Video Segmentation: A Critical Survey

M. R. Banwaskar and A. M. Rajurkar
Department of Computer Science
M. G. M.'s College of Engineering Nanded, Maharashtra, India

Abstract—With recent advances in multimedia technologies, digital TV and information highways, more and more video data is being captured, produced and stored. However, without appropriate techniques that can make the video content more accessible, all these data are hardly usable. Content Based Video Retrieval (CBVR) becomes a proper solution to handling the video databases. The essential first step in CBVR is Video Segmentation. This paper is a critical survey of current trends/ methods for video segmentation. This work has been done with an aim to assist the upcoming researchers in the field of video retrieval to know about the technology and methods available for video segmentation.

Keywords-video segmentation; content-based video retrieval; video indexing;key frame; histogram

1 Introduction

Digital video has become an emerging force in current computer and telecommunication industries for its large mass of data. The development of video compression technology facilitates to the widespread use and availability of digital videos. Expanding applications such as digital libraries, video-on demand, digital video TV/broadcast, multimedia information system have motivated the growing demand of new technologies for efficient retrieval of video data. The area of content-based video retrieval has attracted extensive research during past few years [1-4]. Fig. 1 shows the general structure of content-based retrieval of video databases. To achieve automatic video content analysis, an essential first step is the segmentation of video. It is also a hotspot in video processing technology.

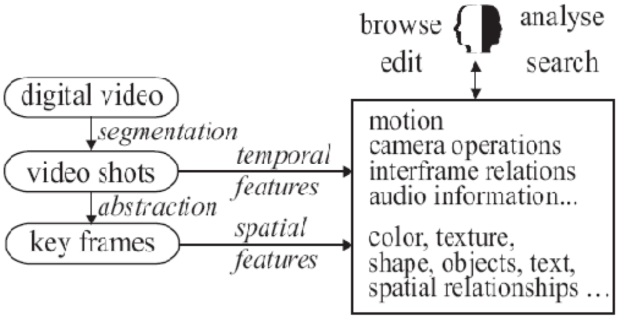

Figure 1. Content-based retrieval of video databases.

This research area is gaining attention from several research communities including image processing, computer vision, pattern recognition and artificial intelligence. The newly established multimedia standards are based on the object content of videos [5]. Therefore, successful representation and processing in these standards require efficient segmentation algorithms.

Vide Video segmentation refers to partitioning video into spatial, temporal, or spatiotemporal regions that are homogeneous in some feature space. As with any segmentation problem, effective video segmentation requires proper feature selection and an appropriate distance measure. Different features and homogeneity criteria generally lead to different segmentations of the same video, for example, color, texture, or motion segmentation.

Specific video segmentation methods should be considered in the context of the requirements of the application in which they are used. Factors that affect the choice of a specific segmentation method include the following:

1.1 Real-time Performance

If segmentation must be performed in real time, for example, for rate control in video telephony, then simple algorithms that are fully automatic must be used. On the other hand, one can employ semiautomatic, interactive algorithms for off-line applications such as video indexing or off-line video coding to obtain semantically meaningful segmentations.

1.2 Precision of segmentation

If segmentation is employed to improve the compression efficiency or rate control, then certain misalignment between segmentation results and actual object borders may not be of concern. On the other hand, if segmentation is needed for object-based video editing or shape similarity matching, then it is of utmost importance that the estimated boundaries align with actual object boundaries perfectly, where even a single pixel error may not be tolerable.

1.3 Scene complexity

Complexity of video content can be modeled in terms of amount of camera motion, color and texture uniformity within objects, contrast between objects, smoothness of motion of objects, objects entering and leaving the scene, regularity of object shape along the temporal dimension, frequency of cuts and special effects, etc. Clearly, more complex scenes require more sophisticated segmentation algorithms. For example, it is easier to detect cuts than special effects such as wipes or fades

.

The goal of this paper is to provide a comprehensive taxonomy and critical survey of the existing approaches for video segmentation. The performance, relative merits and shortcomings of some approaches are discussed in detail. The paper is organized as follows. Section I presents basic-state-of-the-art knowledge of video segmentation. In Section II, video segmentation in uncompressed and compressed domain is addressed. In section III we review few more recent approaches for video segmentation and finally we summarize to conclude the paper in section IV giving few future directions.

2 Basic-state- of- the-art knowledge

In order to analyze a video sequence, it is necessary to break it down into meaningful units that are of smaller length and have some semantic coherence. The unstructured and linear features of video introduce difficulties for end users in accessing the knowledge captured in videos. To extract the knowledge structures and make it easily accessible, it is necessary to segment the video into shorter scenes. Video segmentation can be defined as grouping of the shots into clusters based on certain criteria like space or time. It is the process of typically splitting a video stream into segments at scene changes.

A *shot* is defined as an unbroken sequence of frames taken from one camera. There are two basic types of shot transitions: *abrupt* and *gradual*. *Abrupt* transitions (*cuts*) are simpler, they occur in a single frame when stopping and restarting the camera. Although many kinds of cinematic effects could be applied to artificially combine two shots, and thus to create gradual transitions, most often *fades* and *dissolves* are used. A *fade out* is a slow decrease in brightness resulting in a black frame; a *fade in* is a gradual increase in intensity starting from a black image. *Dissolves* show one image superimposed on the other as the frames of the first shot get dimmer and those of the second one get brighter. Gradual transitions are more difficult to detect than cuts. They must be distinguished from camera operations and object movement that exhibit temporal variances of the same order and cause false positives. It is particularly difficult to detect dissolves between sequences involving intensive motion.

A fast and automatic video segmentation technique based on the video object's semantic similarity is proposed in [6]. This technique aims at foreground and background segmentation via effective combination of color and motion analysis module. The general outline of this segmentation is illustrated in fig.2

In this technique input video is preprocessed using Median filter to reduce the effect of Gaussian noise and binary noise.

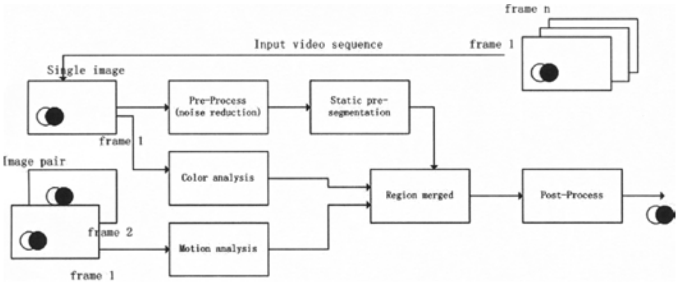

Figure 2. Video segmentation system architecture

Then static segmentation based on watershed algorithm is used and the regions are merged according to their color and motion similarity. Finally, post-processing is done to the regions using mathematical morphology in order to smooth the object boundaries.

3 Video segmentation in uncompressed domain

The majority of algorithms process uncompressed video. Usually, a similarity measure between successive images is defined. When two images are sufficiently dissimilar, there may be a cut. Gradual transitions are found by using cumulative difference measures and more sophisticated thresholding schemes. Based on the metrics used to detect the difference between successive frames, the algorithms can be divided broadly into three categories: pixel, block-based and histogram comparisons.

3.1 Pixel Comparison

Pair-wise pixel comparison (also called template matching) evaluates the differences in intensity or color values of corresponding pixels in two successive frames. The simplest way is to calculate the absolute sum of pixel differences and compare it against a threshold [7]. The main disadvantage of this method is that it is not able to distinguish between a large change in a small area and a small change in a large area. For example, cuts are misdetected when a small part of the frame undergoes a large, rapid change. Therefore, methods based on simple pixel comparison are sensitive to object and camera movements. A possible improvement is to count the number of pixels that change in value more than some threshold and to compare the total against a second threshold [8, 9]. Although some irrelevant frame differences are filtered out, these approaches are still sensitive to object and camera movements. For example, if camera pans, a large number of pixels can be judged as changed, even though there is actually a shift with a few pixels. It is possible to reduce this effect to a certain extent by the application of a smoothing filter. Each pixel is replaced by the mean value of its neighbors before comparison.

3.2 Block-based comparison

In contrast to template matching that is based on global image characteristic (pixel by pixel differences), block-based approaches use local characteristic to increase the robustness to camera and object movement. Each frame *i* is divided into *b* blocks that are compared with their corresponding blocks in *i+1*. Compared to template matching, this method is more tolerant to slow and small object motion from frame to frame. On the other hand, it is slower due to the complexity of the statistical formulas. Additional potential disadvantage is that no change will be detected in the case of two corresponding blocks that are different but have the same density function. Such situations, however, are very unlikely.

3.3 Histogram comparison

A step further towards reducing sensitivity to camera and object movements can be done by comparing the histograms of successive images. The idea behind histogram-based approaches is that two frames with unchanging background and unchanging (although moving) objects will have little difference in their histograms. In addition, histograms are invariant to image rotation and change slowly under the variations of viewing angle and scale [10]. As a disadvantage one can note that two images with similar histograms may have completely different content. However, the probability for such events is low enough, moreover techniques for dealing with this problem have been proposed in [11].

4 Video segmentation in compressed domain

The previous approaches for video segmentation process uncompressed video. As nowadays video is increasingly stored and moved in compressed format (e.g. MPEG), it is highly desirable to develop methods that can operate directly on the encoded stream. Working in the compressed domain offers the following advantages. First, by not having to perform decoding/re-encoding, computational complexity is reduced and savings on decompression time and storage are obtained. Second, operations are faster due to the lower data rate of compressed video. Last but not least, the encoded video stream already contains a rich set of pre-computed features, such as motion vectors (MVs) and block averages, that are suitable for temporal video segmentation. Several algorithms for temporal video segmentation in the compressed domain have been reported.

According to the type of information used, they can be divided into six non-overlapping groups - segmentation based on: 1) DCT coefficients; 2) DC terms; 3) DC terms, macroblock (MB) coding mode and MVs; 4) DCT coefficients, MB coding mode and MVs; 5) MB coding mode and MVs and 6) MB coding mode and bitrate information.

The pioneering work on video parsing directly in compressed domain is conducted by Arman, Hsu and Chiu [12] who proposed a technique for cut detection based on the DCT coefficients of I frames.

Zhang *et al.* [13] apply a pair-wise comparison technique to the DCT coefficients of corresponding blocks of video frames. Both of the above algorithms may be applied only to I frames of the MPEG compressed video, as they are the frames fully encoded with DCT coefficients. As a result, the processing time is significantly reduced but the temporal resolution is low. In addition, due to the loss of the resolution between the I frames, false positives are introduced and, hence, the classification accuracy decreases. Also, neither of the two algorithms can handle gradual transitions or false positives introduced by camera operations and object motion.

5 Review and discussion

A generic approach for managing video data is first to segment a video into groups of related frames called shots by means of shot detection or scene break detection [14-18]. Various approaches for shot boundary detection have been proposed [19-24]. Some of the existing methods are designed based on the fact that the frames within the same shot maintain some consistency in the visual content. In most of the algorithms, firstly a measure was adopted to quantify the degree of dissimilarity between the two frames. Then the scheme for determining the decision threshold was based on the assumption that the dissimilarity measure comes from one of the two cases: one for shot boundaries and one for not-a-shot-boundary. When the dissimilarity surpassed the given threshold, a shot boundary is declared. Wenwei Tan et. al. [25] [ICVGIP]used color histogram for representing the variation of visual content in frames. The difference from I frame to i+1 frame is computed as follows:

$$D_R = \frac{\sum_{k=1}^{M}\sum_{j=1}^{N}\min\left(R_i(j,k),R_{i+1}(j,k)\right)}{M \times N} \text{ ……. (1)}$$

$$D_G = \frac{\sum_{k=1}^{M}\sum_{j=1}^{N}\min\left(G_i(j,k),G_{i+1}(j,k)\right)}{M \times N} \text{ …… (2)}$$

$$D_B = \frac{\sum_{k=1}^{M}\sum_{j=1}^{N}\min\left(B_i(j,k),B_{i+1}(j,k)\right)}{M \times N} \text{ …….. (3)}$$

$$DCH = \frac{1}{3}\left(D_R + D_G + D_B\right)\text{……………..(4)}$$

Where R_i(j,k), G_i(j,k) and B_i(j,k) denote the Red, Green and Blue channel histogram value at color level j for the region (block) k respectively, M is the total number of blocks and N is the total number of color levels. DCH is the sum value of histogram intersection in interval [0,1]. For two identical histograms the DCH is 1, while the the two frames which do not share even a single pixel of the same color (bin), the DCH is 0.

Patrick Ndjiki-Nya et. al. [26] have proposed a spatio-temporal video segmentation framework. This approach corresponds to a split and merges segmentation strategy with tracking abilities. However, this algorithm is prone to over-segmentation, which may be explained by a too rigorous merger criterion. Error rate of this method is also high. Hence, long term motion analysis and a more efficient exploitation of available motion information for tracking via corresponding MPEG-7 descriptors have to be considered.

The average histogram is constructed by accumulating all the pixel values from all frames within a group of frames, i.e.,

$$\text{AH(j)}=\frac{1}{M}\sum_{i=1}^{M} H_i(j), \quad \text{for j= 1,……B} \tag{5}$$

Where H_i denotes the histogram of the ith frame, B is the number of bins in the histogram, and M is the number of frames in the group of frames. However, the representation power of the average histogram may be corrupted by the outlier frames within a shot. A more robust representation is the median histogram, which is formed by choosing the medium values for each of the corresponding histogram bins.

For MPEG-7, a family of alpha- trimmed average histogram has been proposed as a robust color histogram descriptor. This approach considers all the corresponding histogram bin values from all the frames within the group of frame, and is generated using the trimmed mean operator. To obtain an alpha-trimmed average histogram, the corresponding bin values are stored in either ascending or descending order, and then the average value for each bin is computed from the central members of the ordered array.

[27] proposed a new representative scheme, namely Temporally Maximum Occurrence Frame (TMOF), for video retrieval. This TOMF can capture the most significant visual contents within a video shot. The representational power of the TMOF is further enhanced by considering the k most frequently occurring values and the k highest peaks of the probability distribution at each of its pixel position. This method outperforms the alpha-trimmed average histogram representation for video retrieval. Jiri Filip and Michal Haindl have proposed a novel PCA-based approach to temporal segmentation of video sequences [28]. This method provides reliable detection of cuts and promising detection of dissolve transitions in video sequences.

A. Vadivel et al. have proposed a temporal video segmentation method using color-texture histogram generated from HSV color space with the help of soft decision [29]. They have developed web based video segmentation and retrieval system that is available free for use. The distinguishing feature of this method is that the users can load their own video clippings, which can be processed by this system for on line video segmentation and subsequent retrieval of images. The authors claim high precision in their approach due to the use of soft decision in combining color and texture features.

A more recent method for foreground-background video segmentation in real time is proposed by [30]. The main contribution of this work is the definition of likelihood function which is robust to illumination changes, casted shadows and camouflage situations. This method makes use of Quadratic Markov Measure Fields models for binary video segmentation and a parallel implementation of segmentation algorithm that executes in real time.

A work by [31] et al. provides the first transductive segmentation of live video with non-stationary background. In this method segmentation is propagated based on local color models and temporal prior, as well as a dynamic global color model in case of occlusion. This approach also uses a geodesic-based to solve for the final segmentation by incorporating smooth prior and image contrast which is capable of dealing with larger size of input image sequence in real-time. The limitation of this method is that it makes use of only color information. If other cues like shape and texture are accounted for then segmentation quality can be improved.

An effective traffic monitoring video segmentation is proposed in [32]. This method implements three algorithms: background registration algorithm, object detection algorithm and post processing algorithm. The authors claim that they got ideal segmentation results and the speed of the algorithm can meet real time requirement. However, our study reveals that background registration technique cannot make an ideal background model with the impact of camera motion, also shadow elimination cannot deal with strong light changing.

[33] proposed motion based approach to detect the foreground and combine luminance and chromaticity factors to refine the result form compressed domain.

6 Conclusions

With increasing proliferation of digital video contents, efficient techniques for analysis, indexing and retrieval of videos according to their contents have become evermore important. A common step for content-based video analysis techniques available is to segment a video into elementary shots. In this paper we have reviewed existing methods for video segmentation. The majority of algorithms process uncompressed video. Since video is stored in compressed format, several algorithms operate directly on compressed video. The evaluation of various algorithms depends on the type of application. In case of gradual transition detection, an important evaluation criteria is the algorithm’s ability to determine exactly between which frames the transition occurs and to classify the type of the transition (dissolve, fade, etc.). Other essential issues are the sensitivity to the encoder's type and the ease of implementation. Probably the best way for comparison and testing of the different temporal video segmentation techniques is to build a repository that contains Web-executable versions of the algorithms as suggested in [34]. It could be done by either providing a Web interface to the algorithms or by implementing them in a platform-independent language.

7 References

[1] Paul Over, “TRECVID 2005-An overview”, National Institute of Standards and Technology and City University, 2006.

[2] A. F. Smeaton,” Techniques used and open challenges to the analysis, indexing and retrieval of digital video information systems”. 2006.

[3] Haoran Yi, Deepu Rajan, Liang-tien Chia,” A motion based scene tree for browsing and retrieval of compressed videos”, Information Systems 31(2006)638-658.

[4] Cotsaces, Nikos Nikolaidis, “Video shot detection and condensed representation”, IEEE Signal Processing Magazine, Mar. 2006.

[5] P. Salembier, “Overview of the MPEG-7 standard and future challenges for visual information analysis”, EURASIP J. Appl. Signal Process., vol. 4, pp. 343-353, Apr. 2002.

[6] G. Chunsheng, F. Zejun, “BP algorithm for background estimation based on temporal and spatial correlation”, 2010 3rd International Congress on Image and Signal Processing, pp. 308-311.

[7] T. Kikukawa, S. Kawafuchi,” Development of an automatic summary editing system for the audio-visual resources”, Trans. on Electronics and Information J75-A (1992) 204-212.

[8] A. Nagasaka, Y. Tanaka, “Automatic video indexing and full-video search for object appearances, in visual database Systems II” (E. Knuth and L.M. Wegner, eds.), pp. 113-127, Elsevier, 1995.

[9] H.J. Zhang, A. Kankanhalli, S.W. Smoliar, “Automatic partitioning of full-motion video”, Multimedia Systems 1(1) (1993) 10-28.

[10]M. J. Swain, “Interactive indexing into image databases”, Proc. SPIE Conf. Storage and Retrieval in Image and Video Databases, 1993, pp.173-187.

[11] T. N. Pappas, “An adaptive clustering algorithm for image segmentation”, IEEE Trans. on Signal Processing 40 (1992) 901-914.

[12] F. Arman, A. Hsu, M-Y Chiu, “Image processing on compressed data for large video databases”, Proc. of First ACM Intern.Conference on Multimedia, 1993, pp. 267-272.

[13] H.J. Zhang, C.Y. Low, Y.H. Gong, S.W. Smoliar, “Video parsing using compressed data”, Proc. of SPIE Conf. Image and Video Processing II, 1994, pp. 142-149.

[14] K. W. Sze, K. M. Lam, and G. Qui, “Scene cut detection using the colored pattern appearance model”, Proc. of IEEE Int. Conf. Image Processing, Barcelona, Spain, Sep. 2003, pp. 1017-1020.

[15] C. –L. Huang and B. –Y. Liao, ”A robust scene-change detection method for video vetection”, IEEE Transactions on Circuits Systems. Video Technol., vol. 11, no. 12, pp 1281-1288, Dec. 2001.

[16] J. Yu and M. D. Srinath, “ An efficient method for scene cut detection”, Pattern Recogn. Lett., vol. 22, no. 13, pp. 1379-1391, Nov. 2001.

[17] M. S. Lee, Y. M. Yang, and S. W. Lee, “Automatic video parsing using shot boundary detection and camera operation analysis”, Pattern Recognit. vol. 34, no. 3, pp. 711-719, Mar. 2001.

[18] S. –W. Lee, Y. –M. Kim, and S. W. Choi, “Fast scene change detection using direct feature extraction from MPEG compressed videos”, IEEE Trans. Multimedia, vol. 2, no. 12, pp. 240-254, Dec. 2000.

[19] Alan Hanjalic, “Shot- boundary detection unraveled and resolved”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 12, no.2, Feb. 2002.

[20] Hong Lu, Yap-Peng Tan, “An effective post-refinement method for shot boundary detection”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 15,no.11,pp.1407-1421, Feb. 2005.

[21] Giuseppe Boccignone, Angelo Chianese, “Foveated shot detection for video segmentation”, IEEE Transactions on Circuits and Systems for Video Technology, vol. 15,no.3,pp.365-377, Mar. 2005.

[22] A. Vadivel, M. Mohan, “Object level frame comparison for video shot detection”, Proc. of IEEE Workshop on Motion and Video Computing, 2005.

[23] Chen Cai, Zheng Tan, ”An efficient video shot representation for fast video retrieval”, Visual Communications and Image Processing 2005, Proc. Of SPIE vol. 5960.

[24] Hun-woo Yoo, Han-jin Ryo, “Gradual shot boundary detection using localized edge bloks”, Multimedia Tools and Applications 2006, pp.283-300.

[25] Wenwei Tan, Shaohua Teng, ”Research on video segmentation via active learning”, IEEE Int. Conf. on Image and Graphics.

[26] Patrick Ndjiki-Nya, Sebastian Gerke, “Improved video segmentation through robust statistics and MPEG-7 features, Proc. ICASSP pp.777-780, 2009.

[27] K. W. Sze, K. M. Lam, and G. Qiu, “ A new key frame representation for video segment retrieval”, IEEE Transactions on Circuits and Systems for Video Technology, vol.15, no.9, Sept.2005.

[28] Jiri Filip, Michal Haindl, “ Fast and reliable PCA-based temporal segmentation of video sequences” .

[29] A. Vadivel, Shamic Sural, A. K. Majumdar, “Temporal video segmentation using a color-texture histogram ”, Int. J. Signal and Imaging Systems Engineering, Vol. 1, No. 1, pp. 78-87.

[30] Francisco J. Hermandez-Lopez and Mariano Rivera, “Binary segmentation of video sequences in real time”, Ninth Mexican International Conference on Artificial Intelligence, 2010, pp. 163-168,

[31] F. Zhong, X. Qin, Q. Peng, “Transductive segmentation of live video with non-stationary background”, pp. 2189-2196

[32] L. Chen, Xu Liang, X. Wang, “An effective traffic monitoring video segmentation method”, 2010 International Conference on Computer Design and Applications, vol, 5, pp. 593-597.

[33] Z. Leiqi, Z. Qishan, “Motion-based foreground extraction in compressed video”, 2010 International Conference on Measuring Technology and Mechatronics Automation, pp. 711-714.

[34]U. Gargi, R. Kasturi, S. Antani, “Performance characterization and comparison of video indexing algorithms”, Proc. of Conf. Computer Vision and Pattern Recognition (CVPR), 1998.