

A sequence analysis system for video databases

M. Ceccarelli ^a, A. Hanjalic ^b, R.L. Lagendijk ^b

^a Philips Research Laboratories Eindhoven, Storage and Retrieval Group, Prof.Holstlaan 4, 5656 AA Eindhoven, The Netherlands

^b Delft University of Technology, Dept. of Electrical Engineering, Information Theory Group, P.O.Box 5031, 2600 GA Delft, The Netherlands

The proliferation of digital transmissions and video services will lead to a demand for efficient and flexible local storage devices for time-shifting, personal archival and fast access to downloaded material. Visual search tools must be provided for easily locating specific information within huge volumes of video data [1]. In this paper we will consider advanced methods for automated analysis of compressed video sequences, extraction of representative information and its organization in a video database for efficient browsing and retrieval.

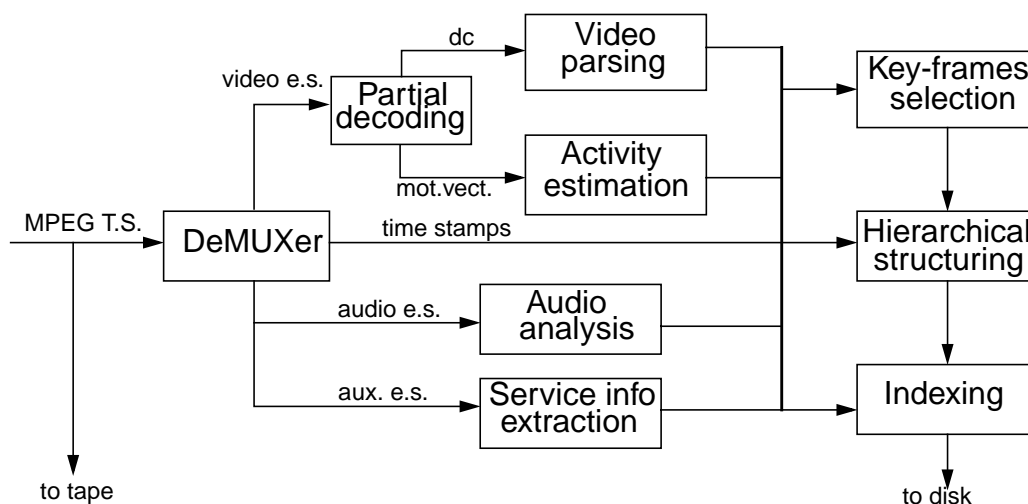
1.0 INTRODUCTION

In the context of the European ACTS project SMASH, technical possibilities for a consumer storage device for multimedia applications are currently under investigation. The main issues involved are the recording of digital video streams and multimedia documents. The current focus of our research is on the implementation of an automatic video abstracting system, able to simplify retrieval operation on large amounts of video data.

The high bit rate and the compressed nature of the video streams together with the large amount of stored data hinder the retrieval task of the user, but the new digital technology adopted in video transmissions coupled with advancement in digital signal processing (motion analysis, pattern recognition, speech recognition) provide an interesting framework for automatic analysis of the video programs. We will focus on storage of MPEG compressed streams [2], used in present and future digital video services, including DVB and ATV transmissions, but the same principles can be applied to other coding systems, e.g. to support non-linear editing of streams from a digital video camcorder (DVC).

With MPEG video streams, the standard trick-modes (VCR-like fast viewing operations) cannot be easily implemented, hence new fast and effective techniques for

Figure 1. Overview of the system for real-time analysis of recorded video streams



video browsing must be found. The descriptive power of textual annotations is unsatisfying for our purposes, a visual representation is needed. The solution could be found by coupling a textual description with representative images, but manual insertion of text and extraction of images would be time consuming, hence the whole system must be automated.

Real time analysis of the video content is performed during the recording operation, aimed at producing a practical reference to the semantic structure of the programme and to extract the information necessary for the indexing module. In order to obtain a reconstruction of the original storyboard, structural analysis is performed to parse the video through a process of *reverse-editing*.

2.0 VIDEO PARSING

The video parsing routines produce a temporal segmentation of the sequences in their structural elementary units, the camera-shots. The *scene change detection* algorithms identify the boundaries between consecutive camera shots by determining the frames where a transition occurs from one shot to another. [The different shots composing the *final cut* may have been assembled, during the editing process, adopting several techniques: cuts (abrupt scene changes), fades (fade in and fade out), dissolves, wipes, and other special mixing effects.]

Many scene change detection techniques operate on the pixel domain, but full decompression of the bit stream would involve Huffman code decoding, inverse quantization, inverse cosine transformation and motion compensation. The algorithms we introduce operate on compressed sequences, requiring just minimal

decoding, since we exploit the available information on frequency component values and motion vectors.

2.1 Analysis in the Compressed Domain

An MPEG stream consists of three types of frames: every 8x8 pixels block of an I frame is encoded with a 2D DCT by only using the information of the corresponding block in the present picture (intra-coded), P frames use also blocks which are predicted from a previous picture by using motion estimation and blocks in B frames can employ forward or backward prediction or both (bidirectional interpolation). In P and B frames what will be encoded in a DCT transformed block is the residual error after prediction.

It is possible to obtain useful information about the encoded I frames without performing IDCT, by using the DC values of the DCT. Since these yield an average value for each 8x8 block in the spatial domain, we can reconstruct a reference image (called DC-picture) reduced from the original by a factor of 8 (by 16 in chrominance components in case of a 4:2:0 sampling format). This subsampled pictures can be used for detecting scene changes [Arman, Hsu, Chiu]. The spatial averaging obtained with DC values also results in less false alarms in cut detection because of decreased sensitivity to local variations [3 Xiong]. However, this technique shows its weakness with standard Group of Pictures (GOPs) involved in video broadcasting, which employ several P and B frames between I-frames. Due to the low temporal resolution, there may be false detections in sequences with high motion, and the scene change cannot be exactly located. In order to obtain even a reduced resolution version of P and B frames, complete IDCT decoding should be performed in order to apply motion compensation. This process would be computationally expensive, especially for B frames, where each block could have been encoded in at least four different manners.

A few proposals avoid full decoding of the predicted frames by adopting considerations on the number of predicted blocks in comparison with the intra-coded blocks as a criterion for detecting scene changes occurring on P and B frames. Of course this number also depends on the search range adopted by the motion estimation algorithm of the encoder, hence false alarms could happen in high motion sequences.

Furthermore, since we mainly concentrated on broadcasted (DVB) sequences, we have to keep into account that professional MPEG encoding systems for content and service providers normally employ a scene change detector on the original sequence to be encoded. In order to optimize the bitrate/quality ratio in the encoded sequence, as a sharp scene change is encountered, the GOP length is adapted in order to have a closed GOP (ending with a P or I frame) in the end of each shot and an I frame corresponding to the first frame of a new shot. Yet it is necessary to improve the robustness of scene change detection in sequences with high motion content or where special mixing effects like fades and dissolves are present, by increasing the temporal resolution of the monitored frames.

The interval between an I and a P frame or between two P frames is normally not larger than two (B) frames, hence in our algorithm DC-pictures are also extracted from P frames. The DC values of predicted macroblocks in P frames can be obtained through an approximated inverse motion compensation as follows: the DC values of the block of the present picture are obtained by an area-weighted average of the four blocks pointed by the motion vector in the previous frame plus the residue error term of the prediction (see figure 1.) Since cascaded prediction ultimately refers to the anchor I frame, the quality of these approximated picture will deteriorate as the distance from the anchor frame increases, but tests revealed reasonable quality for standard 13 frames long GOPs.

2.2 Scene Change Detection

One of the main issues in temporal segmentation techniques is the definition of a valid measure to express the difference between two frames. The histogram of the colour components is one of the most effective metrics and has the advantage of being insensitive to motion. The RGB colour space is commonly used in literature, but the performances of cut detection algorithms in the YUV colour space, adopted by the MPEG standard, were proven to give more satisfactory results [4]. Before parsing, a simple test, based on colour histograms, is used to detect whether the analysed sequence is in black and white, so that proper measures can be employed during the parsing process.

For detection of cuts, the difference between two consecutive frames must be measured. In order to compare the binned distribution of the two frames, we compute the difference of the combined histograms of the U and V chrominance components [Sethi, Pathel 95]. Adopting the Chi-square measure for statistical binned distributions, we can define the global difference between the frame at timecode t_a and the frame at timecode t_b , given their N-bins histograms, as:

$$\chi^2(f_a, f_b) = \sum_{i=0}^{N-1} \frac{(H_i^u(f_a) - H_i^u(f_b))^2}{(H_i^u(f_a) + H_i^u(f_b))^2} + \frac{(H_i^v(f_a) - H_i^v(f_b))^2}{(H_i^v(f_a) + H_i^v(f_b))^2} \quad (1)$$

To determine whether a scene change has occurred, we have to compare the resulting frame to frame difference value with a threshold. The setting of a proper threshold is very much dependent on the analysed sequence, hence it is important to stress the problem of an optimal setting. Considering the variations of the average frame to frame difference, depending on the evolution of the content, an adaptive thresholding can be applied.

The class of frame difference values in correspondence of a scene change must be separated from the normal difference values along a shot. Techniques using mean and standard deviation and k-means clustering algorithms were discarded because missed detections spoil the statistics. Instead we adopt a technique for shot adaptive thresholding based on the concept of a temporal sliding window around the presently analysed frame [], to which we apply a temporal differential filter. When the

ratio of the difference between the present and the past frames is by far larger than the differences between all other neighbouring frames within the window, the present frame is classified as the first of a new shot. By means of this technique we could obtain a quite robust scene change detector with performances between 93% and 98% of correct detection, depending on the analysed sequences, and a quite low false alarm rate, around 4-5%.

3.0 KEY FRAME EXTRACTION

The goal of key frame extraction algorithms is to obtain a synthetic representation of the most meaningful scenes of a program. Theoretically, semantic primitives as objects, actions, events should be used, but such analysis is not currently feasible. Extraction has been based, so far, on scene changes: all frames corresponding to a detected scene change are extracted, but the first frames of shots generally hold low representativity (worst cases being fades and dissolves). Some proposals decide to select a particular frame as a key frame if a certain frame difference measure exceeds a threshold, but still the selected frames may not be the most representative ones since the threshold can only be subjectively chosen. Another important drawback is that the resulting number of key-frames is known only *a posteriori*. In practical applications a limit will exist on the maximum allowed storage space and an excessive number of key frames would not be handy for manipulation of the user. The number of resulting key frames must be set *a priori* and the extraction rate must be adaptive on time and content in order to avoid massive overhead information (e.g. in case of commercials, musical video-clips or trailers).

3.1 Activity Estimation

In order to estimate the effectiveness of a frame in representing a particular scene, the information about temporal and spatial activity is particularly valuable. It can

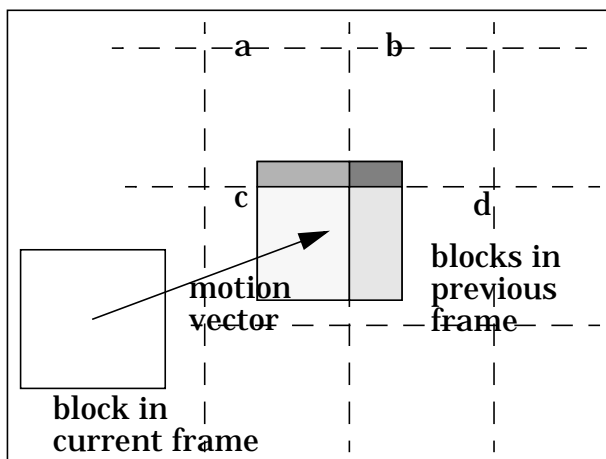


Figure 2. Approximated motion compensation for DC-pictures from P frames.

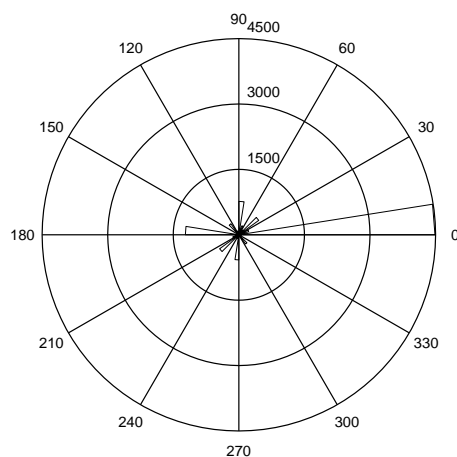


Figure 3. Distribution of phases of motion vectors in a sequence with camera panning.

be noticed that generally a few key frames are sufficient to represent stationary sequences, while more key frames are necessary to represent sequences containing considerable activity. Based on such a criterion, an activity estimation module monitors the sequences by giving a low value where scarce motion is measured and vice-versa.

Information about motion is carried in MPEG sequences by predicted (P) and interpolated (B) frames. The real temporal variation of the content must be distinguished from motion due to *camera operations* or moving background [6], which will not be considered for evaluation of content activity. The dominant motion components and the characteristic patterns of vectors must be examined in order to distinguish different classes of camera operations and detect the areas containing vectors due to translation, rotation or scaling camera operations.

The vectors due to camera operations are detected through histogram of vectors: in order to classify the motion vectors, the total distance from the phase of the modal vector is computed. If a motion vector is found out to be due to a camera operation, it is not taken into account in the total sum of the vector which yields the activity measure due to only object motion.

We must take into account that when an MPEG encoder employs a restricted motion search area, the number of intra-coded macroblocks in predicted frames will increase. Therefore we assume areas with intra-coded macroblocks in predicted frames, as expression of significant variations in the content. Motion vectors are also weighed considering the distance from the centre of the screen, where probability of meaningful action increases.

We can define the obtained sum of the motion vectors of the content as a measure for quantifying the activity of a n-th frame of the shot i:

$$A_i(n) = \sum_k^P (w_k |\vec{m}_k|) + \alpha \cdot \sum_i^I w_i \quad (2)$$

Where m_k are the motion vectors of predicted macroblocks not due to camera operations, I is the number of intra coded macroblocks and w is a weight for the position of the macroblock.

3.2 Key Frame Allocation

The cumulative action of i-th shot can be represented by:

$$C_i = \sum_{n=1}^{N_i} A_i(n) \quad (3)$$

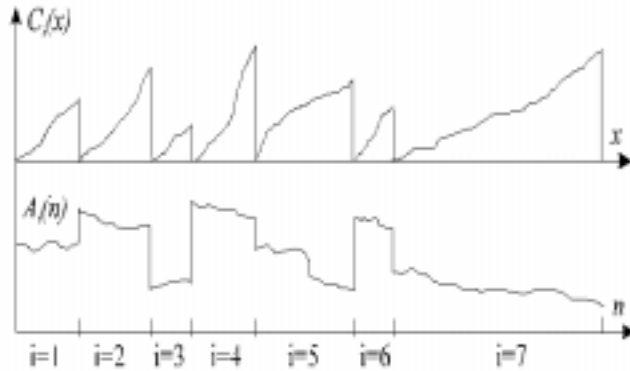


Figure 4. Action measure $A_i(n)$ along several shots and the corresponding cumulative action function $C_i(x)$

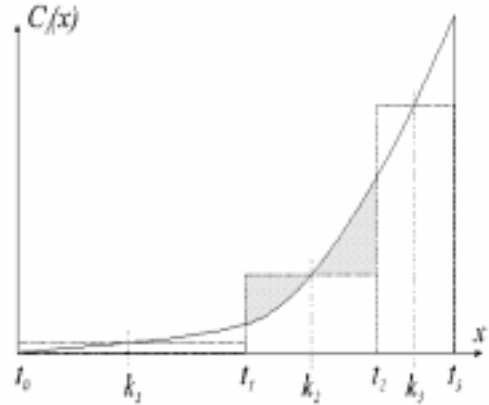


Figure 5. Distribution of key-frames and breakpoints for a varying cumulative action measure $C_i(x)$

The steepness of the resulting non decreasing curve of C_i is proportional to the activity of the content. An example is depicted in figure 4. In this manner it is possible to adapt the key-frame extraction rate according to the content of the sequence, for example by allocating to each shot of the sequence, according to their content activity, one part of the total N key-frames assigned for the whole program. We assign then to the i -th shot a number of key-frames K_i , which is taken proportional to the cumulative action C_i of the shot. This parameters can also be adapted for each program by identifying whether the category of the analysed sequence is movie, sport event, news etc. (e.g. extraction rate can be limited for commercials, trailers and musical video-clips).

3.3 Key Frame Selection

The following step is the actual selection of key-frames to represent each shot. To this end we apply the following criterion. Given the cumulative action $C_i(x)$ for the i -th shot, we distribute the K_i number of key-frames such that the following criterion function is minimised:

$$g(k_1, \dots, k_{K_i}, t_1, \dots, t_{K_i-1}) = \sum_{j=1}^{K_i} \int_{t_{j-1}}^{t_j} |C_i(x) - C_i(k_j)| dx \quad (4)$$

where k_j are the temporal positions of the key frames, t_{j-1} and t_j are the breakpoints between the shot segments represented by the keyframe k_j . A recursive search algorithm can be employed to solve this equation [Lagendijk]. Given the low temporal resolution of the reference frames, a few iterations should be sufficient in most cases. The underlying concept is that, once a given shot is allocated a number of key-

frames, a larger number of key frames is extracted along the shot where higher activity is measured. Figure 5. presents the results of such an optimal allocation for an example sequence having stationary content in the first part and increasing activity in the second part.

4.0 CONCLUSIONS

In this paper an automatic system for abstracting visual content from compressed video sequences has been presented. This has been achieved by identifying a reliable parsing technique, defining a suitable measure for representation of content and by introducing a new approach for key-frame allocation. Through this method it is possible to control the number of key-frames extracted from a video sequence. The extraction is not based on any parameter setting, but is fully automated. The applied numerical algorithm optimally delivers locations for the assigned amount of key-frames in each shot. The analysis of the content gives support for a high representativity of the selected key-frames. Future work includes clustering of key frames for pyramidal search operations and exploitation of audio tracks.

REFERENCES

1. F. Arman, R. Depommier, A. Hsu, M.Y. Chiu: "Content-based Browsing of Video Sequences", Proc. ACM Multimedia-94 pp. 97-103
2. ISO/IEC JTC 1/SC29: "ISO/IEC 13818, Information Technology - Generic coding of moving pictures and associated audio information", November 1994
3. W. Xiong, J.C.M. Lee, M.C.Ip: "Net comparison: a fast method for classifying image sequences", SPIE 2420, 1995, pag 318-328
4. U.Gargi, S.Oswald, D.Kosiba, S.Devadiga, R.Kasturi: "Evaluation of video sequence indexing and hierarchical video indexing", SPIE vol. 2420, 1995, pag 144-151
5. I.K.Sethi, N.Pathel: "A statistical approach to scene change detection", SPIE vol 2420, 1995, pag 329-335[
6. A.Hampapur, R.Jain, T.Weymouth: "Digital Video Segmentation", Proc. ACM Multimedia 1994, pag. 357-364
7. H.J. Zhang, C.Y. Low, S.W. Smoliar: "Video parsing and browsing using compressed data", Multimedia tools and applications, Mar. 95, pp 89-112
8. ETSI (European Telecommunication Standards Institute): "Digital broadcasting systems for television, sound and data services: Specification for Service Information in Digital Video Broadcasting systems", Draft pr ETS 300 468, November 1994

9. M.M.Yeung, B.L.Yeo, W.Wolf, B.Liu: "Video browsing using clustering and scene transitions on compressed sequences", SPIE vol 2420, Feb 1995
10. F.Arman, A. Hsu, M.Y. Chiu: "Image processing on compressed data for large video databases", ACM Multimedia 1993, pages 267--272