

Keyframe Extraction and Shot Boundary Detection Using Eigen Values

Vijetkumar Benni, R. Dinesh, Punitha P., and Vasudeva Rao

Abstract—In this paper, a new technique for keyframe extraction and shot boundary detection is proposed. The proposed technique uses Eigen values for measuring the dissimilarity between the consecutive frames. The behavior of the Eigen values of covariance matrix over certain frames is analyzed to detect the shot boundary and subsequently extract the keyframe. In this paper a novel method for dynamically updating the covariance matrix is also proposed. This greatly reduces the computational time compared to conventional method of calculating the covariance matrix. The proposed technique is tested on a large variety of videos and also works on live videos. The recall and precision of simulation results reveal that the proposed technique is highly efficient and accurate in detecting the abrupt cuts.

Index Terms—Computational efficiency, covariance matrix, dynamic update, Eigen values, keyframe extraction, Shot boundary detection.

I. INTRODUCTION

The increasing demand for the storage of a variety of visual information has made the researchers active in investigating mechanisms to efficiently index the visual data and effectively retrieve the same when needed. Content-based video retrieval (CBVR) is one such area of research which has been catering to search demands from the users. In a CBVR system, a video is first segmented into successive shots in the temporal domain which is a simple process of segmenting the video at certain timelines [1-22, 23-30]. However, to automate the process of shot segmentation, analysis of the subsequent frame for the changes in content is essential. The changes can be abrupt or gradual, which mark the boundaries and shots due to various transition such as hard cuts, dissolve, fade in, fadeout etc. Later the keyframes are extracted from the segmented shots. Keyframes provide a suitable abstraction and framework for video indexing, browsing and retrieval as they are the representative frames of the segmented shots. The use of keyframes greatly reduces the amount of data required in video indexing and provides an organizational framework for dealing with video content. Users, while searching for a video of their interest, browse the videos in a random fashion, by stopping the video at different timelines and viewing only certain keyframes to match the content of search query. Content-based search and retrieval of video is thus a vital and

challenging task for processing digital video database. CBVR hence has various stages of processing such as shot segmentation, keyframe extraction, feature extraction, feature indexing, retrieval mechanism and result ranking mechanism, a lot of work can be found in the literature for each one of these stages. The focus of this paper is only towards shot boundary detection and keyframe extraction. The behavior of Eigen values is used to measure the dissimilarity between the frames of the video to detect the shot boundaries and extract the keyframes.

The remaining part of the paper is organized as follows: in Section II related work is discussed, in Section III the proposed method is explained in detail, experiment results is discussed in Section IV and conclusion and future work in Section V.

II. RELATED WORK

Different shot boundary detection algorithms, such as histogram-based algorithms [1], [2], motion-based algorithms [3], [4] and contour-based algorithms [5] for cut detection; twin-comparison algorithm [1] and production-model based algorithms [6] for gradual transition detection have been proposed in the literature.

The most common approach to obtain boundaries of a shot is based on color information of the frame, [7]-[10]. The main advantages of using color space is its ease of implementation, descriptive characteristic both in spatial and temporal space and real-time applicability due to simplicity to obtain a feature vector, such as histogram [11]. However, color based algorithms have false positives in presence of camera motion, object motion or illumination change. In most cases, post processing is accompanied to compensate for these stumbling blocks of color-based shot detection, such as motion compensation [12] or illumination reduction [7]. However, using post processing introduces the well-known tradeoff between speed and accuracy; the more you improve accuracy the longer it takes.

Shot detection algorithms based on statistical differences between the frames, extract statistical characteristics of the spatial information of the frames and identify the changes in temporal domain to detect the shots. Most commonly used features are the mean and the standard deviation [13], however these algorithms are slow due to high computational cost of statistical formulae. These algorithms also generate false positives on illumination change [14]. Another approach based on split and merge concept [15] uses spectral clustering for grouping of objects in each frame, subsequently apply 2DFLD (2D Fisher's Linear Discriminant) for the detection of shots.

Manuscript received April 24, 2014; revised June 23, 2014.

Vijetkumar Benni, P. Punitha, and Vasudeva Rao are with Imaging Tech Lab, HCL Technologies, Bangalore, India (e-mail: vijetkumar.Benni@hcl.com, drpunitha.s@hcl.com, vasudeva.r@hcl.com).

R. Dinesh was with Imaging Tech Lab, HCL Technologies, Bangalore. He is now with Amazon, Bangalore, India (e-mail: dr.dineshr@gmail.com).

Edge map based [16], [8] approaches for shot detection use the assumption that the changes in spatial domain will result in appearing or fading of edges. The percentage of change in edges decides the shot boundary. The edge detection method does not handle rapid changes in overall scene brightness or scenes which are very dark or very bright. Secondly, motion compensation technique does not handle multiple rapidly moving objects particularly well. For example rapid changes in overall scene brightness can cause a false positive.

Shot boundary detection based on Eigen coefficients and small Eigen value [17], presents an adaptive and non-parametric approach for shot boundary detection, without limiting itself to a specific type of shot transition. The difference between two subsequent frames in the video is computed in terms of Eigen distance between two subsequent frames of a video. The frame difference and the first frame number are regarded as the co-ordinate points on an open contour. The corner detection method is then used to determine the shot boundaries based on automatic computation of frame support, for each frame in the video using the statistical and geometrical properties associated with the small Eigen value of the covariance matrix of a sequence of connected points on the open contour. Shot detection using principal coordinate system [18] uses Eigen space decomposition of the RGB color space to describe the frames in a more descriptive coordinate system.

The subsequent phase after shot detection is Key-Frame Extraction. A keyframe is a representative of each shot detected. Key-frames extraction scheme based on SVD and correlation minimization proposed in [19], apply SVD on each frame of a shot and gather the singular values (eigenvalues) to form a feature vector that compactly describes frame content. Then, key-frames are optimally extracted for each shot, based on the minimization of an objective numerical criterion, i.e., the cross correlation function of frame feature vectors. Another approach based on spectral entropy and mutual information [11] uses the difference of entropy value computed from Eigen value matrix of consecutive frames to decide which frames to choose as keyframe. Eigen-image based video segmentation and indexing [20] is based on a temporally windowed principal component analysis (PCA) of a subsampled version of the video sequence. Two discriminants are derived from the principal components on a frame by frame basis. The discriminants are used for scene change detection and keyframe extraction and classification into relevant clips.

The approach proposed in this paper for shot detection and keyframe extraction, computes the eigenvalues associated with the covariance matrix of the intensity values of consecutive frames. In the proposed method, a frame is said to be a keyframe if the corresponding small eigenvalue exceeds a predefined threshold value and thus the shot is segmented. The proposed method is explained in the following section.

III. THE PROPOSED METHOD FOR SHOT DETECTION AND KEYFRAME EXTRACTION

The proposed method for shot segmentation and keyframe extraction has four stages: generation of data matrix for k

consecutive frames, computation of covariance matrix, computation of Eigen values and shot boundary detection/keyframe extraction. Initially the covariance matrix of k consecutive frames of the video stream is calculated. The Eigen values of this covariance matrix is calculated which are used to find the similarities/dissimilarities between the consecutive frames. Based on the similarities/ dissimilarities the keyframes are extracted. These stages are explained in detail in the following sub-sections.

A. Generation of Data Matrix

Let $\{f_1, f_2, f_3, f_4, \dots, f_k, f_{k+1}, \dots, f_N\}$ be the number of frames in a video V . We consider the first k consecutive frames of the video stream to generate the data matrix D . The data matrix is of order $n \times k$ where n is the size of the image ($M \times N$) and k is the number of frames belonging to V considered for generating the data matrix D . An input image i.e., i^{th} frame of V , with $n = M \times N$ pixels can be treated as a point in an n -dimensional space called the image space. The individual coordinates of this n -dimensional point corresponds to the intensity values of each pixel of the image frame f_i and is arranged in the form of a column vector:

$$P_i = (p_1, p_2, p_3, \dots, p_n) \quad (1)$$

The column vector is formed by appending all the rows of an image together and taking the transpose.

Now we have k images each with n pixels. We can write our entire data set as an $n \times k$ data matrix. Each column of D represents one image of our k frames. So we have:

$$D = [P_1 \ P_2 \ P_3 \ \dots \ P_k] \quad (2)$$

where P_i is the column vector which denotes the intensity values of the i^{th} frame.

B. Computing the Covariance matrix

The purpose of computing the covariance matrix is to find the variations between the intensity levels of consecutive frames, the computed covariance matrix is used to derive the Eigen values. The conventional method followed and the proposed method (Modified method) for computing the covariance matrix is explained in detail.

1) Conventional method (CM)

Generally, the covariance matrix is computed by using the below formula (3).

$$C = U^T U / n - 1 \quad (3)$$

The first step is to move the origin to mean of the data. This is achieved by finding the mean image by averaging the columns of D , then subtract the mean image from each image in the data matrix (i.e., each column of D) to create the mean centered data vector which is U . The covariance matrix C is then computed by using Eq.(3). The matrix C will be of order k and hence we need to find k Eigen values of the matrix which are used for determining the shot boundary/keyframe.

However, in the proposed approach every time when a new frame is extracted from the video, we need to generate the data matrix by eliminating the first frame f_1 and appending the f_{k+1} frame to the data matrix D . Then again compute the covariance matrix for frames f_2 to f_{k+1} . Since each time the covariance matrix has to be computed, the time

complexity increases greatly. To overcome this, we dynamically update the covariance matrix by using the former values which is explained in detail in the next sub-section.

2) Modified method (MM)

A novel method for calculation of covariance matrix is implemented which avoids the recalculation of complete covariance matrix whenever a new frame is appended to the data matrix.

When a frame is appended and the first frame is eliminated from the data matrix, instead of calculating the complete covariance matrix, we reuse the already calculated covariance matrix by only calculating and appending the new covariance data to it and removing the first frame's covariance data.

For illustration let's consider $k = 3$ and f_x, f_y and f_z be the three consecutive frames of the video, whose data matrix is given by $D = [P_x \ P_y \ P_z]$ therefore covariance matrix of size 3×3 C is given by,

$$C = \begin{bmatrix} C_{xx} & C_{xy} & C_{xz} \\ C_{yx} & C_{yy} & C_{yz} \\ C_{zx} & C_{zy} & C_{zz} \end{bmatrix} \quad (4)$$

where

$$\begin{aligned} C_{xx} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (x_i - \mu_x)^2 \\ C_{xy} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (x_i - \mu_x)(y_i - \mu_y) \\ C_{xz} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (x_i - \mu_x)(z_i - \mu_z) \\ C_{yy} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (y_i - \mu_y)^2 \\ C_{yz} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (y_i - \mu_y)(z_i - \mu_z) \\ C_{yx} &= C_{xy}; \quad C_{zx} = C_{xz}; \quad C_{zy} = C_{yz}; \\ C_{zz} &= \left(\frac{1}{n} \right) \sum_{i=0}^n (z_i - \mu_z)^2 \end{aligned}$$

where μ_x, μ_y , and μ_z are the mean of f_x, f_y and f_z respectively.

Now when a new frame i.e., f_a is added and f_x is removed, we have to calculate the covariance matrix for f_y, f_z and f_a . Instead of calculating the whole new covariance matrix, we can use the previously calculated covariance data for f_y and f_z and adding the newly calculated covariance data for f_a to (4) as given in (5)

$$C = \begin{bmatrix} C_{yy} & C_{yz} & C_{ya} \\ C_{zy} & C_{zz} & C_{za} \\ C_{ay} & C_{az} & C_{aa} \end{bmatrix} \quad (5)$$

The advantage of using our method is the reduction in the computational time which avoids the recursive computation

of covariance matrix and thus increasing the flexibility of choosing larger value for k .

C. Calculating the Eigen Values

Let A be an $m \times m$ matrix. The eigenvalues of A are defined as the roots of

$$\text{Determinant } (A\lambda - I) = |A\lambda - I| = 0$$

where I is the $m \times m$ identity matrix. This equation is called the characteristic equation (or characteristic polynomial) and has m roots.

Let λ be an eigenvalue of A . Then there exists a vector x such that:

$$Ax = \lambda x$$

The vector x is called an eigenvector of A associated with the eigenvalue λ . Notice that there is no unique solution for x in the above equation. It is a direction vector only and can be scaled to any magnitude. To find a numerical solution for x we need to set one of its elements to an arbitrary value, say 1, which gives us a set of simultaneous equations to solve for the other elements. If there is no solution we repeat the process with another element. Ordinarily we normalize the final values so that x has length one, that is $x \cdot x^T = 1$.

Suppose we have a 3×3 matrix A with eigenvectors x_1, x_2, x_3 , and eigenvalues $\lambda_1, \lambda_2, \lambda_3$ so:

$$A x_1 = \lambda_1 x_1 \quad A x_2 = \lambda_2 x_2 \quad A x_3 = \lambda_3 x_3$$

Putting the eigenvectors as the columns of a matrix gives:

$$A[x_1 \ x_2 \ x_3] = [x_1 \ x_2 \ x_3] \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

writing:

$$\phi = [x_1 \ x_2 \ x_3] \text{ and } \Lambda = \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix}$$

where ϕ denotes Eigenvector and Λ denotes Eigen values,

which gives the matrix equation:

$$A \phi = \phi \Lambda$$

If the eigenvectors are normalized to unit magnitude, and they are orthogonal, then:

$$\phi \phi^T = \phi^T \phi = I$$

which means that:

$$\phi^T A \phi = \Lambda$$

and:

$$A = \phi \Lambda \phi^T \quad (6)$$

Now the above equation (6) can be implied in the proposed algorithm, since our covariance matrix C is $k \times k$, we can calculate the eigenvectors and eigenvalues of C using the standard technique outlined above. That is to say we solve for ϕ and Λ that satisfy:

$$C = \phi \Lambda \phi^T \quad (7)$$

The k Eigen values of the covariance matrix C is thus obtained from eq.(7).

D. Shot Boundary Detection / Keyframe Extraction

From the k Eigen values computed in Section III-C, the minimum Eigen value chosen is used to find the dissimilarities between the frames. If the minimum Eigen value exceeds the predefined threshold value then the $(k-1)^{th}$ frame will be the shot boundary and k^{th} frame is said to be a keyframe of the next shot indicating the beginning of the new shot.

If the k^{th} frame is not a keyframe then the data matrix is modified by eliminating the first frame and appending the next consecutive frame $(k+1)^{th}$ to the data matrix, else the data matrix is newly created with the consecutive frames starting from k^{th} frame to next k frames.

IV. EXPERIMENTAL RESULTS AND DISCUSSIONS

A. Precision and Recall

In order to reveal the accuracy and superiority of the proposed model, we have conducted an extensive experimentation on various videos. In order to evaluate the performance of the shot detection/keyframe extraction method presented in Section III, the recall and precision measures, were used [21], [22]. Let ' GT ' denote the ground truth set for the detection task under study, ' Det ' the detected (correctly or falsely) set using our methods. The Recall measure, also known as the true positive function or sensitivity, corresponds to the ratio of correct experimental detections over the number of all true detections:

$$Recall = \frac{|Det \cap GT|}{|GT|}$$

where $|GT|$ denotes the cardinality of set GT .

The Precision measure corresponds to the accuracy of the method considering false detections and it is defined as the number of correct experimental detections over the number of all experimental detections:

$$Precision = \frac{|Det \cap GT|}{|Det|}$$

TABLE I: GROUND TRUTH

Sequences	Frames	Cuts	Fades	Other Transitions
House Tour	664	4	-	-
Hasselt	2710	12	-	-
TRECVID 2002	4000	6	-	4
Mechelen Belgium	5015	34	-	7
House Tour	664	4	-	-

Table I shows the ground truth of the video sequences used for experimental purpose. The "House tour" video contains shots of living room and bedrooms which has only cuts. "Hasselt" video is an outdoor video showing different buildings and sculptures with fast panning and zooming of the camera. The video taken from TRECVID 2002 is "The

Saga of happy wanderers" which is also an outdoor video having cuts and other transitions. "Mechelen Belgium" video takes the tour of the city Mechelen of Belgium showing outdoor buildings, sculptures, statues and indoor of the church etc.

TABLE II: RESULTS OBTAINED USING THE PROPOSED APPROACH

Sequences	Cuts		Fades		Other transitions	
	Recall	Precision	Recall	Precision	Recall	Precision
House Tour	1	1	-	-	-	-
Hasselt	1	1	-	-	-	-
TRECVID 2002	0.83	1	-	-	0.25	1
Mechelen Belgium	0.97	1	-	-	0	0

The recall and precision values of the respective video sequences compared with the ground truth is shown in Table II and the keyframes of house tour sequence is shown in Fig. 1. The house tour and Hasselt sequences includes only abrupt cuts which were detected with high accuracy and the value of k was taken as 2. For TRECVID video sequence the experimental results were compared with the ground truth submitted by the NIST TRECVID 2002 [23]. Since there was sudden intensity variations and blurriness in the consecutive frames the value of k was chosen as 4 and the last smallest Eigen value was used for thresholding for TRECVID video sequence. The other transitions included dissolves and wipes which were detected with less accuracy. Most of the videos gave best results with $k = 2$ with the smallest Eigen value chosen for thresholding. The plot of Eigen values and frames for house tour and Hasselt sequences is shown in Fig. 2 and Fig. 3 respectively. The peaks indicate the keyframes of the video sequence.



Fig. 1. Keyframes of 'House Tour' video sequence.

B. Computational time for Covariance Matrix

Our modified method (MM) for calculation of covariance matrix is much faster than the conventional method (CM). Table III shows the total computational time in milliseconds (ms) for calculating the covariance matrix for the first and the second iteration.

In the conventional method, if $k = 4$, then starting from 1st

to 4th frames are used for calculating the covariance in the first iteration. For the next iteration the covariance matrix for 2nd frame to 5th frame is calculated. The process is continued until the keyframe is detected. This leads to recursive calculation of covariance data which results in higher computational time. In Table III the second column corresponds to computation time for conventional method in milliseconds. Ideally time taken for all the iterations in this method is equal.

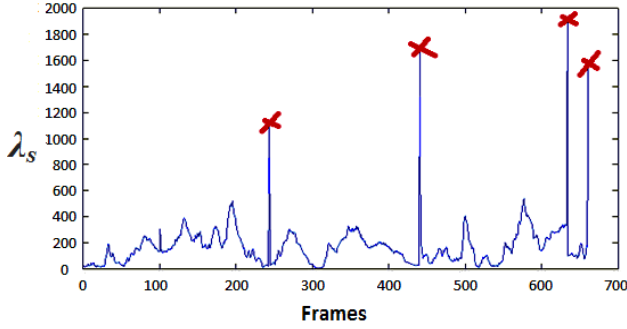


Fig. 2. Plot of Eigen values (λ_s) for 'House tour' video sequence with $k=2$.

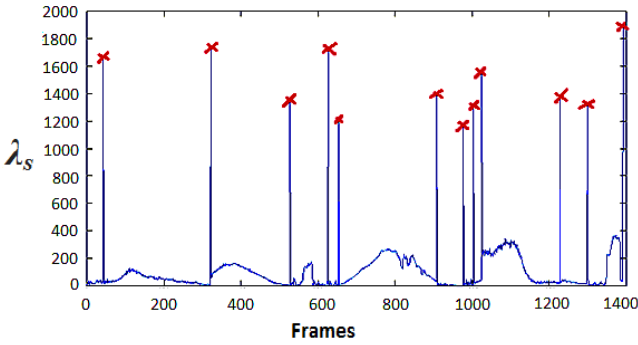


Fig. 3. Plot of Eigen values (λ_s) for 'Hasselt' video sequence with $k=2$. **X** indicating the keyframes.

$$T = t_1 + t_2$$

where, t_1 is the time taken for first iteration, t_2 is the time taken for second iteration and T is total time taken for first two iterations. Here $t_1 = t_2$.

In the modified method, if $k=4$, then starting from 1st to 4th frames are used for calculating the covariance in the first iteration. But for the second iteration, covariance matrix for 2nd frame to 5th frame is to be calculated, instead of calculating the whole new covariance matrix, we can use the previously calculated covariance data for 2nd, 3rd and 4th frame in the first iteration and calculate only 5th frame's covariance as explained in the Section III-B-2. So here we can call the iterations as updating steps. At each iteration, the covariance matrix is updated instead of recalculating it. Time taken from the second iteration onwards will be lesser than time taken for the first iteration. This reduces the computational time and makes advantageous over the conventional method.

$$T = t_1 + t_2$$

where, t_1 is the time taken for first iteration, t_2 is the time taken for second iteration (updating) and T is total time taken for first two iterations. Here $t_2 < t_1$.

Table III shows the value of T for different values of k for both the methods. For experimental purpose only two

iterations are recorded for the frame size of 240x320. As the iterations increases, the difference between the total time taken by conventional method and the modified method also increases. Fig. 4 shows the plot of T in milliseconds for conventional and modified method versus k .

TABLE III: COMPUTATIONAL TIME FOR DIFFERENT k VALUES

Frames(k)	CM in ms(T)	MM in ms(T)
2	6.751	2.66
4	23.395938	7.84432
6	53.339	14.699
8	99.747	24.982
10	170.537821	36.412233

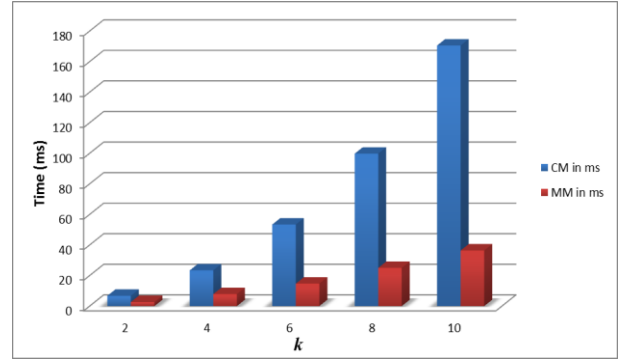


Fig. 4. Plot of computational time for CM and MM for different number of frames (k).

C. Sensitivity of k

If the value of k is smaller, it may detect even the smaller deviations between the frames but is easily prone to false detection due to noise in the frames. Whereas, if k is larger, it may not be able to detect the small deviations, but can overcome the false detection due to noise and blurriness of the image in the frames. The selection of k depends on the application and the quality of the images in the video.

V. CONCLUSION

In this paper a new approach for keyframe extraction/shot boundary detection is introduced. The method has been tested on various videos and is able to detect the abrupt cuts with high precision and accuracy. The results for shot detection are satisfactory and the speed of the proposed approach is quite fast to be applied to real-time surveillance systems and live videos as it is not dependent on future frames. Though the method fails to detect the fades and dissolves transitions the abrupt transitions are detected robustly. In our experiments, the threshold is set empirically but in future it can be made adaptive. In future, we also analyze all k Eigen values to detect the abrupt cuts, fades and dissolves to build a robust system which can detect any transitions with high accurate results.

REFERENCES

- [1] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, issue 1, pp. 10–28, 1993.
- [2] I. K. Sethi and N. Patel, "A statistical approach to scene change detection," in *Proc. SPIE*, 1995, pp. 329–338.

- [3] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Video parsing and browsing using compressed data," *Multimedia Tools and Applications*, vol. 1, issue 1, pp. 89–111, 1995.
- [4] B. Shahraray, "Scene change detection and content-based sampling of video sequences," in *Proc. SPIE Conference on Digital Video Compression: Algorithms and Technologies*, 1995, vol. 2419, pp. 2–13.
- [5] R. Zabih, J. Miller, and K. Mai, "Feature-based algorithms for detecting and classifying scene breaks," in *Proc. the Fourth ACM International Conference on Multimedia*, San Francisco, CA, 1995, pp. 189–200.
- [6] A. Hampapur, R. Jain, and T. Weymout, "Production model based digital video segmentation," *Multimedia Tools and Applications*, vol. 1, issue 1, pp. 9–46, 1995.
- [7] W. Zhao, J. Wang, D. Bhat, K. Sakiewicz, N. Nandhakumar, and W. Chang, "Improving color based video shot detection," in *Proc. IEEE International Conference on Multimedia Computing and Systems*, 1999, vol. 2, pp. 752–756.
- [8] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks," in *Proc. ACM Multimedia*, San Francisco, CA, 1993, vol. 95, pp. 189–200.
- [9] E. Sahouria and A. Zakhor, "Content analysis of video using principal components," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9, issue 8, pp. 1290–1298, 1999.
- [10] N. Haering, "A framework for the design of event detectors," Ph.D. Dissertation University of Central Florida, 1999.
- [11] S. Angadi, V. Naik, "Shot boundary detection and keyframe extraction for sports video summarization based on spectral entropy and mutual information," in *Proc. the Fourth International Conference on Signal and Image Processing (ICSIP)*, 2012, vol. 1, pp. 81–97.
- [12] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems*, vol. 1, issue 1, pp. 10–28, 1993.
- [13] R. Kasturi and R. Jain, *Dynamic vision, in computer vision: principles*, Washington: IEEE Computer Society Press, 1991.
- [14] S. Boreczky and L. A. Rowe, "Comparison of video shot boundary detection techniques," *Electronic Imaging*, vol. 5, issue 2, pp. 122–128, 1996.
- [15] D. S. Guru, M. Suhil, and P. Lolika, "A novel approach for shot boundary detection in videos," in *Proc. the First International Conference, ICMCCA, Series: Lecture Notes in Electrical Engineering*, 2012, vol. 213, pp. 209–220.
- [16] H. Xie, "Keyframe segmentation in video sequences – applied to reconstruction of 3D scene," Master Degree Dissertation in Electrical Engineering, Department of Technology, University of Kalmar, 2008.
- [17] P. Punitha and J. M. Jose, "Shot boundary detection based on Eigen coefficients and small Eigen value," in *Proc. Semantic Multimedia (SAMT), Springer Lecture Notes in Computer Science*, 2009, vol. 5887, pp. 126–136.
- [18] A. Yilmaz and M. A. Shah, "Shot detection using principal coordinate system," in *Proc. International Conference on Internet and Multimedia Systems and Applications*, 2000.
- [19] K. S. Ntalianis and S. D. Kollias, "An optimized key-frames extraction scheme based on SVD and correlation minimization," in *Proc. IEEE International Conference Multimedia and Expo (ICME)*, 2005, pp. 792–795.
- [20] K. J. Han and A. H. Tewfik, "Eigen-image based video segmentation and indexing," in *Proc. International Conference on Image Processing (ICIP)*, 1997, pp. 538–541.
- [21] R. R. Korfhage, *Information Storage and Retrieval*, New York: John Wiley and Sons, 1997.
- [22] K. Fukunaga, *Introduction to Statistical Pattern Recognition*, New York: Academic Press, 1990.
- [23] P. Over, A. Smeaton, and W. Kraaij. (2002). TREC video retrieval evaluation. [Online]. Available: <http://trecvid.nist.gov/trecvid.data.html#tv02>
- [24] S. Porter, M. Mirmehdi, and B. Thosmas, "Temporal video segmentation and classification of edit effects," *Image and Vision Computing*, vol. 21, pp. 1098–1106, 2003.
- [25] T. Y. Liu, K. T. Lo, X. D. Zhang, and J. Feng, "A new cut detection, algorithm with constant false alarm ratio for video segmentation," in *Proc. Visual Communications and Image*, vol. 15 pp. 132–144, 2004.
- [26] R. S. Jadon, S. Chaudhury, and K. K. Biswas, "A fuzzy theoretic approach for video segmentation using syntactic features," *Pattern Recognition Letters*, vol. 22, pp. 1359–1369, 2001.
- [27] C. Lo and S. Wang, "Video segmentation using a histogram based fuzzy c-means clustering algorithm," *Computing, Standards and Interface*, vol. 23 pp. 429–438, 2001.

- [28] A. Hanjalic, "Shot boundary detection: unravelled and resolved?," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, issue 2, pp. 90–105, 2002.
- [29] S. J. F. Guimaraes, M. Couprie, A. D. A. Araujo, and N. J. Leite, "Video segmentation based on 2D image analysis," *Pattern Recognition Letters*, vol. 24, pp. 947–957, 2003.
- [30] C. Huang and B. Liao, "A robust scene-change detection method for video segmentation," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, issue 12, pp. 1281–1288, 2001.



Vijeetkumar Benni has received B.E degree in electronics and communications from Visvesvaraya Technological University, India in 2011.

He is currently working as a lead engineer at Imaging Tech Labs, HCL Technologies, Bangalore, India. His research interests include image processing, computer vision, video analytics and robotics.



Dinesh R. is a senior development manager at Amazon, Bangalore, India. Prior to joining Amazon, he worked for companies HCL, LG, Honeywell, Dream to Reality, Seoul, South Korea. His areas of research interest includes Image processing, pattern recognition, computer vision and document image analysis.

He has over 50 publications to his credit at both international and national journals and conferences.

He also has 12 patents to his credit. He has successfully guided many academic and research projects. He is presently guiding 6 Ph.D. scholars.

Dr. Dinesh has served as program committee member for several national/international conferences. He has been identified as reviewer for several International Journals/conferences. He is a life member of IEEE, Computer Society of India and Society of Statistics, Compute and Applications.



Punitha P. received her BSc, MSc, and Ph.D. degrees in computer science from the University of Mysore in 1999, 2001 and 2006 respectively.

She currently works as a technical manager at Imaging Tech Labs, HCL Technologies, Bangalore and a Visiting Professor at PESIT, Bangalore. Previously, She served PESIT, Bangalore as a professor and the head of MCA, Postdoc RA at University of Glasgow, Scotland, Guest Faculty at University of Mysore, India. Her research interests include image processing, pattern recognition, computer vision, CBIR, video analytics, and intelligent vehicles. She has co-authored over 50 publications in International and National Journals and conference proceedings. She has guided many research projects at master's level and is currently guiding 5 PhD students. She has authored three books and is the editor for ICMCCA-2012 proceedings indexed by springer.

Dr. Punitha has been identified as an editorial board member of several journals/conferences and as a reviewer for top journals of IEEE Transactions, PRL and many conferences. She has been a convener of International conference and national seminars and also program/advisory/organizing committee member of many national/international conferences. She is a life member of IEEE, Computer Society of India and Society of Statistics, Computer and Applications.



Vasudeva Rao completed BSc and MSc in physics from Mangalore University in 1985 and graduated from Bhabha Atomic Research Centre (BARC) Training School in 1986.

He was a scientist in BARC till 1999 and his research interests included control systems, data acquisition systems, machine vision and robotics. Since then he is with HCL Technologies Ltd and at present heading imaging center of excellence as operations director for engineering and R&D services.