An Implementation Of a New Video Parsing Algorithm For Uncompressed Digital Video Stream

Halil Ibrahim Eskikurt
Sakarya University
Technical Education Faculty,
Sakarya, Turkey,
eskikurt@sakarya.edu.tr

Barıs Boru
Sakarya University
Technical Education Faculty,
Sakarya, Turkey,
barisb@sakarya.edu.tr

Abstract: Today, the use of digital videos instead of analog ones has become most popular in terms of their easy recordability and low-cost storage. Hence, huge amount of digital video archives have come out dramatically. The meaningful search in a video has been an important demand in order to access the video database very fast. While the standard picture and text searching methods cannot be used in a digital video, the video indexing has become a popular interest, and lots of research has been done. In this study, an implementation of a new video parsing algorithm design used for uncompressed videos is presented. Through the investigation of the parsing algorithms in the existing literature, lack of these algorithms has been defined and a new algorithm for achieving high performance has been proposed. It is been determined that the proposed algorithm is much better than the other algorithms considering the computational overhead and the performance.

Keywords: Video Parsing, Video Indexing

1. Introduction

With wide spread use of computers and digital equipments, and digital communication getting easier and more common to apply than the analogue communication through the platforms like internet, the need for transferring the video information to the digital medium has arisen, as it happened in all types of information.

Digital video is applied by benefiting from the weakness of the human eye. A human eye can only sense consecutive figures changing more than 15-20 frames in a second. A digital video is obtained by photographing a video scene as much as 15~30 frames in a second. The information about how many frames in a digital video are taken and recorded in a second is called "fps (frame per second)".

As mentioned above, digital video information is digital data coming one after another. If there is a sound record in a video, it consists of both picture frames and sound information.

2. Video Parsing and Indexing

As a result of the technological advances today, almost all videos are used by recording in a digital form. Since digital video records are increasing more and more, the need for fast access appears when a search is desired. For example, to get the information about the video records of a person or a vehicle in an archive of security videos recorded, the old records in the archive should be searched from beginning to end.

As the standard text search methods cannot be used in digital video, the picture comparison methods are not suitable as well, because of the need for computational overhead and the demands for the meaningful search [Cotsaces et al. 2006]. Therefore, special methods should be used for video access. Researchers dealing with video access have made a suggestion that instead of searching in the whole video, it would be better to search in a video index got by processing the video before, and then access to video segments referenced by the suitable results [Cotsaces et al. 2006, Koprinska et al. 2001].

If a subject title in a book is wanted to be accessed, the word can be found from the index or contents of the book and then the page referenced is directed. In a video wanted to make a search, scanning all the information every time would make the access as difficult as scanning the pages in a book one by one. For this reason, to search in a video and be able to access to a wanted scene, it is better to make an index for the video and store it to the special database (Fig. 1). Each video added to the video database could be saved after having an index with a number of procedures. When a browsing is made, the required object is scanned in the index. If a matching is found the video segment referenced is showed to the user.

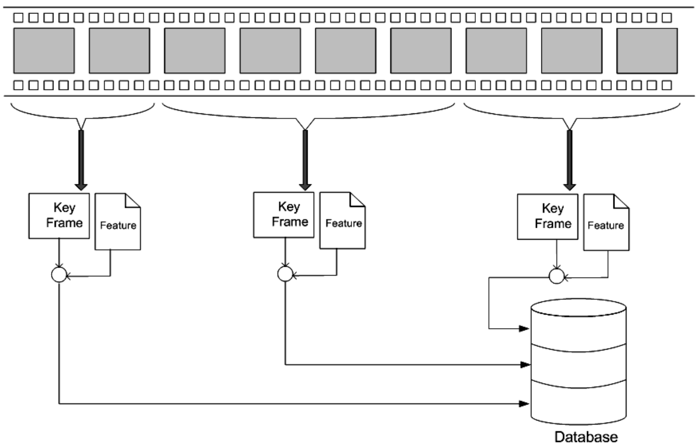

Figure 1: Parsing a video and storing the summarized information to the database by referring to the segments.

Video segments consist of consecutive frames having a meaningful integrity. Features and key frames defined for each segment constitute the video index. Video parsing, the most important section of video indexing, is the grouping of the frames having a meaningful integrity (Fig. 2). Thus, this group is defined with an information in the index. Also, it provides the preparation of defining the segments showed after browsing.

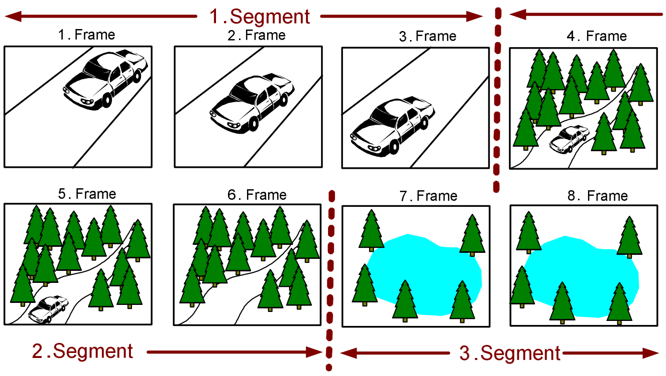

Figure 2: Examples of consecutive video frames and segments.

In the literature, most of researches about video databases and access can be found dealing with video parsing [Cotsaces et al. 2006, Koprinska et al. 2001]. The developed methods in these researches are focused on the right comprehending of video segments as defining the transitions in which there are meaningful changes on consecutive video frames.

There are two type segment transitions, such as gradual and sudden. Sudden segment transitions are usually formed by stopping the record of the camcorder like in filming and restarting for a different scene. Gradual segment transitions are the switching the scene contents gradually as happening by using to combine two segments in the film effects. The best examples of gradual transitions are fade-effect and dissolve effect. Because of the soft change instead of sharp change between consecutive frames, it is difficult to define gradual segment transition. These types of sensing should be separated from camcorder and object movements. Especially, it is too difficult to sense the segment transitions consisting of brightness level change compared to the other segment transitions.

Video parsing algorithms are usually making segment sensing by getting visual changing rates between consecutive frames based on the idea that there is a passing through the segments when the changing rate is high. Parsing algorithms start to browse from the first frame of video and evaluate the consecutive video frames. In some cases, even if there is no meaningful difference between two video frames, some big differences could appear when the numerical values were examined. The best examples of these types of changes are camcorder movement and brightness level change.

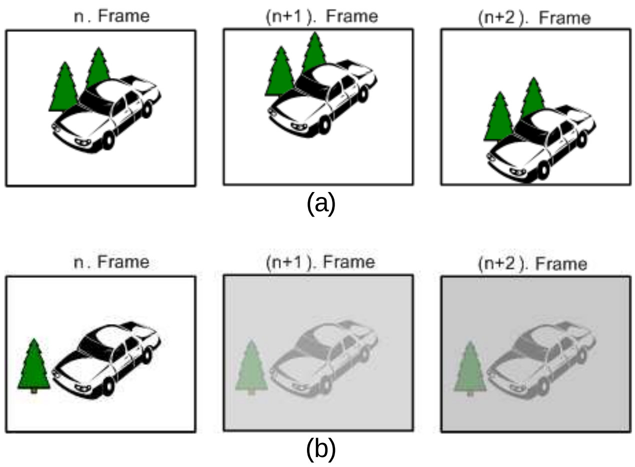

Figure 3: (a) An example of camcorder movements, (b) An example of brightness level changes.

During the filming, even though the contents of the scene are the same, the objects in the scene move a few pixels between two consecutive frames (Fig. 3.a). Similarly again, changing the light source and the camcorder viewing angle result in change in the darkness level of all pixels in the video frame (Fig. 3.b). Although, these are big changes in the pixel values of two consecutive frames, mostly it is not a segment transition. Therefore, parsing algorithms should have low sensibility to camcorder movement and brightness level change.

The parsing algorithms for uncompressed videos usually obtain a similarity rate between the consecutive frames due to the technique used. If the similarity rate of two consecutive frames is too low, it is accepted that there is segment distinction and the video parsing can be done. In these techniques, over which values for consecutive frame similarities can be assumed as a segment transition, are defined with a threshold value. In literature, these types of algorithms are called threshold based algorithms.

The algorithm having the simplest approach among the video parsing algorithms and being the fundamental of the other algorithms is the pixel comparison algorithm. This algorithm defines a change value for each video frame as calculating the total differences of the overlapped pixels between two consecutive video frames. Defined change value is compared with a predefined threshold value and then it is decided whether it is segment transition or not. Due to the higher computational overhead and the sensitivities of the camcorder movement and brightness level change, this simple method is not preferred today. Block based pixel comparison algorithm, developed as an alternative to the pixel comparison algorithm has been designed by [Kasturi et al. 1991] in a way that comparing the overlapped blocks with a similarity rate obtained from the pixel values and average darkness level values as dividing the video frame into blocks. This algorithm increased the computational overhead together with decreasing the sensibility of the brightness level change.

To decrease the computational overhead and the sensibility to the camcorder objects movement, Histogram based comparison methods instead of pixel based approaches have been developed. The main idea of histogram comparison methods is that there is not a big difference with the histograms of two consecutive frames which have unchanging background and objects (moving or not) [Koprinska et al. 2001]. In addition to that, histogram is not sensible to the picture rotation and the changing of the shooting angle. According to these principles, there are a number of studies related to histogram comparisons of the consecutive frames.

In the first histogram based parsing algorithm, the histograms of two consecutive frames have been compared and a segment transition approach has been applied by [Koprinska et al. 2001]. In literature, some amendments to this algorithm can be seen [Boreczky et al. 1996]. In these methods, although the sensibility to the camcorder and object movements is lower, the sensibility to the brightness level change is considerably high, because the brightness level change is completely replacing the histogram of the picture [Gargi et al. 1995].

Block based comparison techniques have been developed by [Swanberg et al. 1993], based upon the histogram comparison techniques as mentioned above. In this technique, a histogram comparison for the overlapped blocks in consecutive frames has been done by dividing the video frames into blocks. Due to the failure of these methods in gradual segment transitions, twin comparison technique has been developed by [Zhang et al. 1994]. In this technique, a second sub-threshold value has been used for sensing the differences between the frames occurred in gradual segment transition and then the values above this sub-threshold value and the differences between the consecutive frames have been added. If the result is higher than the real threshold value, gradual segment transition could be defined. Boreczky and Rowe have decided that twin comparison technique was simple and reliable [Boreczky et al. 1996].

3. The Proposed Algorithm for the Filtered Video Histogram Comparison

The success of parsing algorithms on uncompressed videos has been defined as low level. These algorithms have better sensibilities to the brightness level change and camcorder movements. This work introduces a new algorithm for filtered video histogram comparison to eliminate the disadvantages in the histogram based methods. The video parsing process have benefited from the picture filtering techniques to be more fast and functional. After the whole video stream has been filtered with the picture filtering techniques, it is subjected to histogram comparison.

The main idea of designing the algorithm is to consider pixel movement, occurred from camcorder movement, and to interest with the objects in the scene. Consequently, motion blur and sobel filters have been applied for picture processing. Sobel picture filter has put forward the section, having sharp colour change in the filtered picture. In other words, sobel filter makes the object edges in the picture clear.

To lower the sensibility to the camcorder movements, it is foreseen that the neighbour pixels are copied into themselves with the specified weights. Hence, sharp edge values obtained by sobel filter are being softened and the effects of sections (background and object edge sections) near to the edges are investigated. For this purpose, motion blur filter has been used. With this method following object edge moving, videos have been processed by converting black-white video. The steps of filtering process are shown in Fig. 4.

If the standard formulas of histogram comparison are used to obtain the difference values of consecutive frames, the flat areas, not interested in this method, will be causing false results to increase. To remove this disadvantage, the formula (Equation 1) used in the filtered video histogram comparison method, has been obtained with neglecting "0" value representing black colour in the standard histogram comparison formula. The formula in Equation 1, can be used in digital videos having an 8-bit colour darkness value (darkness values: 0-255, n=256). D value shows the difference value between i and i+1 frames, H value shows the histograms of the corresponding frames and j value shows the darkness values. In the formula, the reason for the darkness values between 1 and 255, as mentioned above, is to neglect the darkness values representing the flat areas in the filtered histograms.

$$D(i,i+1) = \sum_{j=1}^{255} \left| H_i(j) - H_{i+1}(j) \right| \quad (1)$$

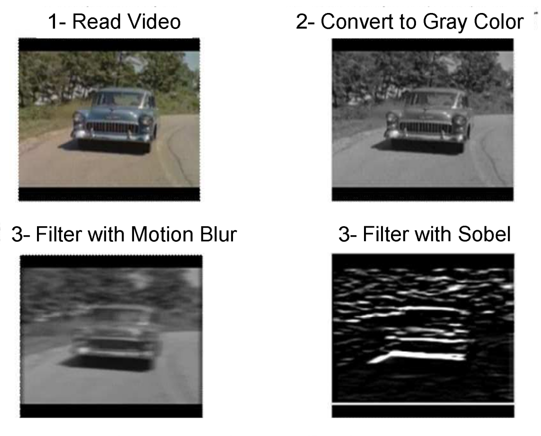

Figure 4: Picture filtering steps.

The working steps for the Filtered Video Histogram Comparison method can be summarized as follows:

1- Making the video gray shading.
2- Filtering the video with Motion Blur Filter.
3- Filtering the video with Sobel Filter.
4- Getting the difference values of consecutive frames with histogram comparison, while neglecting the black components of the filtered video ("0" darkness value).
5- Defining the frames over the threshold value as a segment transition.
6- Making the segments generated using the defined segment transitions.

4. Evaluation and Comparisons

When the results, obtained from the application of proposed algorithm evaluated, it is easy to say that the sensibility to camcorder movements and brightness level changes is much less compared to other uncompressed video segmentation algorithms. Test with several videos showed that especially sensibility to brightness level changes reduced much and videos including camcorder lighting changes have been segmented with very high accuracy.

In Fig. 5, sequential frame differences, resulted by HSV Histogram Comparison and Filtered Video Histogram Comparison for a video including lots of brightness level changes and camcorder movements, has been shown with graphics. HSV Histogram Comparison was chosen for comparing proposed algorithm because it has less sensibility in standard algorithms. Real segment transitions are shown in graphics with an arrow.

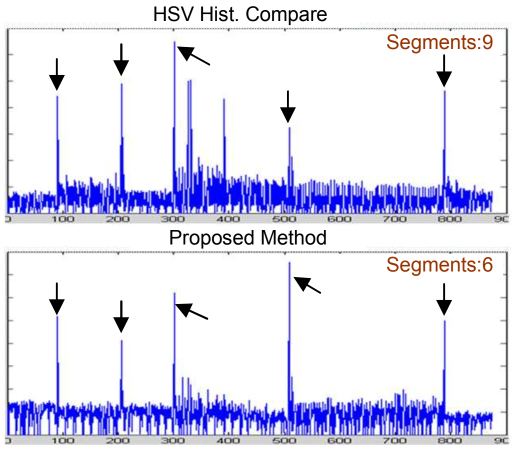

Figure 5: Computed consecutive frame differences for a sample video.

Video, used for comparison, is recorded specially with a poor camera and it has camcorder motions, brightness level changes also it consists of 6 segments. The fourth segment of sample video (between 300th and 515th frames) has lots of brightness level changes and camcorder motions. In the graphics it is shown that, the differences computed with HSV Histogram Comparison algorithm for fourth segment's brightness level changes is higher than real segment transitions. In addition, HSV Histogram Comparison computed difference level for camcorder motion just after the starting of 5th segment as near to real transition difference level.

For measuring of the proposed algorithm, all algorithms mentioned above have been applied to several videos [NIST 2009]. After getting results it is observed that proposed algorithm has better performance than the other algorithms. In Tab. 1, comparison results for Chevrolet.avi [NIST 2009] (including 5 segments) video are given. In results, computing times, number of segments detected by algorithm, accuracy rate for real segment transitions, and extra (unnecessary) number of segments detected have been shown. For the best performance it is expected to have minimum computing time, best accuracy rate (%100), and not to detect any other extra segments.

For comparison of computing overhead, it is clear that filtering steps used in Filtered Video Histogram Comparison algorithm are making computation time longer. But in the computation part of the frame differences, the proposed method computes faster in rate 1/3 because of using only gray scale darkness level. Other methods use Red, Green, Blue levels for computing consecutive frame differences. For this reason their computing times are much longer than the proposed method.

The interface shown in Fig. 6 has been designed to apply the algorithms, include the proposed and existing algorithms. With this interface, a video loaded can be divided into sections according to the parsing method and then these segments can be stored for access.

ALGORITHM	Time (s)	Number of Segment	Accuracy (%)	Extra Segment
Pixel Comparison	153.41	7	100	2
Block Based Pixel Comparison	212.23	9	80	5

RGB Histogram Comparison	2.74	8	40	6
Block Based Histogram Comparison	27,09	7	60	4
HSV Histogram Comparison	34.98	8	100	3
Filtered Video Histogram Comp.	17.67	5	100	0

Table 1: Comparison results for Chevrolet.avi video.

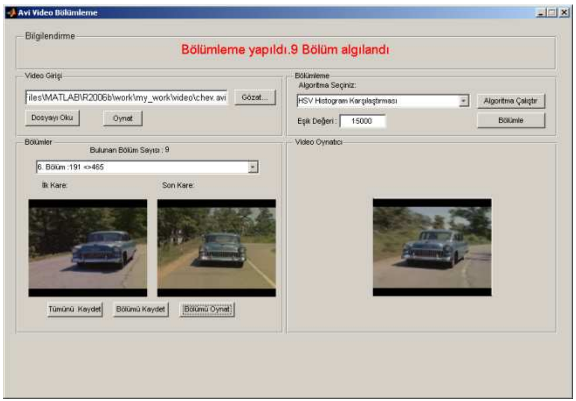

Figure 6: Parsing interface.

5. Conclusion

In this work, a new parsing algorithm design has been proposed which has superiority over the existing parsing algorithms, used with uncompressed videos. In literature, existing algorithms has been investigated and the lack of the algorithms have been defined. The new method has been applied to a number of example video streams. Also the other algorithms have been applied to the same video streams and then a comparison of these algorithms has been made. As a result of the comparison, it is seen that the algorithm has a better segment transition, and an acceptable level of computational overhead.

References

Cotsaces C., & Nikolaidis N., Pitas I. (2006). *Video shot detection and condensed representation – A review*, IEEE Signal Processing Magazine, vol.23, iss.2, 28-37.

Koprinska I., & Carrato S. (2001). *Temporal Video Segmentation: A Survey*, Elsevier Science, Signal Processing: Image Communication, vol.16, iss.5, 477-500.

Kasturi R., & Jain R. (1991). *Dynamic Vision, in Computer Vision: Principles* IEEE Computer Society Press, Washington DC, 469-480.

Zhang H.J., Low C.Y., & Smoliar S.W. (1994). *Video Parsing and Browsing Using Compressed Data*, in Proceedings of SPIE Conf. Image and Video Processing II, 142-149.

Gargi U., Oswald S., Kosiba S., Devadiga S., & Kasturi R. (1995). *Evaluation Of Video Sequence Indexing And Hierarchical Video Indexing*, in Proceedings of SPIE Conference on Storage and Retrieval in Image and Video Databases, 1522-1530.
Swanberg D., Shu C. F., & Jain R. (1993). *Knowledge guided parsing in video databases*, in Proceedings of SPIE Conference, vol.1908, 13-24.

Boreczky J. S., & Rowe L. A. (1996). *Comparison of Video Shot Boundary Detection Techniques*, in Proceedings of IS&T/SPIE International Symposium on Electronic Imaging, vol.2670, 170-179.

NIST National Institute of Standards and Technology (2009). *TREC Video Retrieval Evaluation*, http://trecvid.nist.gov/, USA.