

ACHIEVEMENTS AND CHALLENGES IN VISUAL SEARCH OF VIDEO**

*Alan Hanjalic** , *Reginald L. Legendijk** , *Jan Biemond**

Abstract

This paper reflects on the state-of-the-art and attempts to point out some challenges for further research towards more efficient methods for searching and accessing of stored visual information. The emphasis will be put on investigating the applicability of current approaches for search through stored video data in case of digital video libraries, and especially the so-called home - digital libraries. The development of such interactive home environments is continuously growing and becoming a very important application area with many new requirements, especially concerning the available storage space and improved user-interaction.

1 Introduction

Consumers will not have to wait for a long time until digital libraries become widely spread and the most common way of storing and using large amounts of textual, audio and visual data. Development of such storage systems is primarily spurred forward by an unceasing and very intensive improvements in digital storage media technology, with continuously increasing *Storage capacity / Dimensions* ratio. The main advantage of this way of data storing, i.e. after their transfer in digital form, is, naturally, the preservation of the quality of stored information. But also the almost unlimited possibility to manipulate and browse through data by using comfortable digital user-interfaces (e.g. PC) is to be taken into consideration.

Vast number of users is expected to be connected world-wide via electronic networks to a variety of such large storage systems. At the same time, many users are expected to build up their own "small" digital libraries at home, containing a couple of tens or hundreds of GBytes of multimedia data. In fact, the connection to big public digital libraries strongly suggests that users have some sort of local storage system at home, where they can store the data they get from the public library. Several important reasons can be given for the necessity of the local storage system, such as avoiding the retransmission of data and using the "off-peak"-times in the network for performing the transmission - both leading to reduction of transmission costs.

The development of digital libraries of any kind need not only to consider the improvements of actual storage units (disc, tape). Such large amounts of data are only usable if there are efficient subsystems for finding and accessing the desired parts of stored information. Searching through multimedia data means searching through visual, textual and audio data separately, as well as through different combinations of these data. It should not be difficult to realize, that e.g. one particular camera-drive in the selected movie can be found on much more robust, reliable way, by tracing both the visual and corresponding audio information, i.e. by performing both

* Department of Electrical Engineering, Delft University of Technology, Information Theory Group, P.O. Box 5031, 2600 GA Delft, The Netherlands, e-mail: {alan,inald}@it.et.tudelft.nl

**This work was supported in part by the EU ACTS program under contract AC018: SMASH (Storage for Multimedia Applications Systems in the Home).

Relevant WEB-Page: <http://www-it.et.tudelft.nl/pda/smash>

visual and *audio search*. But also by such combined search procedures, efficient algorithms for tracing of any component of the multimedia information separately, must be available, in order to better “combine their forces”. This paper considers only the *visual* aspect of the whole search strategy, i.e. efficient tracing of stored *visual* information, and in particular of *video data* stored in a *compressed form*.

In section 2 the general scheme of search through stored video-data is presented, containing “pre-search-” as well as actual search- and retrieval procedures. Section 3 gives the scope of present development and some problems yet to be solved concerning the first pre-search-procedure, called video-parsing. Section 4 first gives current general approach for representing detected elementary video-parts (detection done by video-parsing) and some of its drawbacks, especially related to the usage of home-digital libraries. A novel idea for optimizing the key-frame extraction by taking into account user’s and system specifications is presented in the second part of this section.

2 Searching through digital video libraries

In order to enable user to search for, select and retrieve the desired video-part, several activities have to be carried out as a preparation for this user-interaction. The main goal of these procedures is to provide the user an overview over the whole stored video-information. This overview should contain some representative information (*key-information*) about all parts of the stored video. It is also possible to “user-friendly” organize (cluster) this representative information, before presenting it to user. The main intention is that users can browse through and select any of these characteristic data, which leads to retrieval of the corresponding video-part, represented through this data. The following pre-search procedures can be distinguished:

- Partitioning of video-streams (also called video-parsing, shot-change detection), i.e. defining smallest retrievable video-parts
- Representing each elementary video-part through appropriate key-information
- Postprocessing (Clustering) of extracted key-information for more effective user-interaction

Figure 1 shows pre-search-, search- and retrieval procedures on a time-scale, related to the video-stream coming into the storage system. For each of mentioned pre-search-procedures, a number of proposals for improvement of their efficiency and reliability can be found in the recent literature (e.g. [1 -12]).

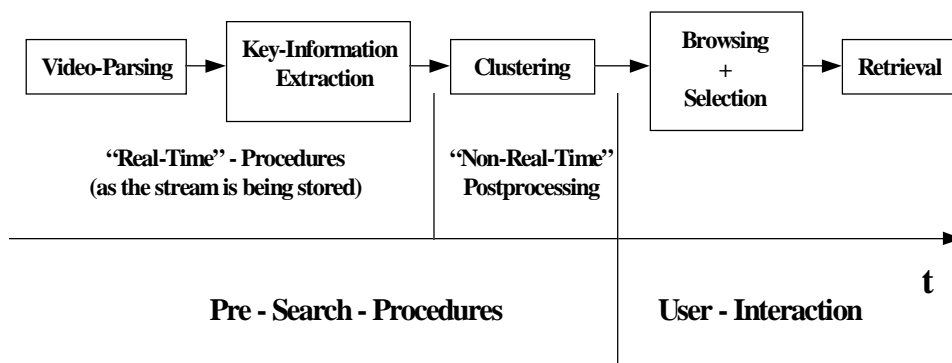


Figure 1: Temporal scheme of all activities leading to the presentation of selected video-part to the user

It is important to realize that pre-search-procedures make the actual search possible since they give an input for user-interaction. The user-friendliness of this interaction depends mostly on the way in which this preparation is done. For that reason, the main goal in the area of visual (video) search is *to optimize all preparation (pre-search) procedures* mentioned above. The scheme in Figure 1 took into consideration the idea of extracting key-information while the incoming stream is being stored. The stream is ready for browsing right after it has been stored.

3 Video-Parsing

The main goal of video-parsing is to mark elementary video-parts in the incoming stream. These parts are smallest retrieval-units, also called *video-shots*. One video-shot can be defined as “an unbroken sequence of frames, e.g. a zoom of a person talking” [12]. Each such collection of consecutive frames is characterized through the fact that frames are strongly related to each other, i.e. they represent one and the same action - “content” of the shot. Major consequence of such interframe-relationship is that the curve of frame-to-frame differences is relatively smooth along all frames of a video-shot, while showing sharp peaks at each place where a shot-change occurs. These light recognizable peaks - Figure 3a (page 6, dotted line) - indicate relative low relation between frames from different shots, in comparison to those within one and the same video-shot.

3.1 Difference - Metrics

Efficient detection of shot-change positions out of frame-to-frame difference curve is the subject of video-parsing, and is generally done by computing and comparing interframe-differences in the stream. Many comparison-algorithms have been proposed in the literature, where most of them are performed in the pixel-domain. Nagasaka and Tanaka investigated in [3] several metrics for calculating frame-to-frame differences, and concluded that the so-called χ^2 - comparison of color-histograms is the most robust comparison method.

The requirement that one difference-metrics should be “robust” is primarily related to the fact that the computed frame-to-frame difference should well represent the content-development of the video, i.e. it should be less sensitive to frame-to-frame changes which are not important for the current action (e.g. some minor motion in the background) in comparison to those characterizing changes in video-content, i.e. shot-changes. The more robust the metrics, the less mistakes by detecting shot-changes, in this case called “false alarms”. However, the used difference-metrics should be sensitive enough in order not to miss “real” shot-changes.

With increasing usage of *compressed* video-streams, new requirements appeared, related to techniques of shot-change detection. In the compressed domain (e.g. MPEG) there is no possibility for performing usual comparison metrics on frames, unless the decompression is done, at least up to one certain level. In [1] a shot-change-detecting algorithm is presented, which employs comparison of consecutive I-frames, and in case of P- and B-frames the motion continuity is investigated. In this approach, only a *partial decoding* is done up to the level, where DCT-coefficients and motion vectors become available (DCT-level). For comparing I-frames, from all 8x8 DCT blocks only DC-coefficients are used, by taking into account that this coefficient is the average of all pixel values within that 8x8 pixel-block. From all DC-coefficients of one frame a so-called “DC-image” (in fact subsampled original image) can be made, which preserves all important properties of the original. Having these DC-images enables usage of all comparison metrics known from a pixel domain, now applied to DC-images, which additionally leads to reduced computational complexity due to smaller image size. Similar approach is presented in [2], where DC-images are reconstructed (or in case of P- and B-frames only approximated) for all frames in the stream.

3.2 Threshold - Selection

While finding an appropriate method and metrics for comparison of consecutive frames is not a significant problem any more, the problem of interpreting difference values, i.e. actual selection of certain values to be shot-changes, still remains the major obstacle in practice. The proper selection of difference values is usually done by setting *thresholds*, although in [2] one other approach with a *sliding window* is proposed. There the decision about the shot-change is made after investigating the relationship among difference values within the window. Furth et al. give in [4] a statistical approach for determining the threshold, based on measuring mean-value μ and standard deviation σ of frame-to-frame differences. The threshold T is variable and for each difference-value estimated as

$$T = \mu + \alpha\sigma \quad . \quad (1)$$

The parameter α still need to be assigned an appropriate value. Experimental results in [5] suggest that α should have values between 5 and 6. Similar idea is used also in [6]. Since all statistical parameter should be reset after each shot-change, every missed detection or “false alarm” influences threshold-values for coming shots on a negative way, causing a burst of detection-mistakes. The window-approach in [2] partially avoids this problem, because the investigation of frame-to-frame differences is done locally.

By considering the application of shot-change detection algorithms as a feature in digital libraries, there is one general problem to be solved, such as *making the shot-change detection be completely automatical and work reliably for video-streams with any possible content-development coming into the storage system*. From current approaches, it can be easily recognized that such automatics is not possible, since the effectiveness of the detection strongly depends on applied *parameter values*.

4 Extraction of the key-information

The user’s browsing, selection and eventual manipulation of stored video-data can be efficiently realized by the concept of key-information, which is stored separately and which fully represents content of stored videos. What is herewith represented are actually video-shots, detected by video parsing. The user selects the desired shot after recognizing its content from the key-information. *Key-frames*, i.e. some characteristic frames of a shot, are most suitable for representing the shot-content. For more efficient user-interaction, the number of key-frames extracted for one particular shot should correspond to the “action” within the shot. Much action needs more key-frames for good representation than a “slow camera-drive over a nice landscape”. In [6] is proposed to take always the first frame after detected shot-change as one key-frame. Then, a threshold for measuring the content-development is defined - similarly to the procedure by video-parsing - and a frame-comparison along the shot in relation to this threshold is done. Each jump of a frame-to-frame difference above the threshold leads to extraction of an additional key-frame for that shot. This approach can be taken as a representative of the current state-of-the-art concerning the key-frame extraction.

4.1 Drawback of current approaches

In home-digital libraries with limited storage space and (possibly) fast user-interaction, the mentioned approach for key-frame extraction causes two problems. Firstly, the number of key-frames per shot is known only *a posteriori*. The user is confronted with the size of the key-

information for the whole video-stream after the extracting procedure, leading to a danger that not all key-frames can be stored because of the lack of storage space or to uncontrolled usage of the storage space. A large number of key-frames could also make postprocessing (clustering) last too long. In case of keeping extracted key-information on portable or other storage units not connected to the user-interface, the process of downloading large key-information to the interface storage unit may last too long. Secondly, the threshold for selection of content-representing shot-frames can only be chosen on an *individual (subjective)* basis. This results in many different possible collections of (generally unpredictable) key-frames for one and the same video-shot.

4.2 Key-information extraction as an optimization problem

From the above discussion we conclude there is a need for formulating the key-frame extraction as a more coherent and consistent problem. The “best” key-frame collection from a given (parsed) video-stream should follow from an optimization of a given criterion.

The optimization of key-frame extraction leads to “objective’ (best possible) representation of a certain action-statistics of a shot, i.e. shot-content. As a consequence, we first have to address the question as “what is the best measure for the shot-content?”

By taking into account limited storage space, we can set the following requirement: The maximum number of extracted key-frames for certain amount of received video-data may be specified by the user. This leads to the optimization problem, presented schematically in Figure2:

After the maximum number N of key-frames for representing the whole video stream is given, first spread the total amount of N key-frames over all detected video-shots, where the number of assigned frames $L(i)$ per shot i is variable and dependent on shot’s content. In the next step, select $L(i)$ key-frames from the shot i , which best represent the shot-content. This is done for all detected shots in given video.

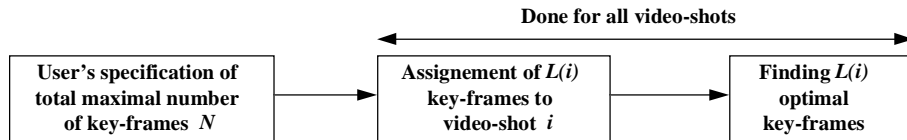


Figure 2: Optimization leading to the compromise between optimal key-frame based shot-representation and user’s and/or system specifications

In the following we present one idea of solving this problem, i.e. of implementing the optimization algorithm.

In order to be able to assign a certain number of frames to a shot depending on shot-content, a measure for that content must be defined first. Since the action within the shot is determined by differences between consecutive frames, it may be possible to derive the content-measure from accumulated frame-to-frame differences along the shot.

Figure 3a shows frame-to-frame differences (computed as differences in color-histograms) between each two consecutive I-frames of a MPEG1-sequence as well as accumulated differences along each detected shot. Shot changes with sharp peaks are clearly visible. Comparison shows that frame differences within the shot 3 (frames 17-23) are smaller than those by frames in the shot 4 (frames 24-31) by similar shot-length. It can be said that the content of the shot 4 is richer than the content of the shot 3. Since such content-relationship of these two shots can well be

represented by taking the area under the curve of the accumulated difference above the shot and related to the shot-length, we take in our approach this area, normalized by the number of frames in the shot, as a measure for the shot-content.

Once the allowed number of key-frames ($L(i)$) for the shot i is determined, an algorithm is started aiming to select $L(i)$ best possible frames for representing that shot. The main idea of this optimization can be seen in Figure 3b. Since the area mentioned above, defined as a measure for the shot-content, can be approximated through consecutive rectangles, each corresponding to one shot-frame and each having the height equal to the value of accumulated differences by that frame, the case of having only $L(i)$ frames can be treated similarly. For $L(i)$ key-frames per shot i $L(i)$ “sliding” rectangles with variable width are defined in such a way that their height is always equal to the accumulated-difference value in the middle of their base. After the algorithm has defined the optimal positions of rectangles and their width, so that they best “fill” the given area, central frames of rectangles are chosen to be key-frames for that shot. The rectangles used in this approach can be interpreted in such a way, that the central frame of each rectangle (selected key-frame) serves as representative of all frames included in that rectangle.

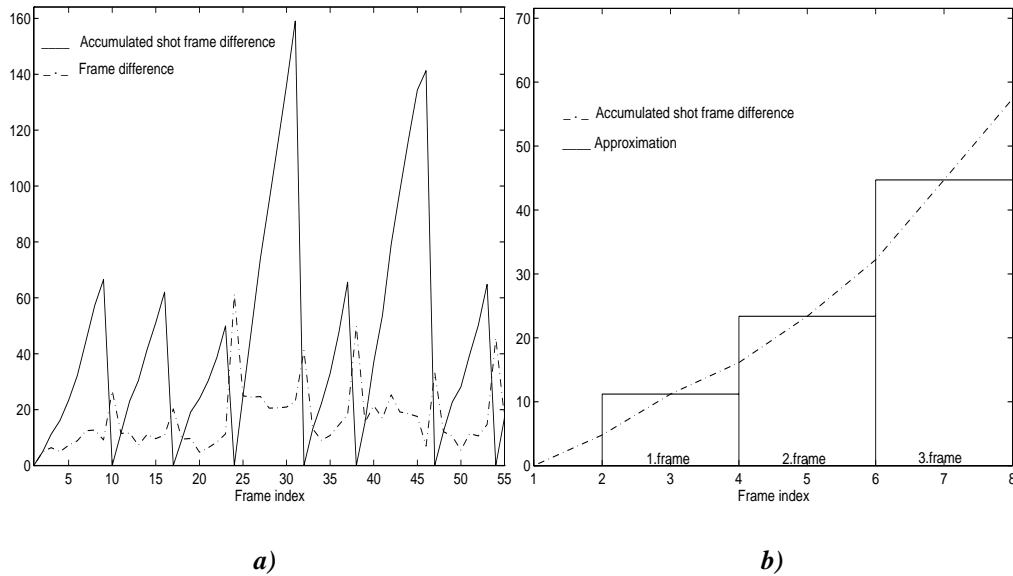


Figure 3a: Curve of differences between consecutive I-frames (dotted) and curve of accumulated differences for each detected shot

Figure 3b: One approximation of the curve of accumulated differences over the shot 3 through rectangles, each corresponding to one of 3 assigned key-frames

5 Summary and Conclusions

In this paper we pointed out to some problems concerning the search through stored video-data, which are to be solved - especially when applying the search procedure in a home-digital library. Those problems are automatization of the shot-change detection and optimization of the key-frame extraction by taking into consideration user’s specifications. Related to the second item, an optimization problem is posed and a novel idea for finding its solution is presented.

The main advantage of proposed optimization approach is that the key-frame extraction is based on analysis of the actual shot-content (after defining an appropriate content-measure) and

not on using thresholds. The result of the approach is a compromise between user's wishes and/or system characteristics on one side and optimal selection of key-frames for representing the shot-content.

References:

- [1] J. Meng, Y. Juan, S.-F. Chang: "Scene Change Detection in a MPEG Compressed Video Sequence", IS&T/SPIE Symposium Proceedings, Vol. 2419, San Jose, USA, February 1995
- [2] B. Yeo, B. Liu: "Rapid Scene Analysis on Compressed Video", IEEE Transactions on Circuits and Systems for Video technology, Vol.5, No.6, December 1995
- [3] A. Nagasaka, Y. Tanaka: "Automatic Video Indexing and Full-Video Search for Object Appearances", Visual Database Systems II (E. Knuth and L.M. Wegner, Eds.), pp.113-127, Elsevier,1992
- [4] B. Furth, S.W. Smoliar, H. Zhang: "Video and Image Processing in Multimedia Systems", Kluwer Academic Publishers, 1995
- [5] H.J. Zhang, A. Kankanhalli, S.W. Smoliar: "Automatic Partitioning of Full-Motion Video", ACM/Springer Multimedia Syst. 1(1), pp. 10-28, 1993
- [6] H. Zhang, C.Y. Low, S.W. Smoliar: "Video Parsing and Browsing Using Compressed Data", Multimedia Tools and Applications, 1, pp 89-111, Kluwer Academic Publishers, 1995.
- [7] M.M. Yeung, B. Yeo, W. Wolf, B. Liu: "Video Browsing using Clustering and Scene Transitions on Compressed Sequences", IS&T/SPIE Multimedia Computing and Networking, February 1995
- [8] M.M. Yeung, B. Liu: "Efficient Matching and Clustering of Video Shots", Proceedings of ICIP, Vol.1, pp 338-341, Washington, D.C., USA, 1995.
- [9] B. Yeo, B. Liu: "On the Extraction of DC Sequence from MPEG Compressed Video", Proceedings of ICIP, Vol.2, pp 260-264, Washington, D.C., USA, 1995.
- [10] U. Gargi, S. Oswald, D. Kosiba, S. Devadiga, R. Kasturi: "Evaluation of Video Sequence Indexing and Hierarchical Video Indexing", SPIE Conference on Storage & Retrieval for Image & Video Databases, Vol. 2420, 1995
- [11] G. Ahanger, T.D.C. Little: "A Survey of Technologies for Parsing and Indexing Digital Video", Journal of Visual Communication and Image Representation, Vol.7, No.1, pp.28-43, March 1996
- [12] R. W. Picard: "Light-Years from Lena: Video and Image Libraries of the Future", Proceedings of ICIP '95, Vol.1, pp. 310-313, Washington, D.C., USA, 1995