

# Analysis and Review of Formal Approaches to Automatic Video Shot Boundary Detection

Mr. Hattarge A.M.<sup>1</sup>, Prof. K.S. Thakre<sup>2</sup>

Department of IT, Sinhgad College of Engineering, Pune, India<sup>1</sup>

Associate Professor, Department of IT, Sinhgad College of Engineering, Pune, India<sup>2</sup>

**Abstract:** Today, large volume of digital videos has become available online to the masses. There is the drastic growth in the amount of the multimedia data, because of the improvements in data storage, acquisition and communication technologies, which are supported by improvements in processing of audio and video signals. People are looking for the videos that contain certain information of their interest. Such a search is facilitated by Content Based Video Retrieval methods. Three major steps of CBVR are, dividing the videos into the segments, extracting the features from video, and after that retrieving videos based on information in query. Out of these 3 steps segmentation of video is most prominent step as the retrieval results are based on the segmentation boundaries. Segmentation divides the video into shots. A shot is a segment of the video that consists of one continuous action in time domain. The boundaries of such shots can be detected by using various different techniques like histogram, discrete cosine transforms, motion vector, edge tracking, and block matching methods. But due to the motion of objects and camera these shot boundaries can be detected falsely. This paper presents a comparative study of different methods/algorithms that have been proposed in literature.

**Keywords:** Shot boundary detection, scene change, fades, dissolves, wipe, video retrieval.

## I. INTRODUCTION

As multimedia applications gained a huge popularity over the last decade, the need for their efficient management is increasing. Videos have become very popular in many areas such as communications, education and entertainment. A huge collection of video clips, live TV programs and movie pictures can be found on the Internet. Multimedia information indexing and retrieval are required to describe, store, and organize multimedia information and to assist people in finding multimedia resources conveniently and quickly. When people search for a particular video, a large database of videos is required to be searched for the results. This search is very complex due volume of data and diversity of data. This search can be speeded up by using content based video retrieval (CBVR). That means, the videos are searched according to the contents of the video. For this reason videos are needed to be segmented. So segmentation of videos is very important and crucial step in CBVR. Segmentation divides video into shots. A shot is a consecutive sequence of frames captured by a camera action that takes place between start and stop operations, which mark the shot boundaries [1]. There are strong content correlations between frames in a shot. Therefore, shots can be considered as the fundamental units to organize the contents of video sequences and the primitives for higher level semantic annotation and retrieval tasks. Shot boundaries are generated due to change in the scene (i.e. scene transitions). These transitions can be abrupt or gradual. Abrupt transitions are called as hard cuts or simply

Cuts. It is a hard boundary or clear cut which appears by a complete shot over a span of two serial frames. It is mainly used in live transmissions.

Gradual transitions are classified as

1. A *fade*: Two different kinds of fades are used: The fade-in and the fade-out. The fade-out emerges when the image fades to a black screen or a dot. The fade-in appears when the image is displayed from a black image. Both effects last a few frames.

2. A *dissolve*: It is a synchronous occurrence of a fade-in and a fade-out. The two effects are layered for a fixed period of time e.g. 0.5 seconds (12 frames). It is mainly used in live in-studio transmissions.

3. A *wipe*: This is a virtual line going across the screen clearing the old scene and displaying a new scene. It also occurs over more frames. It is commonly used in films such as *Star Wars* and TV shows

There are many approaches that have been proposed in literature to detect shot boundaries.

## II. LITERATURE SURVEY

The features used for shot boundary detection include color histogram [2] or block color histogram, edge change ratio, motion vectors [3], [4], together with more novel features such as scale invariant feature transform [5], corner points [6], information saliency map [7], etc. Color histograms are robust to small camera motion, but they are not able to differentiate the shots within the same scene, and they are



sensitive to large camera motions. Edge features are more invariant to illumination changes and motion than color histograms, and motion features can effectively handle the influence of object and camera motion. However, edge features and motion features as well as more complicated features cannot in general outperform the simple color histograms [8].

To measure similarity between frames using the extracted features is the second step required for shot boundary detection. Current similarity metrics for extracted feature vectors include the 1-norm cosine dissimilarity, the Euclidean distance, the histogram intersection, and the chi-squared similarity [9], [10], [11], as well as some novel similarity measures such as the earth mover's distance [2] and mutual information [12], [13], [14]. The similarity measures include pair-wise similarity measures that measure the similarities between consecutive frames and window similarity measures that measure similarities between frames within a window [15]. Window based similarity measures incorporate contextual information to reduce the influence of local noises or disturbances, but they need more computation than the pair-wise similarity measures. Using the measured similarities between frames, shot boundaries can be detected. Current shot boundary detection approaches can be classified into threshold-based and statistical learning-based.

#### A. Threshold-Based Approach:

This is the simplest method for segmentation. The threshold-based approach first measures the pair-wise similarities between frames and then compares it with a predefined threshold. When a similarity is less than the threshold, a boundary is detected. The threshold can be global, adaptive, or global and adaptive combined.

1) The global threshold-based algorithms use the same threshold, which is derived from observation or experiment, over the whole video, as in [16]. The major limitation of the global threshold-based algorithms is that local content variations are not effectively incorporated into the estimation of the global threshold, therefore influencing the boundary detection accuracy.

2) The adaptive threshold-based algorithms compute the threshold locally within a sliding window. Detection performance is often improved when an adaptive threshold is used instead of a global threshold. However, estimation of the adaptive threshold is more difficult than estimation of the global threshold and users are required to be more familiar with characteristics of videos in order to choose parameters such as the size of the sliding window.

3) Global and adaptive combined algorithms adjust local thresholds, taking into account the values of the global thresholds. Quenot et al. [17] define the thresholds for cut transition detection, dissolve transition detection, and flash detection as the functions of two global thresholds that are obtained from a tradeoff between recall and precision. Although this algorithm only needs to tune two global

thresholds, the values of the functions are changed locally. The limitation of this algorithm is that the functional relations between the two global thresholds and the locally adaptive thresholds are not easy to determine.

#### B. Statistical Learning-Based Approach:

The statistical learning-based approach regards shot boundary detection as a classification task in which frames are classified as shot change or no shot change depending on the features that they contain. Supervised learning and unsupervised learning are both used.

1) *Supervised learning-based classifiers:* The most commonly used supervised classifiers for shot boundary detection are the support vector machine (SVM) and Adaboost.

a) SVM [9], [18]: Chavez et al.[19] use the SVM as a two class classifier to separate cuts from non cuts. A kernel function is used to map the features into a high dimensional space in order to overcome the influence of changes in illumination and fast movement of objects. The SVM-based algorithms are widely used for shot boundary detection [20] because of their following merits.

i) They can fully utilize the training information and maintain good generalization.

ii) They can deal efficiently with a large number of features by the use of kernel functions.

iii) Many good SVM codes are readily available.

b) Adaboost: Herout et al.[21] make cut detection a pattern recognition task to which the Adaboost algorithm is applied. Zhao and Cai [3] apply the Adaboost algorithm to shot boundary detection in the compressed domain. The color and motion features are roughly classified first using a fuzzy classifier, and then each frame is classified as a cut, gradual, or no change frame using the Adaboost classifier. The main merit of the Adaboost boundary classifiers is that a large number of features can be handled: These classifiers select a part of features for boundary classification.

c) Others: Other supervised learning algorithms have been employed for shot boundary detection. For instance, Cooper et al. [15] use the binary k nearest-neighbor (kNN) classifier, where the similarities between frames within the particular temporal interval are used as its input.

The merits of the aforementioned supervised-learning approaches are that there is no need to set the thresholds used in the threshold-based approaches, and different types of features can be combined to improve the detection accuracy. The limitation is their heavy reliance on a well-chosen training set containing both positive and negative examples.

2. *Unsupervised learning-based algorithms:* The unsupervised learning-based shot boundary detection algorithms are classified into frame similarity-based and frame-based. The frame similarity-based algorithms cluster the measurements of similarity between pairs of frames into two clusters: the cluster with lower values of the similarities



corresponds to shot boundaries and the cluster with higher values of the similarities corresponds to non boundaries. Clustering algorithms such as K-means and fuzzy K-means have been used. The frame based algorithms treat each shot as a cluster of frames that have similar visual content. The merit of clustering-based approaches is that the training dataset is not needed. Their limitations are that temporal sequence progression information is not preserved, and they are inefficient in recognizing the different types of gradual transition. Shot boundary detection approaches can be classified into uncompressed domain-based and compressed domain-based. To avoid time-consuming video decompression, the features available in the compressed domain such as discrete cosine transform coefficients, DC image and MB types, and motion vectors can be directly employed for shot boundary detection [3]. However, the compressed domain-based approach is highly dependent on the compression standards, and it is less accurate than the uncompressed domain-based approach. Recently, the detection of gradual transitions has received more attention. Priyadarshinee Adhikari, *et al.* [22] made use of color histogram metrics. The difference between the histograms of two consecutive frames is evaluated resulting in the metrics. Further scaling of the metrics is performed using some log function to avoid ambiguity and to enable the choice of apt threshold for any type of videos which involves minor error due to flashlight, camera motion, etc. To extract robust frame difference from consecutive frames, they used verified  $\chi^2$  test which shows good performance comparing existing histogram based algorithm. Color histogram comparison ( $d_{r,g,b}(f_i, f_j)$ ) is calculated by histogram comparison of each color space of adjacent two frame ( $f_i, f_j$ ) and it is defined as,

$$d_{r,g,b}(f_i, f_j) = (|H_i^r(k) - H_j^r(k)| + |H_i^g(k) - H_j^g(k)| + |H_i^b(k) - H_j^b(k)|) \quad (1)$$

$H_i^r(k), H_i^g(k), H_i^b(k)$  represents the number (N) of bean (k) of each color space (r,g,b) in i frame  $f_i$ . Performance of this method is good but this method cannot be used with compressed video. This approach fails to identify the shot boundaries for a video which is compressed.

Lenka Krulikovská, *et al.* [23], proposed a fast algorithm for shot cut detection in which frames are compared and the cut position is determined according to their similarity evaluated by selected measure and threshold and decide if the frames are within one shot or not based on their similarity. The frame set as actual is compared to the frame distant by a defined step. They decide if the frames are within one shot or not based on their similarity. If frames are within one shot, the distant frame is set as new actual frame. If the compared frames belong to different shots, the procedure for searching the position of shot change starts. Most of the current methods employ frame by frame comparison, where pairs of successive frames are compared. Therefore these methods are highly time demanding and computationally complex. Lenka Krulikovská, *et al.* solved this problem by

comparing the frame set to a frame distant by a defined step. But this method has the limitations like it cannot be used in real time applications. In this method measure with known range of values has to be used for the evaluation of similarity of compared frames, which is difficult and this method also does not work well to detect gradual transitions.

Krishna K Warhade, *et al.* [24], presented a work in which they have used an Algorithm that uses dual tree complex wavelength transform followed by spatial domain structural similarity algorithm in the presence of motion and tested results of algorithm against various metrics. They observed that the algorithm performed better than traditional metrics like likelihood ratio and histograms in terms of improved Recall, precision, and F1 measure. They used recall, and precision metrics for evaluation of shot detection algorithms. Where recall is defined as

$$R = \frac{C}{C + M} = \frac{C}{D} \quad (2)$$

Where as precision is defined as,

$$R = \frac{C}{C + FP} \quad (3)$$

Where D is the total number of shot boundaries in the test video sequence,

C is the number of shot boundaries correctly detected by algorithm,

M is the number of shot boundaries missed by algorithm,

And FP is the false positives detected by the algorithm.

Also to rank the performance of different algorithms, F1 measure have been used i.e., harmonic average of Recall and precision and is defined as,

$$F1(R, P) = \frac{2 \times R \times P}{R + P} \quad (4)$$

As natural image signals are highly structured and their pixels exhibit strong dependencies, especially when they are spatially proximate. These dependencies carry important information about the structure of the objects in the visual scene. The spatial domain structural similarity (SSIM) algorithm has been proposed by Wang *et al.* [25]. The structural information in an image represents the structure of the object in the scene which is independent of average luminance and contrast.

Hence they propose and explore possibility of SSIM as shot boundary detection metric. The SSIM index between consecutive frames are obtained by,

$$SSIM(i, i+1) = \frac{(2\mu_i\mu_{i+1} + C_1)(2\sigma_{i,i+1} + C_2)}{(\mu_i^2 + \mu_{i+1}^2 + C_1)(\sigma_i^2 + \sigma_{i+1}^2 + C_2)} \quad (5)$$

$\sigma_{i,i+1}$ , For  $1 \leq i \leq N-1$ , where  $\mu_i$  and  $\mu_{i+1}$  are the mean of the structure feature of a current frame and next consecutive frame, respectively,  $\sigma_i$  and  $\sigma_{i+1}$  are the standard deviation of structure feature of the current frame and next consecutive frame, respectively,  $C_1$  and  $C_2$  are small constants to avoid instability. The value of is obtained by,

$$\sigma_{i,i+1} = \frac{1}{PQ} \sum_{x=1}^P \sum_{y=1}^Q \{(SF[x, y, i] - \mu_i)(SF[x, y, i+1] - \mu_{i+1})\} \quad (6)$$

But their algorithm does not work to eliminate the disturbances due to illumination and fast camera motion in YUV color space. As well this algorithm fails to differentiate between gradual transitions and motions.

In another research Z. Li, J. Jiang, *et al.* [26] proposed An Effective and Fast Scene Change Detection Algorithm for MPEG Compressed Videos. This algorithm exploits the MPEG motion estimation and compensation scheme by examining the prediction status for each macro-block inside B frames and P frames. As a result, locating both abrupt and dissolved scene changes is operated by a sequence of comparison tests, and no feature extraction or histogram differentiation is needed. This improves the speed of algorithm and precision & recall rate but the algorithm is very complex. Anastasios Dimou, *et al.* [27] presented a method for Scene Change Detection for H.264 codec. This method uses dynamic threshold technique and is based on the extraction of the sum of absolute differences between consecutive frames from the H.264 codec. Macro blocks in H.264 are further tiled in smaller blocks. Each block can be compared with the respective block in the previous frame. As a similarity metric Sum of Absolute Differences (SAD) is used:

$$SAD(n) = \sum_{i=0}^{N-1} \sum_{j=0}^{M-1} |F_n(i, j) - F_{n-1}(i, j)| \quad (7)$$

Where  $F_n$  is the  $n^{\text{th}}$  frame of size  $N \times M$ ,  $i$  and  $j$  denotes the pixel coordinates.

SAD values are further stored and used to evaluate the best compression scheme. The encoder can use a pre-defined number of previous frames as a reference to make the best decision. These differences serve as a criterion for the choice of the compression method as well as for the temporal prediction. It uses a sliding window to extract local statistical properties (mean value, standard deviation) which is further used to define a continuously updating automated threshold. Use of only previous frames for the detection is the major drawback of this method and also it fails to detect gradual transitions.

Amudha J, *et al.* [28] proposed a less complex algorithm for detecting shot boundaries. Algorithm is less complex as Shot boundaries are detected by comparing measures obtained from the saliency regions of frames. Video sequence is divided into frames and each frame is given to the visual attention model which outputs a saliency map. All the consecutive saliency maps from the frames are compared with two statistical metrics mean and variance. Shots are identified based on the threshold values and further analysis on the patterns of gradual transitions has been studied. The architecture of the proposed method is given in fig.1,

Shu-Ching Chen, *et al.* [29] proposed a new method for scene change detection using unsupervised segmentation algorithm and object tracking technique. The seed idea is to compare the segmentation mask maps between two

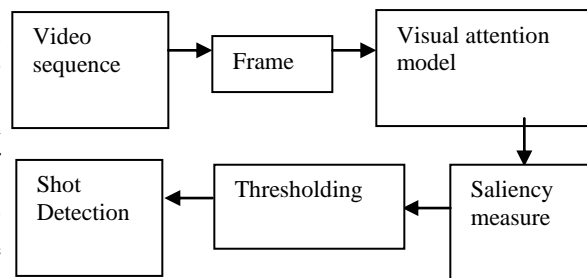


Fig 1: Architecture

successive video frames. In addition, the object tracking technique is employed as a complement to handle the situations of scene rotation without any extra overhead. In this algorithm, the partition and the class parameters are treated as random variables. The method for partitioning a video frame starts with a random partition and employs an iterative algorithm to estimate the partition and the class parameters jointly

The advantages of using unsupervised segmentation are:

- It is fully unsupervised, without any user interactions.
- The algorithm for comparing two frames is simple and fast.
- The object level segmentation results can be further used for video indexing and content analysis.

But this method cannot be used for compressed videos.

Young-Min Kim, *et al.* [30] proposed an algorithm for fast scene change detection using direct feature, extraction from mpeg compressed videos. They divided this algorithm as into direct edge information extraction and scene change detection through matching between two consecutive frames. This algorithm based on mathematical formulation which extracts edge information directly from MPEG video data using the relation of AC coefficients and orientation histogram for frame matching. As like many other algorithms this algorithm also fails to detect gradual transitions. Vasileios T. *et al.*[31] proposed work for scene change detection. In this method the shots are clustered into groups based only on their visual similarity and a label is assigned to each shot according to the group that it belongs to. Then, a sequence alignment algorithm is applied to detect when the pattern of shot labels changes, providing the final scene segmentation result. In this way shot similarity is computed based only on visual features, while ordering of shots is taken into account during sequence alignment. This method does not work if visual contents of shot in a scene changes continuously.

### III. PROPOSED ALGORITHM

We have analyzed that most of the algorithms developed for Automatic Shot Boundary Detection are not adaptive and they are giving more efficient outputs for only few specific



video inputs. Shot Boundary Detection can be improved if we will find an adaptive algorithm which will extract the different features of an image and using them calculates a unique feature which can distinctively define the image. By comparing the values of it for consecutive images we can distinguish the cuts, fades and dissolves.

Such a unique feature can be calculated by considering the following individual features for each frame to identify the abrupt and gradual transitions.

- **HSV Histogram:** A histogram is nothing but a function that calculates the number of observations which can be then divided into disjoint categories known as bins. HSV histogram can be calculated for individual H, S, V components. We can also calculate the cumulative histogram and use as important feature.

- **Edge Change Ratio:** The ECR attempts to compare the actual contents of two frames. It transforms both frames to edge pictures, i.e. it extracts the probable outlines of the objects within the pictures. Afterwards it compares these edge pictures using dilations to compute the probability that the second frame contains the same object as the first frame.

- **iii. Discrete Cosine Transform:** The discrete cosine transform (DCT) represents an image as a sum of sinusoids of varying magnitudes and frequencies. We can convert the normalize frame into gray color space, it's equally divided into non-overlapped 8X8 blocks. The DCT operates on an X block of N x N image samples and creates Y, which is an N x N block of coefficients. The DCT can be described in terms of a transform matrix A. The forward DCT is given by  $Y = AXA^T$ , where X is a matrix of samples, Y is a matrix of coefficients, and A is an N x N transform matrix.

The CUT or abrupt transition can be identified by calculating the difference index between two consecutive frames by using the formula

$$\text{DifInd} = \frac{DCTc}{ECR + HSVHistc + Grayc + GrayHistco} \quad (8)$$

Here, DCTc, Histogramc and Greyc are the correlations for two consecutive frames.

For this difference index adaptive threshold is calculated using the following method.

1. Perform division of frames in fix group size (i.e. Clustering).
2. Mean & standard deviation can be calculated for each such group of the DifInd.
3. Calculate the threshold value for CUT detection using following equation.

Threshold = mean + a \* standard deviation

Where, 'a' is an integer constant.

4. If for a particular frame difference index is greater than threshold announce it as CUT.

To detect the fade in/out effect the Entropy can be used as a feature. Fade is generated or ends in a monotone image.

Entropy is a statistical measure of randomness of the image.

Entropy is defined as

$$E = -\sum (p * \log_2 p)$$

Where, p represents the histogram count.

Dissolves are generated by linear combination of two different shots. Dissolves can be identified using following steps.

**Step 1:** Take the frame between two consecutive CUTS

**Step 2:** Find different index for each frame using equation (8)

**Step 3:** If a DifInd value is less than threshold that frame is possible candidate for dissolve effect.

Find first and last frame which show such characteristics.

**Step 4:** Find ECR values for the first frame to last frame of above step.

**Step 5:** If ECR value is greater than particular threshold then all those frames are having dissolve effect

#### IV. CONCLUSION

In this paper we have presented a review on different algorithms and techniques that have been proposed for video shot boundary detection. There are many algorithms proposed, but different algorithms work differently in different situations. Some algorithms are limited to particular color space; some of them fail to identify gradual transitions, some algorithms work for compressed videos while some works only for uncompressed videos. So we have proposed an adaptive approach that combines different features of video and calculates a unique feature which can be used to identify the hard cuts and gradual transitions both.

#### REFERENCES

- [1] C. H. Yeo, Y. W. Zhu, Q. B. Sun, and S. F. Chang, "A Framework for sub-window shot detection," in Proc. Int. Multimedia Modelling Conf., Jan. 2005, pp. 84-91
- [2] C. H. Hoi, L. S. Wong, and A. Lyu, "Chinese university of Hong Kong at TRECVID 2006: Shot boundary detection and video search," in Proc. TREC Video Retrieval Eval., 2006. Available: [http://www.nlp.ir.nist.gov/projects/tvpubs/tv6.papers/chines\\_e\\_uhk.pdf](http://www.nlp.ir.nist.gov/projects/tvpubs/tv6.papers/chines_e_uhk.pdf)
- [3] Z.-C. Zhao and A.-N. Cai, "Shot boundary detection algorithm in compressed domain based on adaboost and fuzzy theory," in Proc. Int. Conf. Nat. Comput., 2006, pp. 617-626.
- [4] S. V. Porter, "Video segmentation and indexing using motion estimation," Ph.D. dissertation, Dept. Comput. Sci., Univ. Bristol, Bristol, U.K., 2004.
- [5] Y. Chang, D. J. Lee, Y. Hong, and J. Archibald, "Unsupervised video shot detection using clustering ensemble with a color global scale invariant feature transform descriptor," EURASIP J. Image Video Process., vol. 2008, pp. 1-10, 2008.
- [6] X. B. Gao, J. Li, and Y. Shi, "A video shot boundary detection algorithm based on feature tracking," in Proc. Int. Conf. Rough Sets Knowl. Technol. (Lect. Notes Comput. Sci.), 4062, 2006, pp. 651-658.
- [7] X. Wu, P. C. Yuan, C. Liu, and J. Huang, "Shot boundary detection: An information saliency approach," in Proc. Congr. Image Signal Process., 2008, vol. 2, pp. 808-812.
- [8] Yuan, H. Wang, L. Xiao, W. Zheng, J. Li, F. Lin, and B. Zhang, "A formal study of shot boundary detection," IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 2, pp. 168-186, Feb. 2007.

- [9] G. Camara-Chavez, F. Precioso, M. Cord, S. Phillip-Foliguet, and A. de A. Araujo, "Shot boundary detection by a hierarchical supervised approach," in Proc. Int. Conf. Syst., Signals Image Process., Jun. 2007, pp. 197–200.
- [10] H. Lu, Y.-P. Tan, X. Xue, and L. Wu, "Shot boundary detection using unsupervised clustering and hypothesis testing," in Proc. Int. Conf. Commun. Circuits Syst., Jun. 2004, vol. 2, pp. 932–936.
- [11] C. Choudary and T. C. Liu, "Summarization of visual content in instructional videos," IEEE Trans. Multimedia, vol. 9, no. 7, pp. 1443–1455, Nov. 2007.
- [12] L. Bai, S.-Y. Lao, H.-T. Liu, and J. Bu, "Video shot boundary detection using petri-net," in Proc. Int. Conf. Mach. Learning Cybern., 2008, pp. 3047–3051.
- [13] C. Liu, H. Liu, S. Jiang, Q. Huang, Y. Zheng, and W. Zhang, "JDL at TRECVID 2006 shot boundary detection," in Proc. TREC Video Retrieval Eval. Workshop, 2006. Available: [http://www.nlpir.nist.gov/projects/tvpubs/tv6.papers/cas\\_jdl.pdf](http://www.nlpir.nist.gov/projects/tvpubs/tv6.papers/cas_jdl.pdf)
- [14] D. Xia, X. Deng, and Q. Zeng, "Shot boundary detection based on difference sequences of mutual information," in Proc. Int. Conf. Image Graph., Aug. 2007, pp. 389–394.
- [15] M. Cooper, T. Liu, and E. Rieffel, "Video segmentation via temporal pattern classification," IEEE Trans. Multimedia, vol. 9, no. 3, pp. 610–618, Apr. 2007.
- [16] Z. Cernekova, I. Pitas, and C. Nikou, "Information theory-based shot cut/fade detection and video summarization," IEEE Trans. Circuits Syst. Video Technol., vol. 16, no. 1, pp. 82–90, Jan. 2006.
- [17] G. Quenot, D. Moraru, and L. Besacier. (2003). "CLIPS at TRECVID: Shot boundary detection and feature detection," in Proc. TREC Video Retrieval Eval. Workshop Notebook Papers [Online]. Available: <http://www.nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2003>.
- [18] K. Matsumoto, M. Naito, K. Hoashi, and F. Sugaya, "SVM-based shot boundary detection with a novel feature," in Proc. IEEE Int. Conf. Multimedia Expo., Jul. 2006, pp. 1837–1840.
- [19] G. C. Chavez, F. Precioso, M. Cord, S. P. -Foliguet, and A. de A. Araujo, "Shot boundary detection at TRECVID 2006," in Proc. TREC Video Retrieval Eval., 2006. Available: <http://www.nlpir.nist.gov/projects/tvpubs/tv6.papers/dokuz.pdf>
- [20] P. Over, T. laneva, W. Kraaij, and A. F. Smeaton. (2005). "TRECVID 2005—An overview," in Proc. TREC Video Retrieval Eval. Workshop.[Online]. Available: <http://www.nlpir.nist.gov/projects/tvpubs/tv.pubs.org.html#2005>.
- [21] A. Herout, V. Beran, M. Hradis, I. Potucek, P. Zemečik, and P. Chmelar, "TRECVID 2007 by the Brno Group," in Proc. TREC Video Retrieval Eval., 2007. Available: <http://www.nlpir.nist.gov/projects/tvpubs/tv7.papers/brno.pdf>
- [22] Priyadarshinee Adhikari, Neeta Gargote, Jyothi Digge, and B.G. Hogade, "Abrupt Scene Change Detection," in B.G. 2008, World Academy of Science, Engineering and Technology 42 2008, pp: 711-716.
- [23] Krulikovská, Jaroslav Polec, Tomáš Hirner, "Fast Algorithm of Shot Cut Detection," in World Academy of Science, Engineering and Technology 67 2012.
- [24] Krishna K Warhade, Shabbier N. Merchant and U.B.Desai, "Performance evaluation of shot boundary detection metrics in the presence of object and camera motion," in IETE Journal of research, vol 57, issue 5, sep oct 2012.
- [25] Z Wang, A C Bovik, H R Sheikh and E P simoncelli, "image quality assessment: from error visibility to structural similarity", IEEE transactions on image processing, vol. 13 No. 4 pp. 600-12, April 2004.
- [26] Z. Li, J. Jiang, G. Xiao, and H. Fang, "An Effective and Fast Scene Change Detection Algorithm for MPEG Compressed Videos," in H. 2006, ICIAR 2006, LNCS 4141, pp. 206-214.
- [27] Anastasios Dimou, Olivia Nemethova and Markus Rupp, "Scene Change Detection for H.264 Using Dynamic Threshold Techniques," in 2005, Proceedings of 5th EURASIP Conference on Speech and Image Processing, Multimedia Communications and Service.
- [28] Amudha J, Radha D, Naresh Kumar P, "Video Shot Detection using Saliency Measure," in International Journal of Computer Applications (0975 – 8887) Volume 45– No.2, May 2012.
- [29] Shu-Ching Chen, Mei-Ling Shyu, Cheng-Cui Zhang, R. L. Kashyap, "Video Scene change Detection method using Unsupervised Segmentation and object Tracking," in R. L.: 2001, Multimedia and Expo, 2001, ICME 2001, IEEE International Conference on, 22-25 AUG. 2001, pp. 56-59.
- [30] Young-Min Kim, Sung Woo Choi and Seong, "Fast Scene Change Detection using Direct Feature, Extraction from MPEG Compressed Videos," in IEEE Transactions on multimedia, Dec. 2000, No. 2, issue.4, pp: 240-254, 2000.
- [31] Vasileios T. Chasanis, Aristidis C. Likas, and Nikolaos P. Galatsanos, "Scene Detection in Videos Using Shot Clustering and Sequence Alignment," in IEEE transactions on multimedia, vol. 11, no. 1, january 2009.