# Efficient Key-Frame Extraction and Video Analysis

Janko Calic and Ebroul Izquierdo
Multimedia and Vision Research Lab, Queen Mary, University of London,
e-mail: {janko.calic, ebroul.izquierdo}@elec.qmul.ac.uk

## Abstract

*Content based video indexing and retrieval has its foundations in the analyses of the prime video temporal structures. Consequently, technologies for video segmentation and key-frame extraction have become crucial for the development of advanced digital video systems. Conventional algorithms for video partitioning and key-frame extraction are mainly implemented autonomously. By focusing the analysis on the compressed video features, this paper introduces a real-time algorithm for scene change detection and key-frame extraction that generates the frame difference metrics by analysing statistics of the macro-block features extracted from the MPEG compressed stream. The key-frame extraction method is implemented using difference metrics curve simplification by discrete contour evolution algorithm. This approach resulted in a fast and robust algorithm. Results of computer simulations are reported.*

## 1. Introduction

The contemporary development of various multimedia compression standards combined with a significant increase in desktop computer performance, and a decrease in the cost of storage media, has led to the widespread exchange of multimedia information. The availability of cost effective means for obtaining digital video gained the easy storage of digital video data, which can be widely distributed over various types of networks or storage media. Unfortunately, these collections are often not catalogued and are accessible only by the sequential scanning of the sequences. To make the use of large video databases more feasible, we need to be able to automatically index, search and retrieve relevant material.

It is important to stress that even with leading edge hardware accelerators, factors such as algorithm speed and storage resources are concerns that still must be addressed. For example, although compression provides tremendous space savings, it can often introduce processing inefficiencies when decompression is required to perform spatial processing for indexing and retrieval. With this in mind, one of the initial considerations in development of a system for video retrieval is an attempt to enhance access capabilities within existing compression representations.

The main problem of streaming-media structuring is definition of modality-specific mappings from the domain of each medium into general hypermedia data model. Three distinct approaches to this problem are possible: statistical, syntactical, or semantic. The traditional engineering approach to audio and video data processing has been statistical through signal processing techniques. Syntax concerns only the relationships among symbols and the ways in which they can be manipulated, while semantics concerns the relationships among symbols and their human-dependent meanings. Gonzales in 0 advocates the proposition that syntax, not semantics, is the key to converting stream-based media into hypermedia automatically. Unlike semantics, automatic syntactic analysis does not require any external or prior knowledge. The versatility of syntax is that while it can exist on its own, independent of any human interpretation or intervention, the argument can also be made that semantic understanding can arise from syntactical analysis. Using a syntactical approach, we can potentially generate systems with semantic meaning automatically, although the meaning itself is unknown to the syntactical process.

Cognitively, predominant feature in video is its higher-level temporal structure. People are unable to perceive millions of individual frames, but they can perceive episodes, scenes, and moving objects. A scene in a video is a sequence of frames that are considered to be semantically consistent. Scene changes therefore demarcate changes in semantic context. Segmenting a video into its constituent scenes permits it to be accessed in terms of meaningful units.

Algorithms for scene change detection can be classified according to the features used for processing into uncompressed and compressed domain algorithms. Algorithms in the uncompressed domain utilise information directly from the spatial video domain: pixel-wise difference [2], histograms [3], edge tracking [4], etc. These techniques are computationally demanding and time consuming, and thus inferior to the compressed features based approach.

Since compressed-domain methods have become dominant in this field [5] we will concentrate more on features extracted directly from compressed video. We will specifically focus on the use of MPEG compressed streams.

Initial work in compressed-domain methods was done by Yeo and Liu [6]. They proposed the algorithm that analyses a sequence of reduced images extracted from low frequency coefficients in the DCT transform domain called the DC sequence. An interesting approach was proposed by Lee *et al*. [7], where they exploit information from the first few AC coefficients in the transformation domain, and track binary edge maps to segment the video. Two approaches similar to our metrics computation method are proposed by Kobla et al. [8] and Pei et al. [9], where the authors have analysed information extracted from MPEG motion estimation variables in various ways.

In current video indexing systems, after temporal segmentation of the analyzed sequence, a set of frames that best represent the visual content of the scenes is extracted. These frames are called key-frames and are used in the latter task of video indexing. An effective approach to key-frame extraction, based on temporal variation of low-level image features such as colour histograms and motion information, has been proposed by Zhang, et al. [10]. The key idea of this approach is that the number of key-frames needed to represent a segment should be based on temporal variation of video content in the segment. In this approach, the density of key-frames or the abstraction ratio can be controlled according to the user's need by adjusting the threshold for determining "significant" colour histogram changes and the overlap ratio of key-frames in panning sequences. However, the exact number of resultant key-frames will be determined *a posteriori* by the actual content of the input video. This fact has been argued to be a disadvantage of this type of key-frame extraction approach by Hanjalic et al. [11]. Our method offers trade-off between predetermined number of key-frames and controlling the level of abstraction ratio by determining "significant" feature changes.

This approach to the content-based video analysis that separates algorithms for temporal video segmentation and key-frame extraction is accepted without any criticism. Tasks of temporal video segmentation and key-frame extraction were always autonomous and were inadequately using processing resources.

In this paper, we present a novel approach to the problem of key-frame extraction. The algorithms for the temporal segmentation and the extraction of key-frames are unified in one robust algorithm with real-time capabilities. The initial research objectives were directed towards the performance of the main video processing algorithms in the compressed domain, using the established international video standards: MPEG 1-2, H.263 and in future MPEG4. The only feature used in the MPEG-stream analysis is statistics of MacroBlock prediction types [12]. A general difference metrics is generated and a specific discrete curve evolution algorithm is applied for the metrics curve simplification.

The proposed algorithm shows high accuracy and robust performance running in real-time with the good customisation possibilities. Future work will be oriented towards performance improvement in algorithm speed and preciseness.

This paper is organized as follows. In Section 2 the difference metrics for detection of visual content changes is presented. Section 3 describes the key-frame extraction algorithms build on the same difference metrics, as well as introduces discrete contour evolution for curve simplification. Overall results are presented in Section 4, while the Section 5 brings final conclusions and a summary of the paper.

## 2. Frame Difference Metrics

MPEG-2 encoders compress video by dividing each frame into blocks of size 16x16 called *MacroBlocks* (MB) [13]. Each MB contains information about the type of its temporal prediction and corresponding vectors used for motion compensation. The character of the MB prediction is defined in a MPEG variable called *MBType*:

- *Intra* coded, MB is not predicted at all
- *Forward* referenced, MB from the previous reference frame is predicting forward the corresponding MB in current frame
- *Backward* referenced, MB from the next reference frame is predicting backwards the corresponding MB in current frame
- *Interpolated*, both reference frames are predicting MB in current frame by interpolation.

Since the MPEG sequence has a high temporal redundancy within a shot, a continuously strong inter-frame reference will be present in the stream as long as no significant changes occur in the scene [14]. The “amount” of inter-frame reference in each frame and its temporal changes can be used to define a metric, which measures the probability of scene change in a given frame. We propose to extract only MBType information from the MPEG stream and, by analyzing it, measure this “amount” of inter-frame reference.

Without loss of generality we assume that in analyzed MPEG stream *Group Of Pictures* (GOP) will have the standard structure [IBBPBBPBBPBBPBB]. Observe that this frame structure can be split into groups of three having the form of a triplet: IBB or PBB. In the sequel, both types of the reference frames (I or P) are denoted as $R_i$, front bi-directional frame of the triplet as $B_i$, while the second bi-directional frame is denoted as $b_i$. Thus, the MPEG sequence can be analyzed as a group of frame-triplets in the form:

$$R_1 B_2 b_3 \;\; R_4 B_5 b_6 \; \ldots \;\; R_i B_{i+1} b_{i+2} \; \ldots$$

This convention can be easily generalized to any other GOP structure.

As mentioned above, a high visual similarity within a sequence should result in high percentage of predicted MBs in both bi-directional B frames and predicted P frames and lack of intra coded MBs. More precisely, if two frames are strongly referenced then the most of the MBs in predicted frame would have the corresponding prediction type: forward, backward or interpolated, depending on the type of reference. Thus, we can define a metric for the visual frame difference by analyzing the statistics of MBTypes in each frame.

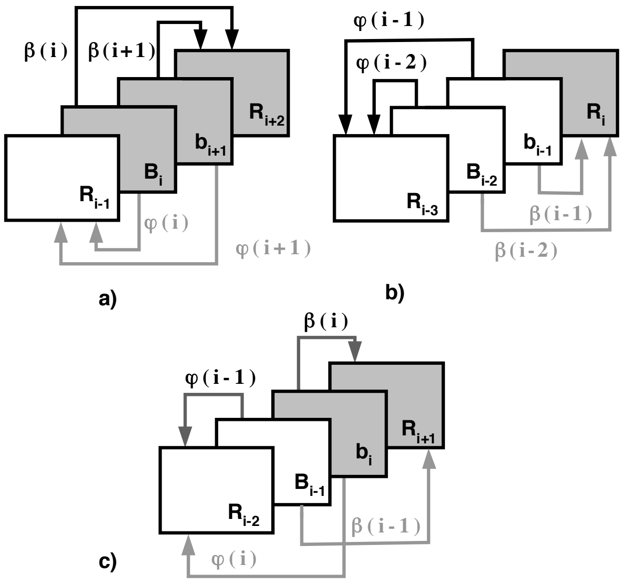

**Figure 1. Content change in a frame triple**

The possible locations of a content change (i.e. cut) in a frame triplet are depicted in Figure 1. If the front referenced frame $B_i$ is the first frame with different visual content (a), the next reference frame $R_{i+2}$ predicts backwards a significant percentage of MBs in both $B_i$ and $b_{i+1}$. If the content change occurs at the rear reference frame $R_i$ (b), then the bi-directional frames $B_{i-2}$ and $b_{i-1}$ will be mainly predicted forwards by the previous reference frame $R_{i-3}$. Finally, if the content change occurs at $b_i$ (c), then $B_{i-1}$ will be strongly predicted forward by the previous reference frame $R_{i-2,}$ while $b_i$ will be predicted backwards by the next reference frame $R_{i+1}$.

Let $\Phi_T(i)$ be the set containing all forward referenced MBs and $B_T(i)$ the set containing all backward referenced MBs in a given frame with index i and type T. In the same manner, we define sets of intra coded MBs as $I_T(i)$ and interpolated MBs as $\Pi_T(i)$. Then we denote the cardinalities of the corresponding sets as: $\varphi_T(i)$, $\beta_T(i)$, $\iota_T(i)$ and $\pi_T(i)$. The metric $\Delta(i)$ used to determine a visual difference measure within a frame triplet is defined as:

$$\Delta(i) = k_{\varphi B}\varphi_B + k_{\varphi b}\varphi_b + k_{\beta B}\beta_B + k_{\beta b}\beta_b + \\ k_{\iota B}\iota_B + k_{\iota b}\iota_b + k_{\pi B}\pi_B + k_{\pi b}\pi_b$$

By analysing the prediction character and behaviour in one frame triplet, we can estimate the changes in visual content within. Depending on the frame type, there are three different linear combinations of variables $\varphi_T(i)$, $\beta_T(i)$, $\iota_T(i)$ and $\pi_T(i)$ for both bi-directional frames in a frame triplet. Each linear combination has two main coefficients that are directly proportional to the visual content change within predicted and reference frame in a frame triplet (k=+1), and two that are inversely proportional (k=-1) to it. Additional factors $k_\pi$ and $k_\iota$ are describing overall change in a triplet, one in direct ($k_\iota$) and one in inverse ($k_\pi$) proportion. The coefficient values are determined by the rule of thumb, and are presented in Table 1.

| | T(i)=R | T(i)=B | T(i)=b |
|---|---|---|---|
| $k\varphi B$ | +1 | -1 | +1 |
| $k\varphi b$ | +1 | -1 | -1 |
| $k\beta B$ | -1 | +1 | -1 |
| $k\beta b$ | -1 | +1 | +1 |
| $k\iota B, k\iota b$ | +0.5 | | |
| $k\pi B, k\pi b$ | -0.5 | | |

**Table 1. Coefficients in the linear combination**

## 3. Key-frame extraction

### 3.1. Gaussian filtering

The raw difference metrics defined in the previous section has a strong noise that makes further processing of the data almost impossible. However, the source of this noise is in the discontinuous nature of the difference metrics. Since the metrics value is determined separately for each frame and the content change is based on frame triplet element, low-pass filtering with kernel proportional to triplet length would eliminate the noise. The filter with Gaussian pulse response (Figure 2) is applied:

$$h(i) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{i^2}{2\sigma^2}}$$

Where $i \in [-4\sigma, 4\sigma]$, and $\sigma=1.5$ . The value for $\sigma$ is chosen to maximize the smoothing within one frame triplet.

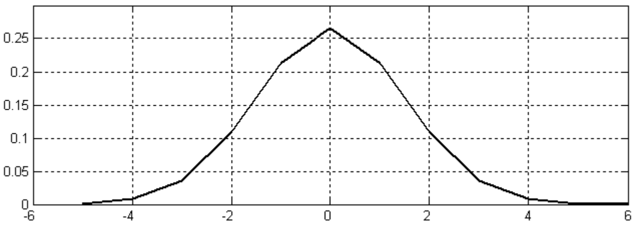

**Figure 2. Pulse response of the Gaussian filter**

Metrics with suppressed noise is calculated as a convolution of Gaussian filter pulse response and the raw noisy metrics:

$$\Delta = \Delta_N \otimes h$$

Example of noise suppression from a difference metrics is given in Figure 3.

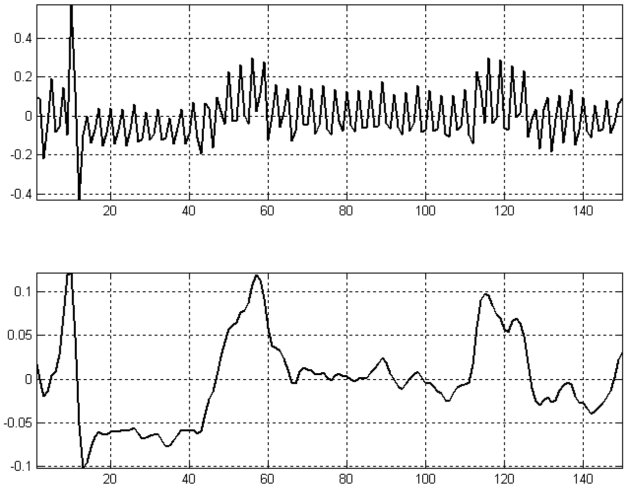

**Figure 3. Noise suppression**

### 3.2. Discrete Contour Evolution

In order to extract a number of representative frames from the sequence the previously defined difference metrics Δ(i) is simplified in a way that spurious and small changes in the metrics curve are discarded without any influence on the main features of the difference metrics. The algorithm that has these features is Discrete Curve Evolution (DCE). Main properties of DCE are [15]:

- It leads to the simplification of curve complexity, in analogy to evolutions guided by diffusion equations, with
- No blurring (i.e. peak rounding) effects and no dislocation of relevant features, due to the fact that the remaining vertices do not change their positions
- The relevance measure K is stable with respect to noisy deformations, since noise elimination takes place in the early stages of the evolution
- It allows us to find digital line segments in noisy metrics due to the relevance order of the repeated process of digital linearization.

Flowchart of DCE algorithm is depicted in Figure 4.

Let $D_m = s_0, \ldots, s_{m-1}$ be a decomposition of a digital curve S into consecutive digital line segments. The algorithm that computes the decomposition $D_k$ for each stage of the discrete curve evolution k>3 until we reach wished number of key points (NOKP). Approximate number of key-frames (NOKF) after the DCE algorithm is half of NOKP. The exact NOKF can be determined only *a posteriori*. The input video sequence has NOF frames.

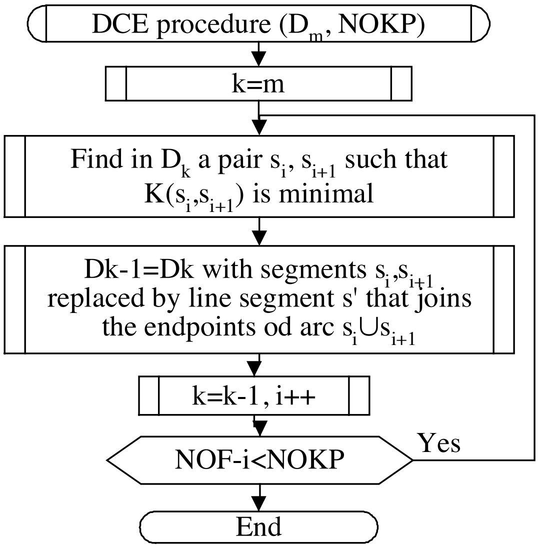
DCE procedure (D_m, NOKP)
k=m
Find in D_k a pair s_i, s_i+1 such that K(s_i,s_i+1) is minimal
Dk-1=Dk with segments s_i,s_i+1 replaced by line segment s' that joins the endpoints od arc s_i∪s_i+1
k=k-1, i++
NOF-i<NOKP
Yes
End

**Figure 4. Flowchart of the DCE algorithm**

Key-frames positions are determined by locations of the local minima in simplified metrics curve, while shot change central points are located as the local maxima.

### 3.3. Relevance Order

Our evolution process is guided by a relevance order. We assign to every pair of two adjacent line segments $s_1,s_2$ in a decomposition of a given digital curve S a cost $K(s_1,s_2)$ which represents the significance of the contribution of arc $s_1 \cup s_2$ to the shape of S. We order pairs of adjacent line segments with respect to this significance cost. We will call this order a *relevance order*. The linearization cost $K(s_1,s_2)$ of any supported arc $s_1,s_2$ depends on its length, its global curvature and area below the arc. It seems that an adequate measure of the relevance of arc $s_1 \cup s_2$ for the shape of a given object can be based on turn angle $\beta(s_1,s_2)$, on the lengths of the segments $l(s_1)$, $l(s_2)$ and the area of the region enclosed by $s_1 \cup s_2$. We assume that the larger both lengths, area enclosed and the total turn of the arc, the greater is its contribution to the shape of a difference curve. Thus, the cost function $K$ is monotonically increasing with respect to the arc lengths, area enclosed and the total curvature. This assumption can be justified by the simple analysis of the Figure 5.

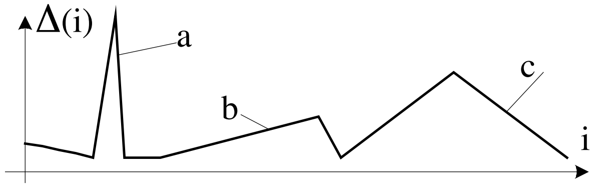

**Figure 5. Relevance order examples**

The peak-like change, depicted in Figure 5a is strong and fast change in visual content. Obviously it is result of a cut change in the video sequence and it has a strong turn angle. Case b) is very long change with long arc segments. Third case c) shows a gradual transition with big area enclosed, but without huge turn angle. These three simple examples depict three criteria for relevance order, and introduce the main ideas for definition of the relevance measure.

### 3.4. Relevance Measure

For each two adjacent line segments $s_1$, $s_2$ in the decomposition of a digital curve S, we determine the relevance measure $K(s_1, s_2)$, which represents the significance of the contribution of arc $s_1 \cup s_2$ to the shape of S. The value $K(s_1, s_2)$ can be interpreted as the cost required for linearization of arc $s_1 \cup s_2$. Let $s_1$=AB and $s_2$=BC be two consecutive line segments in the decomposition of curve S, so that $\beta=\alpha_1+\alpha_2$ is the turn angle. The corresponding cost function $K(s_1, s_2)$ is given by the equation:

$$K(s_1,s_2) = \left|\beta(s_1,s_2)\cdot(l_1+l_2)\cdot P_{\Delta(ABC)}\right|$$

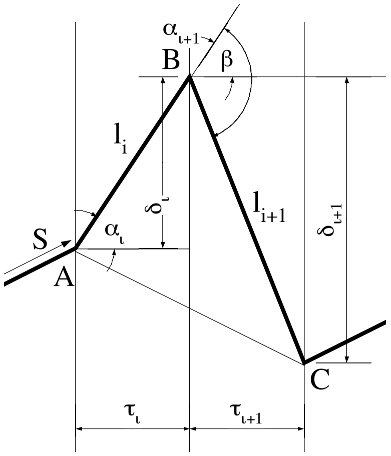

**Figure 6. DCE linearization**

Observing an arc linearization example given in Figure 6, formulae for each element in equation above for relevance measure are given as:

$$\delta_i = \Delta(i+1) - \Delta(i) \quad , \quad l_i = \sqrt{\tau_i^2 + \delta_i^2}$$

$$\beta(s_i, s_{i+1}) = acrtg(\delta_i/\tau_i) - acrtg(\delta_{i+1}/\tau_{i+1})$$

$$P_{\Delta ABC} = \frac{1}{2}(\delta_i\tau_i + \delta_{i+1}\tau_{i+1})$$

## 4. Results

The collection of C++ classes called Mpeg Development Classes, implemented by Dongge et al. [16], was used as the main tool for manipulating the MPEG streams, while Berkeley mpeg2codec was used as the reference MPEG codec. Test sequences were produced by Multimedia & Vision Research Lab, Queen Mary, University of London, while some were provided by Computer Vision Department, Dublin City University, Dublin, Ireland.

### 4.1. Shot Detection statistics

Since the algorithm comparison in temporal video analysis is mainly based on scene change detection, we will present the experimental results starting with shot detection statistics.

The applied statistical performance evaluation of temporal segmentation of the video sequences is "based on the number of missed detections (MD's) and false alarms (FA's), expressed as recall and precision" [17]:

$$Recall = \frac{Detects}{Detects + MD's}, \quad Precision = \frac{Detects}{Detects + FA's}$$

We took manually detected positions of the shot boundaries as the ground truth, defining in that way number of missed detections and false alarms. There were three main categories of video material analysed:

- NEWS; long monotonous sequences with mainly abrupt changes,
- SOAP OPERA; average shot length with some gradual changes and editing effects
- COMMERCIALS; short shots with a lot of gradual changes and editing effects

The shot changes detection procedure showed excellent results for different types of changes with almost 100% accuracy, as seen in Table 2.

| | Detect | Missed | False | Recall | Prec. |
|---|---|---|---|---|---|
| News | 87 | 2 | 6 | 98% | 94% |
| Soap | 92 | 2 | 9 | 98% | 91% |
| Comm | 127 | 9 | 16 | 94% | 88% |

**Table 2. Shot changes detection results**

### 4.2. Key-frame extraction results

Objective evaluation of how representative is the given set of key-frames is a very difficult task. After few experiments with different abstraction rate and different video content, the conclusion is that the algorithm shows subjectively better results for news and soap operas, while the content of the commercials is presented reasonably good with our video summary generated from the extracted set of the key-frames. The Figure shows three steps in DCE algorithm of a short commercial clip.

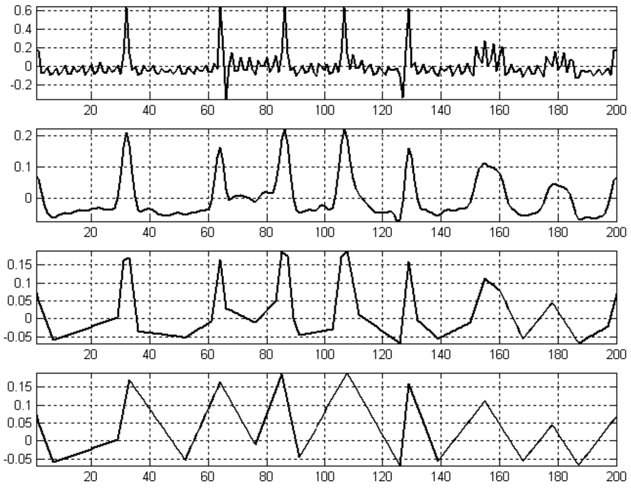

**Figure 7. DCE algorithm results**

It can be seen that the DCE algorithm deletes less important changes one by one without dislocating the vertices of the main difference metrics.

## 5. Conclusions

A novel key-frame extraction technique based on the difference metrics extracted directly from the MPEG compressed domain and discrete contour evolution is proposed. First, an algorithm for extraction of a frame difference metrics that uses inter-frame reference derived only from the statistics of the *MacroBlock* types was introduced. Second, a novel discrete contour evolution method was applied in the algorithm for key-frame extraction for curve simplification. Finally, the experimental results were presented in Section 4. As in all major methods that use the MB coding information, applied technique is relatively simple, requires minimum decoding and produces good accuracy [18]. The experimental results show high robustness in both temporal video segmentation and the extraction of the representative key-frame for a given scene.

We are investigating the possibilities of improving the advanced real-time method for initial steps of shot detection a key-frame extraction in the task of content-based video indexing and retrieval by using faster discrete curve simplification.

## 6. References

[1] Gonzalez R. "Hypermedia data modelling, coding, and semiotics" [Journal Paper] Proceedings of the IEEE, vol.85, no.7, July 1997, pp.1111-40. Publisher: IEEE, USA.

[2] H. Zhang, A. Kankanhalli and W. Smoliar, "Automatic partitioning of full-motion video", Multimedia Systems, vol.1, no.1, pp. 10-28, 1993.

[3] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances", Proc. IFIP 2nd Working Conf. Visual Database Systems, pp.113-127, 1992.

[4] R. Zabih, J. Miller and K. Mai, "A feature-based algorithm for detecting and classifying scene breaks", Proc. ACM Multimedia '95, pp.189-200, 1995.

[5] Zhang H. J., "Content-based Video Browsing and Retrieval", from "Handbook of Multimedia Computing" editor-in-chief Furht B., CRC Press, Boca Raton, Florida, USA, 1999.

[6] Boon-Lock Yeo, Bede Liu, "Rapid scene analysis on compressed video", IEEE Transactions on Circuits & Systems for Video Technology, vol.5, no.6, Dec. 1995, pp.533-44. Publisher: IEEE, USA

[7] Seong-Whan Lee, Young-Min Kim, Sung Woo Choi, "Fast scene change detection using direct feature extraction from MPEG compressed videos", IEEE Transactions on Multimedia, vol.2, no.4, Dec. 2000, pp.240-54. Publisher: IEEE, USA

[8] Kobla V., Doermann D.S., Lin K. -I., Faloutsos C., "Compressed domain video indexing techniques using DCT and motion vector information in MPEG video", Proceedings of SPIE conference on Storage and Retrieval for Image and Video Databases V, Volume 3022, pp. 200-211, February 1997.

[9] Soo-Chang Pei, Yu-Zuong Chou, "Efficient MPEG compressed video analysis using macroblock type information", IEEE Transactions on Multimedia, vol.1, no.4, Dec. 1999, pp.321-33. Publisher: IEEE, USA

[10] H. J. Zhang, C. Y. Low, and S.W. Smoliar, "Video parsing and browsing using compressed data", Multimedia Tools Appl. 1, 1995, 91–113.

[11] A. Hanjalic and R.L. Langendijk, "A New Key-Frame Allocation Method for Representing Stored Video Streams", Proc. of 1st Int. Workshop on Image Databases and Multimedia Search, 1996.

[12] J. Calic and E. Izquierdo, "Temporal Segmentation of MPEG video streams", submitted to Special issue on Image Analysis for Multimedia Interactive Services, EURASIP Journal on Applied Signal Processing, 2001

[13] LeGall D., Mitchell J.L., Pennbaker W. B., Fogg C.E., "MPEG video compression standard", Chapman & Hall, New York, USA, 1996

[14] J. Calic and E. Izquierdo, "Towards Real-Time Shot Detection in the MPEG Compressed Domain", Proceedings of the Workshop on Image Analysis for Multimedia Interactive Services, WIAMIS'2001, Tampere, Finland, May 2001.

[15] Latecki LJ, Lakimper R., "Convexity rule for shape decomposition based on discrete contour evolution" Computer Vision & Image Understanding, vol.73, no.3, March 1999, pp.441-54. Academic Press, USA

[16] Dongge Li, Sethi I. K. "MDC: a software tool for developing MPEG applications", Proceedings IEEE International Conference on Multimedia Computing and Systems. IEEE Comput. Soc. Part vol.1, 1999, pp.445-50 vol.1. Los Alamitos, CA, USA

[17] Gargi U., Strayer S., "Performance Characterisation of Video-Shot-Change Detection Methods", IEEE Trans. on Circuits and Systems for Video Technology, Vol.10, No.1, February 2000

[18] I. Koprinska, S. Carrato, "Detecting and classifying video shot boundaries in MPEG compressed sequences", in: Proc. IX Eur. Sig. Proc. Conf.(EUSIPCO), Rhodes, 1998, pp. 1729-1732.