Journal of Applied Sciences 6 (8): 1679-1685, 2006
ISSN 1812-5654

Real-Time Algorithms for Video Summarization

[1,2]L. Boussaid, [1,3]A. Mtibaa, [1,2]M. Abid and [4]M. Paindavoine
[1]Unité de Recherche CES, [2]Ecole Nationale d'Ingénieurs de Sfax,
[3]Ecole Nationale d'Ingénieurs de Monastir, [4]Laboratoire LE2i, Tunisia

Abstract: An important number of methods for scene break detection and automatic video abstracting have been investigated in recent years. These methods have to detect two kinds of scene changes known as abrupt and gradual transition. The present study we present two old and reliable algorithms on which we brought modifications in order to reduce complexity while preserving the same accuracy and satisfy the real-time constraints of new multimedia applications.

Key words: Scene change detection, cut, dissolve, local histogram, edge change ratio, real-time

INTRODUCTION

Nowadays, digital video are widely used in many applications domain and as consequence immense opportunities are being created in the field of video analysis and video content description researches.

Applications such as interactive TV, video-on-demand, automated industry inspection and digital libraries constitute the center of interest of the recent researches.

On the other hand, with the advance in TV broadcasting, data compression, storage and networking technology, the amount of video data has grown enormously in the last years. This important amount of audiovisual material has to be effectively organized and managed in order to insure a rapid retrieval and reuse. Since manual video annotation is monotonous and very expensive, automatic video segmentation is required.

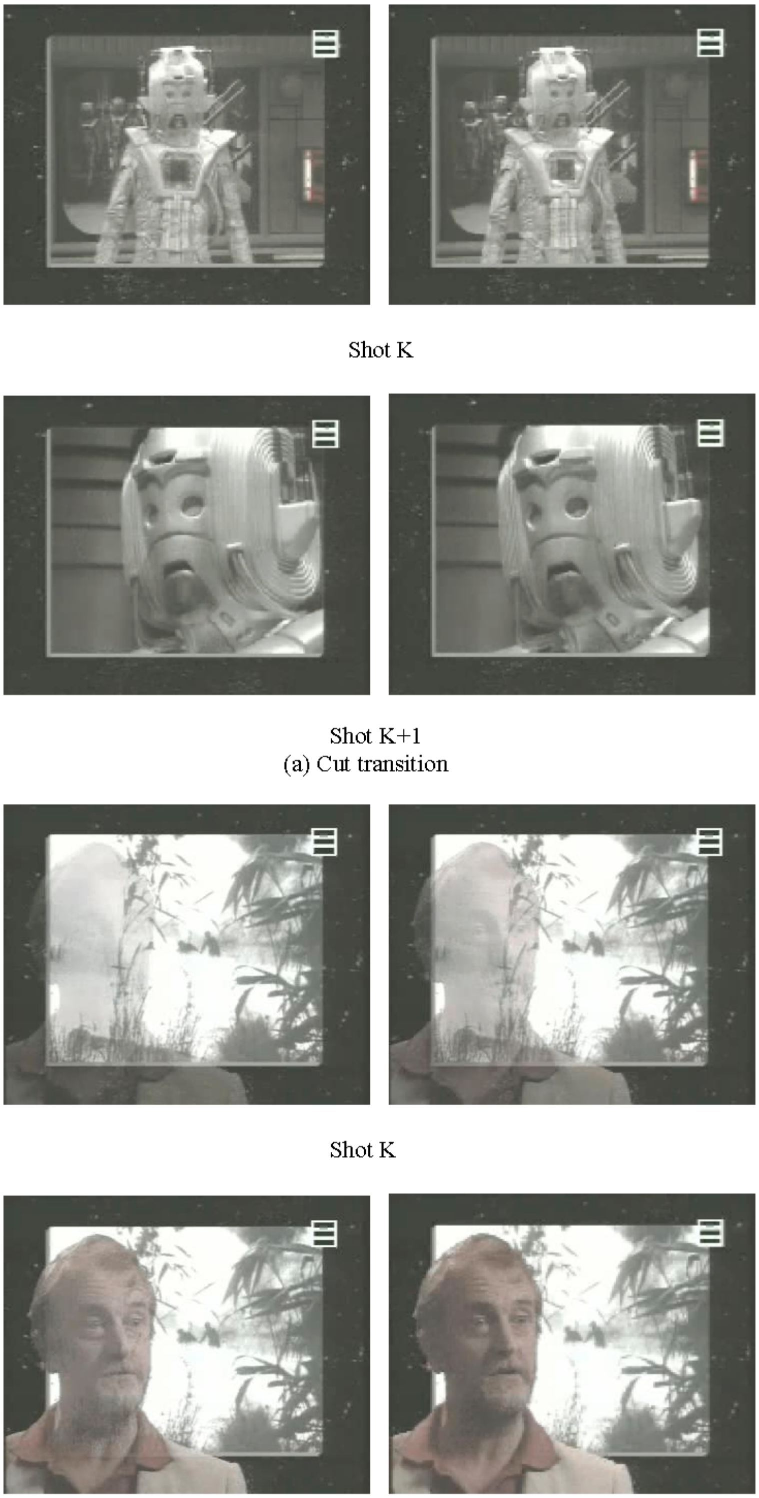
The main purpose of the recent studies on video analysis and content description is to provide high level semantic content from low level features. However, before any manipulation on audiovisual documents, hierarchical structure must be determined.

In this way, a standard hierarchical video model was defined. This model is composed of some elementary units as scenes, shots and frames. In this structure a shot is defined as an unbroken sequence of frames from a single camera, whereas a scene is a set of shots with semantic link, location unit and action unit (Lienhart *et al*., 1998).

In produced video such as television or movies, shots are separated by different types of transitions, or boundaries. Although well-known video editing programs such as Adobe Premiere or Ulead Media Studio provide more than 100 different types of edits, we classify in general transition effects into two categories (Lienhart and Jan, 1999). The simplest transition is a cut, an abrupt shot change that occurs between two consecutive frames. Gradual transition such as fades and dissolves are more complex. Shot boundaries are fades when the frames of the shot gradually change from or to black and can be dissolves when the frames of the first shot are gradually morphed into the frames of the second (Porter *et al*., 2001). Figure 1 shows an example of transition effects.

Most of the existing methods of video segmentation have to challenge the difficulty of finding shot boundaries in the presence of camera or object motion and illumination variations which can lead to false detection. In other cases, frames that have different structures but similar color distributions can give a missed detection (Mas *et al*., 2003). The study of the state of the art shows that several methods for shot boundary detection were proposed. These methods can operate in different environments such as temporal, frequency, uncompressed and compressed domains. Lefèvre *et al*. (2003) distinguish two classes of techniques: Those which could be done off-line and have high complexity and others which are dedicated for real-time applications. Since we are interested in hardware implementation of automatic video segmentation for real-time applications (Boussaid *et al*., 2004, 2005a), we have focused our work on methods that offer good results and perform minimal computation.

Corresponding Author: L. Boussaid, Unité de Recherche CES, Tunisia



Shot K

Shot K+1
(a) Cut transition

Shot K

Shot K+1
(b) Dissolve transition

Fig. 1: Example of transition effects

SHOT BOUNDARY DETECTION: STATE-OF-THE-ART

An important variety of shot boundary detection algorithms was proposed in the last decade. The study of the current state of the art shows that we can classify these algorithms into three generations. The first generation concerns methods witch measure distance of similarity between adjacent frames by using elementary feature extraction such as pixel differences, global and local histogram differences, motion compensated pixel differences and DCT coefficient differences (Ardizzone and Cascia, 1997; Abdel-Mottaleb *et al*., 1996; Yu and Wolf, 1997; Arman *et al*., 1993). In the second generation, techniques were developed by combining shot boundary detection algorithms (e.g., by using audio and video features) (Taskiran and Delp, 1998; JungHwan and Hua, 2000; Wactlar *et al*., 1999). Although these techniques have brought improvement to the quality of shot change detection, they increased complexity and computational time. The most recent algorithms have introduced intelligent algorithms such as fuzzy approaches and those based on neural network (Jadon *et al*., 2001; Xiang and Suganthan, 2002).

For real-time segmentation of video, (Dailianas *et al*., 1995) has evaluated complexity of many methods by estimating the number of operations when measuring dissimilarity between two consecutive frames. In this way, he has used an assumption that addition, subtraction and multiplication require time equivalent to one operation, whereas divisions take approximately four times more. Implementation issue such as assignment of variables to registers, use of pointers and arrays, memory access time and others, are ignored.

SCENE BREAK DETECTION ALGORITHMS

The study of the state of the art shows that histogram-based methods are the most suitable for cut detection whereas edge-based approaches are the most reliable for dissolve detection. The study presents the specifications of two efficient methods based on local histograms and edge change ratio.

Local histograms-based algorithm: Local histogram was proposed by Nagasaka *et al*. (1992). They divided each frame into 16 blocks and computed local histograms before evaluating a difference metric. Local histogram based method has been shown to perform well for shot cut detection.

Let H (f,k) be the value of the histogram for frame f and for the discrete value of the intensity k. The value of k is in the range [0,N-1], where N is the number of discrete values a pixel can have. The metric of the dissimilarity between consecutive frames f_i and f_{i+1} is given by:

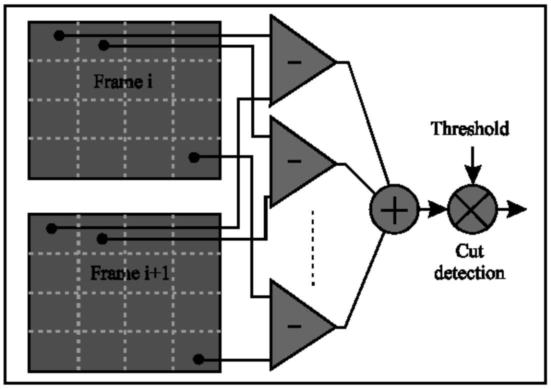


Fig. 2: Cut detection principle

$$D(f_i,f_{i+1}) = \sum_{c\in RGB}\sum_{b=1}^{16}\sum_{j=0}^{N-1} |H(f_i,c,b,j)-H(f_{i+1},c,b,j)| \qquad (1)$$

Where c is the luminance of red, green and blue components of the picture and b represents the number of blocks.

To detect break shots, the metric $D(f_i,f_{i+1})$ is compared to a global threshold. When this metric exceeds the value of threshold, it indicates that a shot transition has occurred. The principle of cut detection based on local histograms approach is shown in Fig. 2.

Edge change ratio based algorithm: During a fade-out object edges gradually disappear, while during a fade-in they gradually show up. One measure of the change of contours is the Edge Change Ratio (ECR) proposed by Zabih *et al.* (1995). During a fade-in $ECR^{In} >> ECR^{Out}$ and the reverse $ECR^{Out} >> ECR^{In}$ is true during a fade-out. The ECR approach which is described in Fig. 3 consists of the following steps:

- Compute edge detection for two consecutive frames by using the Canny edge detector;
- Count number of edge pixels;
- Dilate and invert edges;
- Compute pixels out and pixels in;
- Count pixels out and pixels in;
- And compute the Edge Change Ratio (ECR) which is defined as the maximum between pixels out and pixels in.

This approach suffers of being sensitive to object motion, text appearance and zooming.

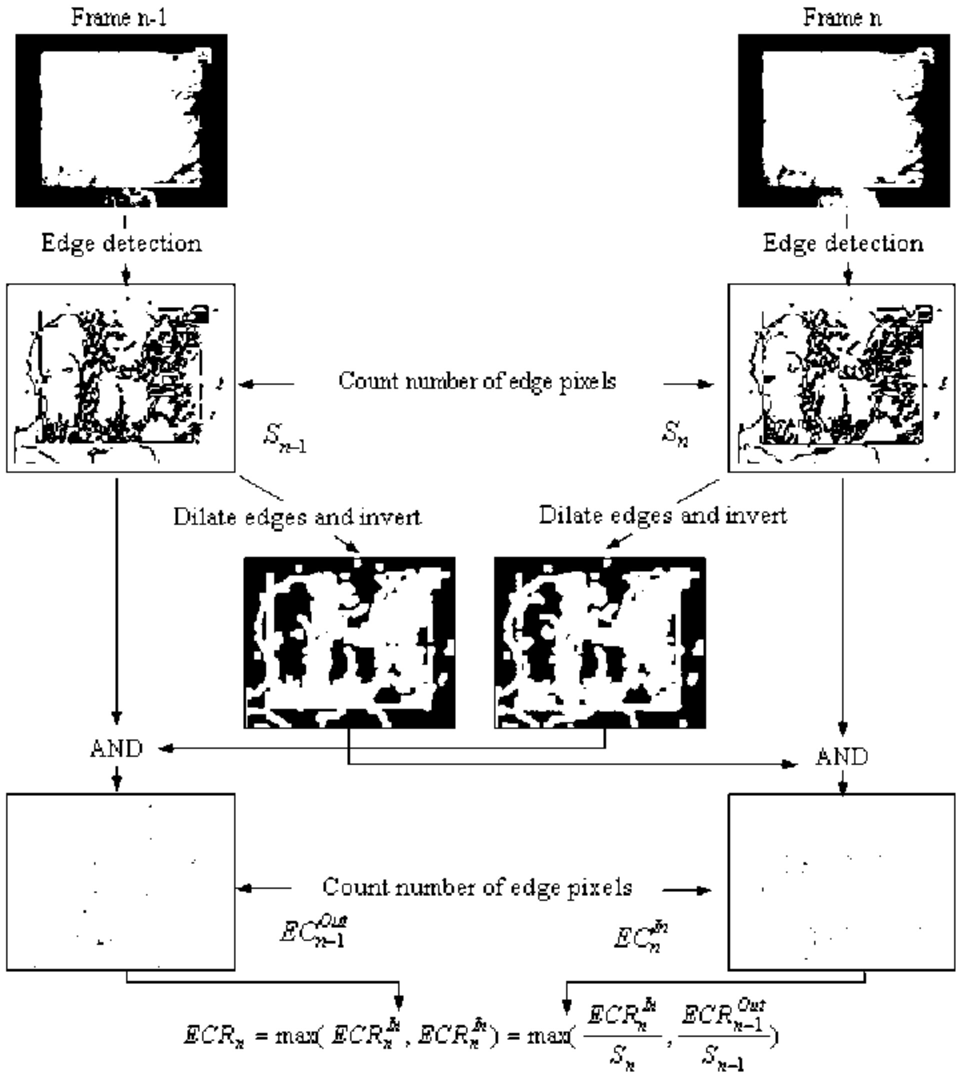

Frame n-1
Frame n
Edge detection
Edge detection
Count number of edge pixels
S_{n-1}
S_n
Dilate edges and invert
Dilate edges and invert
AND
AND
Count number of edge pixels
EC_{n-1}^{Out}
EC_n^{In}
$ECR_n = \max(ECR_n^{In}, ECR_n^{In}) = \max(\frac{ECR_n^{In}}{S_n}, \frac{ECR_{n-1}^{Out}}{S_{n-1}})$

Fig. 3: Calculation graph of the Edge Change Ratio (ECR)

EXPERIMENTAL RESULTS

Cut detection: Along the experiments done in (Boussaid *et al.*, 2005b), The present study tried to reduce complexity and speed up the algorithm while preserving the same accuracy.

To do so, local histograms approach have performed on a set of video sequences, in different color spaces, different types of quantization and different formats of sub-sampling. These experiments have shown that working in the gray space and uniform quantization at 4 levels (bins) presents reliable results and relatively low computation time. Eq. 1 becomes:

$$D(f_n,f_{n+1}) = \sum_{b=1}^{16}\sum_{j=0}^{3} |H(f_{n+1},b,j)-H(f_n,b,j)| \qquad (2)$$

Although this modified approach performs well and reliably for cuts detection, it gives poor quality of detection especially when slowly transition occurs. Figure 4 shows that performing the new version of local histograms method on a documentary video type (The National History of an ALIEN) while using a relatively high threshold permits to avoid false detection and lead to a perfect ratio of cut detection. Unfortunately, dissolve was strongly missed.

Dissolve detection: Not only the Canny edge detector presents an important computation complexity but also it suffers, in the absence of motion compensation, to be sensitive to object motion. Moreover, text appearance, zooming and mainly slowly dissolves cause false and

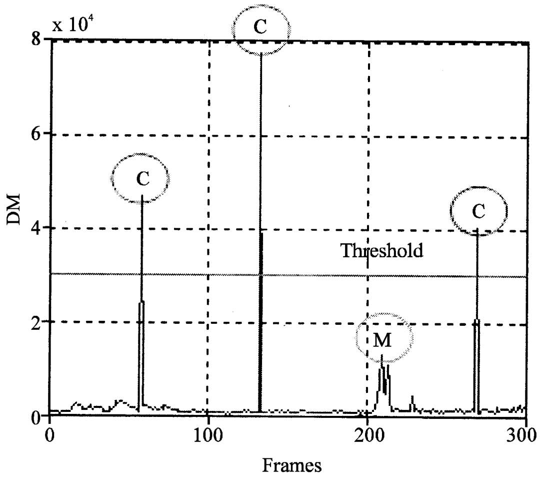


Fig. 4: Effectiveness of the local histograms detector

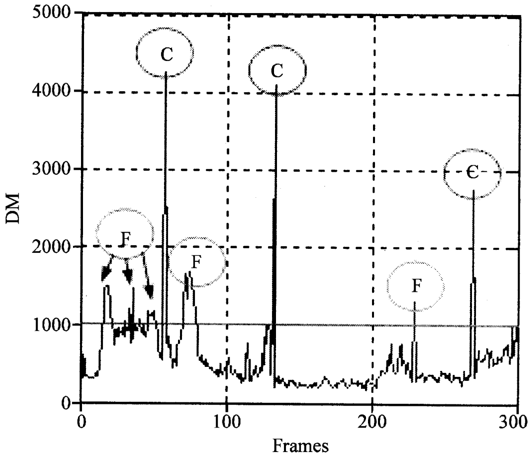


Fig. 5: Effectiveness of the Canny edge detector

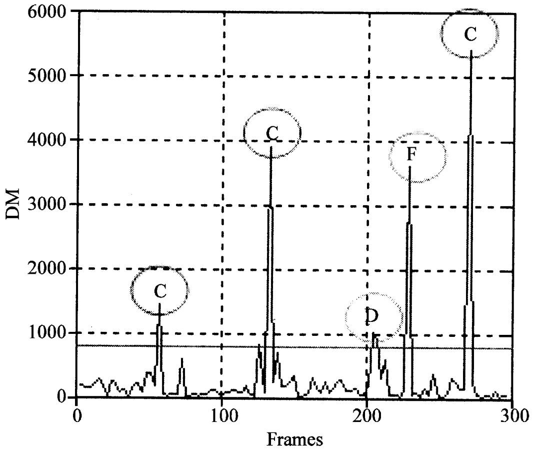


Fig. 6: Effectiveness of the Sobel edge detector with skipping frames

missed detections. The results of dissolve detection are shown in Fig. 5.

We denote in the following C as a cut, D as a dissolve, F as a false and M as a missed detection.

To reduce the ECR complexity algorithm and improve the effectiveness of dissolve detection we proceed as follows:

- Substitute the Canny detector by the Sobel detector
- Count pixels edge
- Compute the dissimilarity metric between non adjacent frames (we skip an interval of 3 frames to increase image difference especially for slowly dissolves)

A light modification was brought in the specifications of Sobel filter. In fact, after computing horizontal and vertical derivation on a frame, resultant pixels are determined as follows:

$$G = (|G_x| + |G_y|)/8 \qquad (3)$$

We divided in (3) the resultant pixel value by 8 instead of 9 because it consumes less computation time (a simple shift right).

The results presented in Fig. 6 show that our new approach performs well for the dissolve detection though, it leads to false detection during text appearance.

Video abstracting: To extract key frames, we combine the both new algorithms while considering that all detected peaks by local histograms represent cuts. On the other hand, peaks detected by skipping frames based method and which don't coincide with cuts or nearby cuts are dissolves.

Key frames provide a suitable abstraction and framework for video indexing, browsing and retrieval. The use of key frames greatly reduces the amount of data required in video indexing and provides an organizational framework for dealing with video content.

Much research work has been done in key frame extraction (Nagasaka *et al*., 1992; Grünsel and Tekalp, 1998; Wolf, 1996). The simplest proposed methods are choosing for each shot only one frame usually the first one, regardless of the complexity of visual content.

The process of video abstracting described in Fig. 7, shows well detection for cuts and dissolves but a high sensitivity to text appearance.

Comparison of methods: When real-time segmentation of video is required, an estimation of the number of operations to evaluate the Dissimilarity Measure (DM) between frames is required.

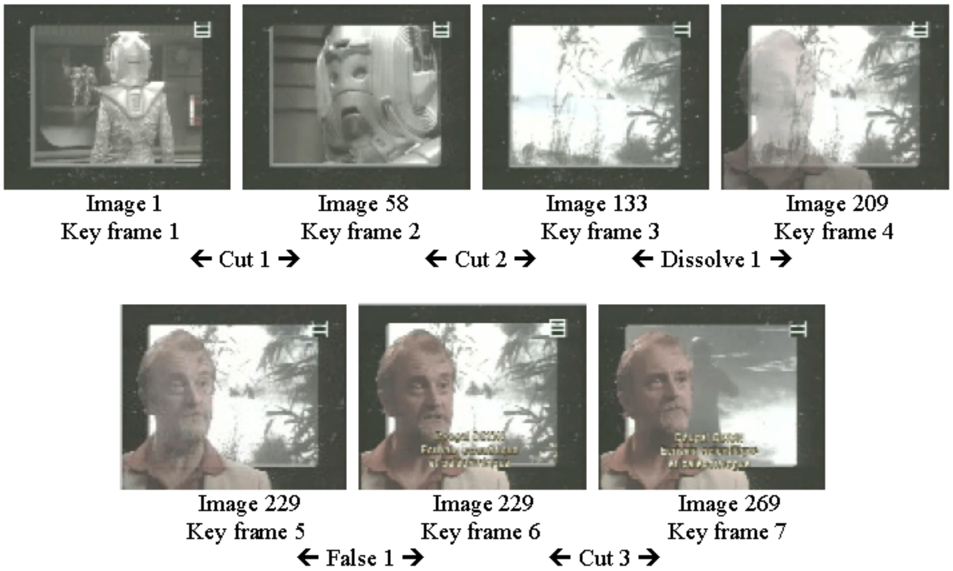


Fig. 7: Video summarization tested on documentary video type (The Natural History of an ALIEN)

Table 1: Computational time for various methods

Method	Number of operations
Pixel pair difference	9P
Red histogram difference	P+2N
χ^2 red histogram difference	P+7N
Edge change fraction	26P
Local histogram (RGB)	3P+6Nb
Local histogram (YC_bC_r)	3P+6Nb
Local histogram (C_bC_r)	2P+4Nb
Local histogram (Gray levels)	P+2Nb
Sobel edge detector [(1)]	= 38P

[(1)] Sobel is performed with skipping 3 frames

Considering the assumption presented earlier, we present in Table 1 the computational requirements for the main known algorithms, local histograms across 4 color spaces and the Sobel edge detector with skipping three frames. We denote P the number of pixels in a frame, N the number of bins (level of quantization), b the number of blocks in the frame (b = 16).

CONCLUSIONS AND FUTURE WORK

The present study have used two old and reliable algorithms used for cut and dissolve detection. During experiments, a first set of tests was performed on local histograms method. The use of this method in the gray space (at 4 bins) has offered good performances and significant diminution in computational resources. In the same way, the Sobel edge detector with a skipping interval of three frames has considerably improved the dissolve detection and reduced computation time. However, sensitivity to text appearance and object motion can lead to false detections.

Our future work consists in implementing hardware detectors of cuts and dissolves on an FPGA-based platform for the requirements of real-time multimedia applications such as video-on-demand.

REFERENCES

Abdel-Mottaleb, M. *et al.*, 1996. Content-based image and video access system. In Proceedings of ACM International Conference on Multimedia, Boston, MA., pp: 427-428.

Ardizzone, E. and M. Cascia, 1997. Automatic Video Database Indexing and Retrieval. Multimedia Tools and Applic., 4: 29-56.

Arman, F. *et al.*, 1993. Image processing on compressed data for large video database. Proceedings ACM Multimedia Anaheim, CA., 93: 267-272.

Boussaid, L. *et al.*, 2004. Hardware implementation solutions applied to automatic video segmentation based on histograms of blocks. Premier Cong. Intl. de Signaux Circuits Sys., (SCS 2004), Monastir Tunisie. pp: 610-615.

Boussaid, L. *et al.*, 2005a. Hardware design of a video content descriptor on an FPGA based Platform. Journées Francophones sur l'Adéquation Algorithme Architecture (JFAAA'05), Dijon France.

Boussaid, L. *et al.*, 2005b. A real-time shot boundary detection algorithm based on local histogram. SSD'05, IEEE International Conference on Signals Systems Decision and Information Technology, Sousse, Tunisia. Volume 3: 973-959-01-9/©2005/ 9885IEEE.

Dailianas, A. *et al.*, 1995. Comparison of automatic video segmentation algorithms. In International Issues in Large Commercial Media Delivery Systems, Proceedings SPIE 2615, pp: 2-16.

Grünsel, B. and A.M. Tekalp, 1998. Content-based video abstraction algorithms. In Proceeding of IEEE International Conference on Image Processing (ICIP'98), Chicago IL, pp: 128.

Jadon R.S. *et al.*, 2001. A fuzzy theoretic approach for video segmentation using syntactic features. Pattern Recognition Lett. Elsevier Science Inc., 22: 1359-1369.

Jung Hwan, O. and K.A. Hua, 2000. An efficient and cost-effective technique for browsing and indexing large video databases. In Proceedigns of ACM SIGMOD International Conference on Management of Data, Dallas, TX., pp: 415-426.

Lefèvre, S. *et al.*, 2003. A Review of Real-Time Segmentation of Uncompressed Video Sequences for Content-Based Search and Retrieval. Real Time Imaging, 9: 73-98.

Lienhart, R. *et al.*, 1998. A systematic method to compare and retrieve video sequences. In Proc. storage and retrieval for image and video databases. VI. 3312: 271-282.

Lienhart, R. and Jan, 1999. Comparison of automatic shout boundary detection algorithms. Proc. SPIE Conf. on Storage and Retrieval for Image and Video Databases VII, San Jose, CA, pp: 290-301.

Mas, J. *et al.*, 2003. Video shot boundary detection using color histogram. TRECVID2003 Conf., Gaithersburg, Maryland, USA.

Nagasaka, A. *et al.*, 1992. Automatic video indexing and full-video search for object appearances. In Video Database Sys. II, pp: 113-127.

Porter, S.V. *et al.*, 2001. Detection and Classification of Shot Transitions. In: Proc. 12th Br. Machine Vision Conf. Cootes, T. and C. Taylor (Ed.), BMVA Press, pp: 73-82.

Taskiran, C. and E.J. Delp, 1998. Video scene change detection using the generalized trace. In Proceedings of IEEE International Conference on Acoustic, Speech and Signal processing (ICASSP), Seattle, Washington, pp: 2961-2964.

Wactlar, H.D. *et al.*, 1999. Lessons learned from building terabyte digital video library. Computer, pp: 66-73.

Wolf, W., 1996. Key frame selection by motion analysis. In Proceeding IEEE, ICASSP'96, 2: 1228-1231.

Xiang, C. and P.N. Suganthan, 2002. Neural network based temporal video segmentation. Intl. J. Neural Sys., 12: 263-269.

Yu, H. and W. Wolf, 1997. A visual search system for video and image databases. In Proceedings IEEE International Conference on Multimedia Computing and systems. Ottawa, Canada, pp: 517-524.

Zabih, R., J. Miller and K. Mai, 1995. A feature based algorithm for detecting and classifying scene breaks. In Proceedings of ACM Multimedia '95, San Francisco, CA., pp: 189-200.