} S_{yx} S_{xx}^{-1} S_{xy} S_{yy}^{-1/2}$.
5: Compute orthogonal eigen vectors $u_i, v_i$ and $r = rank\,(S_{xy})$
$\alpha_i = S_{xx}^{-1/2} u_i, \beta_i = S_{yy}^{-1/2} v_i, i = 1, 2, \cdots r$
6: Choose first $d\,(i.e., 100)$ pairs of $\alpha_i$ and $\beta_i$ to make $W_x$ and $W_y$.
$W_x = (\alpha_1, \alpha_2, \cdots, \alpha_d)\,, W_y = (\beta_1, \beta_2, \cdots, \beta_d)\,.$
7: Compute Canonically Correlated Feature Vector (CCFV)
$CCFV = {W_x}^T x + {W_y}^T y$
8: **End Procedure**.

### 2.3 Graph Modularity Clustering

We then construct the video similarity graph (VSG) using the data points (i.e., I-frames) in the refined feature space (CCFV) as its vertices. VSG is represented by $G = (V, E, W)$, where $V$ is the set of nodes, $E$ is the set of edges connecting the nodes and $W$ is the set of weights corresponding to the strength of edges. The weights $W_{ij}$ between two frames is defined as :-

$$W_{ij} = exp\left(-d_{ij}^2/\sigma^2\right) \tag{1}$$

where $d_{ij}$ is the histogram intersection distance [9] between frames $i$ and $j$. $\sigma$ is a scaling parameter that determines the extent of similarity between two frames. As suggested in [8], $\sigma = \beta * max\,(d)$, where $\beta \leq 0.2$ and d is the set of all pairwise distances. In the VSG, the edges can be grouped into intra-cluster edges (edges whose end points are in the same cluster) and inter-cluster edges (edges whose end points are in different clusters). Our objective is to preserve

the intra-cluster edges and remove the inter-cluster edges which connect the individual clusters in an efficient manner. In our method, we prune certain edges depending on the difference between edge weight and scaling parameter, until there is no improvement in graph modularity [11]. Modularity$M(c_1, c_2, \cdots, c_k)$ of a graph clustering over $k$ known clusters $c_1, c_2, \cdots, c_k$ is defined as

$$M(c_1, c_2, \cdots, c_k) = \sum_{i=1}^{k} \delta_{i,i} - \sum_{i \neq j} \delta_{i,j} \tag{2}$$

where $\delta_{i,j} = \sum_{\{v,u\} \in E, v \in c_i, u \in c_j} w(v,u)$, with each edge$\{v,u\} \in E$ included at most once in the computation. High value of modularity indicates good clustering. Remaining connected components of the final VSG after end of edge pruning represent individual clusters. Algorithm 2 summarizes the steps involved in graph modularity clustering.

**Algorithm 2. Graph Modularity Clustering**
**INPUT:** Video Similarity Graph (VSG), $E$, $W$, $N$= number of I-frames, $\sigma$
**OUTPUT:** $Clusters c_1, c_2, \cdots, c_k$
1: **Procedure** $(V, E, W, c_1, c_2, \cdots, c_k)$
2: *for* $i = 1$ *to* $N$
3: *for* $j = 1$ *to* $N$
4: $Dev_{ij} \stackrel{\Delta}{=} d_{ij} - \sigma$
5: End
6: End
7 : **Repeat**
8: Select edges which has high value of *Dev*, remove the edge from VSG
9: Find connected components from the VSG
10: Calculate Modularity ($M$)
11: **Until** no improvement in Modularity over two successive iterations.
12: Obtain individual clusters from final VSG.
13: **End Procedure**.

### 2.4 Key Frame Extraction

The frames which are closest to the centroids of each cluster are deemed as the key frames. Finally, the key frames are arranged in a temporal order to make the produced summary more understandable.

## 3 Experimental Results

### 3.1 Evaluation Dataset and Performance Measures

Ten video segments from Open Video (OV) projects [12] are used in the experiments to test the performance of our proposed method. The test set presents

a variety of video genres with different durations (46 sec. to 2 min). Each test video is in MPEG-1 format with a frame rate of 29.97 and the frames having dimensions of 352 x 240 pixels. Long videos are avoided due to limitation of annotation by a subject. The test set has also an intersection with the work in [5,6]. All the experiments are performed on a machine with Intel(R) core(TM) i5-2400 processor and 8 GB of DDR2-memory. Unlike other multimedia research areas, a consistent evaluation framework for video summarization is seriously missing possibly due to the lack of an objective ground-truth. In this work, we evaluated our technique based on the Comparison of User Summaries (CUS) mechanism proposed by Avila et al. [5]. This evaluation scheme is also adopted by various video summarization approaches for performance comparison [3, 5-6, 10]. In this evaluation scheme, the manually created user summaries are taken as references for comparison with the summaries generated by the automated methods. The performance was validated by $F_1$-measure which is a function of both precision and recall [6,10].

$$F_1 = \frac{2 * precision * recall}{precision * recall} \tag{3}$$

Here precision is the ratio of the number of matching frames to the total number of frames in the automatic summary and recall is the ratio of the number of matching frames to the total number of frames in the user summary.

### 3.2 Performance Comparison

In this section we make a comparative performance analysis of our proposed method with two well-known recent methods VSUMM [5] and VISON [6] using $F_1$-measure. Results of the proposed method can be seen at `https://sites.google.com/site/ivprgroup/result/vkcg`. The user summaries, and the key frames produced by the approach [5] are available at `https://www.sites.google.com/site/vsummsite/` and for the approach [6] are available at `http://www.liv.ic.unicamp.br/~jurandy/summaries/`. Table 1 presents the mean $F_1$-measure achieved by all the methods for several video categories. The results indicate that our proposed method performs better than VSUMM and VISON (improvement of 14.83% over VSUMM and 19.75% over VISON). In order to verify the statistical significance of those results, the confidence intervals for the differences between paired means are computed [5-6, 10]. Since the confidence intervals (with a confidence of 98%) do not include zero in both the cases (from Table 2), the results presented in Table 1 confirm that proposed method outperforms both VISON and VSUMM in statistically significant manner. Fig. 1 presents the key frames produced by the approaches for the video **The Voyage of the Lee, segment 15**. The user summaries for the same video are presented in Fig. 2. From Fig. 2, it can be noted that for most of the user summaries, our proposed method achieves higher $F_1$ value as compared to the other methods. The summary with the highest quality is achieved by our approach, which can also be confirmed by a visual comparison with the user summaries.

**Table 1.** Mean $F_1$ measure Comparative performance analysis for several video categories

| Category | Videos | VSUMM | VISON | Proposed Method |
|---|---|---|---|---|
| Documentary | 6 | 0.622 | 0.619 | 0.749 |
| Educational | 2 | 0.781 | 0.739 | 0.804 |
| Lecture | 2 | 0.758 | 0.673 | 0.860 |
| **Weighted Average** | **10** | **0.681** | **0.653** | **0.782** |

**Table 2.** Difference between Mean $F_1$ measure at a confidence of 98%

| Method | Confidence Interval (98%) | |
|---|---|---|
| | Min. | Max. |
| Proposed method - VISON | 0.142 | 0.269 |
| Proposed method - VSUMM | 0.094 | 0.173 |

**Fig. 1.** Summarization results for the video *The Voyage of the Lee, segment 15*: **Top row** →Proposed Method (Mean $F_1$-measure: 0.717), **Middle row** → VISON [6] (Mean $F_1$-measure: 0.515), **Bottom row** → VSUMM [5] (Mean $F_1$-measure: 0.666)

**Fig. 2.** User summaries for the video *The Voyage of the Lee, segment 15:* **Row1** → USER1: $F_1$(Proposed Method) = 0.591, $F_1$(VISON) = 0.362, $F_1$(VSUMM) = 0.568, **Row2** →USER2: $F_1$(Proposed Method) = 0.538, $F_1$(VISON) = 0.500, $F_1$(VSUMM) = 0.750, **Row3** → USER3: $F_1$(Proposed Method) = 0.768, $F_1$(VISON) = 0.428, $F_1$(VSUMM) = 0.600, **Row4** → USER4: $F_1$(Proposed Method) = 0.833, $F_1$(VISON) = 0.615, $F_1$(VSUMM) = 0.788, **Row5** → USER5: $F_1$(Proposed Method) = 0.857, $F_1$(VISON) = 0.671, $F_1$(VSUMM) = 0.626

## 4 Conclusion

In this paper, we propose a novel automatic key frame based video summarization technique using multi-feature fusion with canonical correlation analysis and graph modularity clustering in the compressed domain. Experimental results show that our technique outperforms the work described in [5] and [6]. Future work will focus on integration of visual attention model and inclusion of shape features to design an efficient system for search-and-retrieval of video sequences. Another direction of future research is to produce personalized key frames with various forms of unobtrusively sourced user-based informations.

## References

1. Ejaz, N., Mehmood, I., Baik, S.W.: Efficient visual attention based framework for extracting key frames from videos. Signal Processing: Image Communication 28, 34–44 (2013)
2. Cong, Y., Yuan, J., Luo, J.: Towards Scalable Summarization of Consumer Videos Via Sparse Dictionary Selection. IEEE Transactions on Multimedia 14, 66–75 (2012)
3. Ejaz, N., Tariq, T.B., Baik, S.W.: Adaptive key frame extraction for video summarization using an aggregation mechanism. J. Visual Communication and Image Representation 23, 1031–1040 (2012)
4. Chowdhury, A.S., Kuanar, S.K., Panda, R., Das, M.N.: Video Storyboard Design using Delaunay Graphs. In: 21st IEEE International Conference on Pattern Recognition, pp. 3108–3111 (2012)
5. Avila, S.E.F., Lopes, A.P.B., Luz Jr., A., Araujo, A.A.: VSUMM: A mechanism designed to produce static video summaries and a novel evaluation method. Pattern Recognition Letters 32, 56–68 (2011)
6. Almeida, J., Leite, N.J., Torres, R.S.: VISON: Video Summarization for Online applications. Pattern Recognition Letters 33, 397–409 (2012)
7. Sun, Q.S., Zeng, S.G., Liu, Y., Heng, P.A., Xia, D.S.: A new method of feature fusion and its application in image recognition. Pattern Recognition Letters 38, 2437–2448 (2005)
8. Shi, J., Malik, J.: Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 22, 888–905 (2000)
9. Ciocca, G., Schettini, R.: A innovative algorithm for key frame extraction in video summarization. J. of Real-time Image Processing 1, 69–88 (2006)
10. Panda, R., Kuanar, S.K., Chowdhury, A.S.: VISUC: Video Summarization With User Customization. In: IEEE International Conference on Communications, Devices and Intelligent Systems, pp. 89–92 (2012)
11. Schaeffer, S.E.: Graph clustering. Computer Science Review 1, 27–64 (2007)
12. The Open Video Project, `http://www.open-video.org`
13. YouTube Press Statistics, `http://www.youtube.com/t/press_statistics`