A survey on Characteristics and Parameters of frame to detect Shot Transition and various Shot Detection Techniques

Patel Raunakraj L.
M.E. (Computer science & Engineering)
Computer Science and Engineering, L.D. College of Engineering, Ahmedabad , Gujarat

***Abstract* - In this technological world the digital data and storage has taken a prime place in our routine applications. It contains a large amount of data including videos and features extracted from it. Designing systems which can manage this extensive data and make it easily accessible for search and query, which is in text form, is a very challenging and potentially rewarding problem. However, the vast majority of research in video indexing and retrieval techniques has taken place in the field of multimedia. Like for authored or produced video such as news or movies, and spontaneous and broadcast video such as sporting events. This paper mainly focuses on the analysis of various shot boundary detection methods and features those are used to detect it. Shot boundary detection is the fundamental step in the content based video retrieval. It is even a major research issue, since this has been used as an important parameter in the video retrieval process. Here paper is divided into sections. Starting with introduction to shot boundary detection, than we have section contains features which are used in classification of shot. Next section includes popular shot boundary detection algorithms and Techniques. Lastly we have formal comparison of various video shot boundary techniques and observations.**

Keyword – Shot detection, Content based video retrieval

I. Introduction

Large volume of online video data is available today. People are interested to search specific information only from videos. It is hard to surf the big database and whole volume of video. As structure of video data is unsuitable for traditional text based or query based form. So, Content Based Video Retrieval (CBVR) methods are used to search specific content. CBVR mainly contains three major steps, first segmenting the video, second feature extraction from it and third retrieving a video based on information query. So segmentation of video is basic and very crucial step in this method. To understand segmentation process, it is necessary to know the structure of video; where segmentation is used to divide video into shots.

Videos are structure from frames, shots, scenes, video clips in ascending order. Frames are stills or simply pictures we can say. Shot is unbroken sequence of frames from a single camera. Thus shot is having strong content correlation between frames of it. A scene can be explained by one or more adjoining shots that are focused on particular object or group of objects. For example in cricket sport, batsman hitting a boundary may be captured by different angles and cameras which can show batsman’s shot style and trajectory of ball. Now two cameras can show different path for ball which are different shots but they belongs to the same scene.

Shot boundary is generated due to transaction in scene. These transactions can be smooth or abrupt. Abrupt transaction is instantaneous transaction between two subsequent shots. So a hard cut is an abrupt shot change occurs in one shot.

Smooth or gradual transaction occurs over multiple frames which can be classified as below.

1. Fade: A fade is a slow change in brightness of frames. Two different kinds of fades are used: Fade-out when the image fades to a black screen or a dot. And Fade-in when the image is displayed from a black image.
2. Dissolve: A dissolve occurs when the images of the first shot get dimmer and the images of the second shot get brighter, with frames within the transition showing one image superimposed on the other. So it looks like both frames are dissolving into each other.
3. Wipe: A wipe is a virtual line going across the screen clearing the old scene and displaying a new scene. So pixels from second shot replace related pixels from current shot in certain pattern.

II. Literature review

Detection of shot boundaries is basic and first step in video - retrieval, analysis, indexing. So Shot Boundary Detection (SBD) is necessary for all these applications and thus it is studied for a long time and researched widely. Still an algorithm which can run on vast video category is missing, as many algorithms are able to work on specified bunch of videos. Following are some popular and well accepted methods.

Pixel Differences

The easiest way to detect if two frames are significantly different is to count the number of pixels that change in value more than some threshold. This total is compared against a second threshold to determine if a shot boundary has been found. This method is sensitive to camera motion. Zhang, Kankanhalli, and Smoliar[1] implemented this method with the additional step of using 333 averaging filter before the comparison to reduce camera motion and noise effects. They found that by selecting a threshold tailored to the input sequence good results were obtained, although the method was somewhat slow. We note that manually adjusting the threshold is unlikely to be practical. Shahraray[2] divided the images into 12 regions and found the best match for each region in a neighbourhood around the region in the other image. This matching process duplicates the process used to extract motion vectors from an image pair. The pixel differences for each region were sorted, and the weighted sum of the sorted region differences provided the image difference measure. Gradual transitions were detected by generating a cumulative difference measure from consecutive values of the image differences.

Hampapur, Jain, and Weymouth[3] computed what they call chromatic images by dividing the change in gray level of each pixel between two images by the gray level of that pixel in the second image. During dissolves and fades, this chromatic image assumes a reasonably constant value. They also computed a similar image that detects wipes. Unfortunately, this technique is very sensitive to camera and object motion.

Statistical Differences

Statistical methods expand on the idea of pixel differences by breaking the images into regions and comparing statistical measures of the pixels in those regions. For example, asturi and Jain[4] compute a measure based on the mean and standard deviation of the gray levels in regions of the images. This method is reasonably tolerant of noise, but is slow due the complexity of the statistical formulas. It also generates many false positives ~i.e., changes not caused by a shot boundary.

Histograms

Histograms are the most common method used to detect shot boundaries. The simplest histogram method computes gray level or color histograms of the two images. If the bin-wise difference between the two histograms is above a threshold, a shot boundary is assumed. Ueda, Miyatake, and Yoshizawa[5] used the color histogram change rate to find shot boundaries. Nagasaka and Tanaka[6] compared several simple statistics based on gray level and color histograms. They found the best results by breaking the images into 16 regions, using a x2 test on color histograms of those regions, and discarding the eight largest differences to reduce the effects of object motion and noise.

Swanberg, Shu, and Jain[7] used gray level histogram differences in regions, weighted by how likely theregion was to change in the video sequence. This worked well because their test video CNNheadline News had a very regular spatial structure. They did some simple shot categorization by comparing shots with the known types e.g., anchor person shot! in a database. They were also able to group shots into higher level objects such as scenes and segments by matching the shot types with the known temporal structure.

Zhang, Kankanhalli, and Smoliar[1] compared pixel differences, statistical differences, and several different histogram methods and found that the histogram methods were a good trade-off between accuracy and speed. In order to properly detect gradual transitions such as wipes and dissolves, they used two thresholds. If the histogram difference fell between the thresholds, they tentatively marked it as the beginning of a gradual transition sequence, and succeeding frames were compared against the first frame in the sequence. If the running difference exceeded the larger threshold, the sequence was marked as a gradual transition.

To reduce the amount of processing needed, they compared nonadjacent frames and did finer level comparisons if a possible break was detected.

Compression Differences

Little et al.[8] used differences in the size of JPEG compressed frames to detect shot boundaries as a supplement to a manual indexing system. Arman, Hsu, and Chiu[9] found shot boundaries by comparing a small number of connected regions. They used differences in the discrete cosine transform ~DCT! coefficients of JPEG compressed frames as their measure of frame similarity, thus avoiding the need to decompress the frames. A further speedup was obtained by sampling the frames temporally and using a form of binary search to find the actual boundary. Potential boundaries were checked using a color histogram difference method.

Edge Tracking

Zabih, Miller, and Mai[10] compared color histograms, chromatic scaling, and their own algorithm based on edge detection. They aligned consecutive frames to reduce the effects of camera motion and compared the number and position of edges in the edge detected images. The percentage of edges that enter and exit between the two frames was computed. Shot boundaries were detected by looking for large edge change percentages. Dissolves and fades were identified by looking at the relative values of the entering and exiting edge percentages. They determined that their method was more accurate at detecting cuts than histograms and much less sensitive to motion than chromatic scaling.

Motion Vectors

Ueda, Miyatake, and Yoshizawa[5] and Zhang, Kankanhalli, and Smoliar1 used motion vectors determined from block matching to detect whether or not a shot was a zoom or a pan. Shahraray[2] used the motion vectors extracted as part of the region-based pixel difference computation described above to decide if there is a large amount of camera or object motion in a

shot. Because shots with camera motion can be incorrectly classified as gradual transitions, detecting zooms and pans increases the accuracy of a shot boundary detection algorithm. Motion vector information can also be obtained from MPEG compressed video sequences. However, the block matching performed as part of MPEG encoding selects vectors based on compression efficiency and thus often selects inappropriate vectors for image processing purposes.
These algorithms can be classified according to feature they use.

III. Classification of SBD algorithms

These algorithms can be classified in various dimensions they use like features of frame, part of frame (spatial domain) on which of features are calculated, temporal domain – calculations applied on which two consecutive frames is decided here and shot change detection method.

Features Used

Taking specific features from frame and working on it is necessary. Most of all shot change detection algorithms work on the video domain by extracting a small number of features from each video frame. These are extracted either from the whole frame or from a subset of it. This is called a region of interest (ROI). Following are popular features those are used generally.

1. Luminance/colour: The simplest feature that can be used to characterize a frame or ROI is its average gray scale luminance [11]. Which we can say intensity of pixels

2. Luminance/colour histogram: Just to use colour than it is better feature to use gray scale or colour histogram of ROI. It is quite easy to compute and mostly robust against translational, rotational and zooming camera motion, for the above reasons it is widely used [12].

3. Image edges: Other popular choice of feature is edge information in frame. Edges can be used as its numbers or particular for objects to extract ROI statistics. They are invariant to illumination changes and most motion, and they correspond somewhat to the human visual perception. But calculating edges is having higher computational cost and it is noise sensitivity.

4. Transform coefficients: Discrete Fourier Transformation and Discrete Cosine Transform are a classic way to describe the texture of a ROI.

5. Multiple features: Many algorithms extract several types of features and use them in combination or for subsequent processing.

A number of other features are used in the specific methods where they are more relevant.

Spatial Domain

Processing whole frame is quiet lengthy and time consuming as well as need more processing power. So calculating ROI and extracting features for that particular only is current scenario. This ROI can be any part of frame as described below.

1. Rectangular block: Popular method is to segment each frame into equal-sized blocks of size 3x3, 9x9, 25x25 any like this and extract a set of features per block [13].

2. Arbitrarily shaped region: Feature extraction can also be applied to arbitrarily shaped and sized regions. Here we are extracting particular part of frame from where shot change is easily detected and it is reliable also. As our interest is for change in particular part of frame only. Like particular logos or symbols in frames.

3. Full frame: The algorithms that extract features from the full frame [14] is very resistant to motion, but tend to have poor performance at detecting the change between two similar shots as change ratio of feature is very less.

Temporal Domain

Another important aspect of shot boundary detection algorithms is comparison between features calculated. So in which band two consecutive frames are compared is called the temporal window of frames. These can be one of the following:

1. Two frames: This is the simplest way to detect discontinuity. Here high value of the discontinuity metric between two successive frames is getting detected. But this approach fails when there is so much variation in video or short lived moments in video which affects the normal progress.

2. N-frame window: Here we are computing discontinuity of consecutive frames between decided N-frame windows. So one metric is maintained and values of calculated feature is stored and compared in this window. Here dynamic or static values can be used to make window of specific size N.

3. Entire current shot: Here in this method we calculate feature value for all frames in shot. Now for shot, value can be derived by average value or equation applying to frame value. And this value will be compared to next frame to check whether they belong to same shot or not.

4. Entire video: It is possible to take characteristic of whole video into consideration while detecting shot change. But here the major problem is video may have drastic variations inside it, between shots. So deriving a common parameter is not reliable.

Shot Change Detection Method

1. Thresholding: Here ROI's feature calculated value is compared to certain constant threshold. So the problem here is this threshold might be different for different parts of video. So threshold is needed to adjust by hand for various part of video.

2. Adaptive Thresholding: solution to hand change of thresholding is adaptive threshold. Here threshold vary for different temporal domain. Which is machine calculated.

3. Trained classifier or heuristic approach: defining the data with shot change and not shot change, we can train our machine to classify the future data. Heuristic approach can also be applied to domain specific data set.

4. User interaction: Transaction those are hard to find automatically that can also be detected by manual force, but this task is time consuming and tedious.

IV. Observation and conclusion

Colour histogram method is reliable in finding hard cuts. While edge change ratio method can detect shot change of hard cut and fades as well as dissolve also. With contrast feature dissolve is easy to identify and standard deviation is reliable in detecting fades.

References

[1] H. J. Zhang, A. Kankanhalli, and S. W. Smoliar, ''Automatic partitioning of full-motion video,'' Multimedia Systems 1-1 10–28-1993

[2] B. Shahraray, ''Scene change detection and content-based sampling of video sequences,'' in Digital Video Compression: Algorithms and Technologies, Proc. SPIE 2419, 2-13-1995

[3] A. Hampapur, R. Jain, and T. Weymouth, ''Digital video segmentation,'' Proc. ACM Multimedia 94, pp. 357–364, San Francisco, CA-1994

[4] R. Kasturi and R. Jain, ''Dynamic vision,'' in Computer Vision: Principles, R. Kasturi and R. Jain, Eds., IEEE Computer Society Press, Washington -1991

[5] H. Ueda, T. Miyatake, and S. Yoshizawa, ''IMPACT: an interactive natural-motion-picture dedicated multimedia authoring system,'' Proc. CHI. 1991, pp. 343–350 ACM, New York 1991

[6] A. Nagasaka and Y. Tanaka, ''Automatic video indexing and fullvideo search for object appearances,'' in Visual Database Systems II, E. Knuth and L. Wegner, Eds., pp. 113–127, Elsevier Science Publishers 1992

[7] D. Swanberg, C. F. Shu, and R. Jain, ''Knowledge guided parsing and retrieval in video databases,'' in Storage and Retrieval for Image and Video Databases, Proc. SPIE 1908, 173–187 1993

[8] T. D. C. Little, G. Ahanger, R. J. Folz, J. F. Gibbon, F. W. Reeve, D. H. Schelleng, and D. Venkatesh, ''A digital on-demand video service supporting content-based queries,'' Proc. ACM Multimedia 93, pp. 427–436, Anaheim, CA ~1993

[9] F. Arman, A. Hsu, and M-Y. Chiu, ''Image processing on encoded video sequences,'' Multimedia Systems 15 211–219 ~1994

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

[11] B. Shahraray, ''Scene change detection and content-based sampling of video sequences,'' in Digital Video Compression: Algorithms and Technologies, Proc. SPIE 2419, 2–13 1995

[12] P. Campisi, A. Neri, and L. Sorgi " Automatic dissolve and fade detection for video sequences" In Proc. Int. conf. on Digital Signal Processing, July 2002.

[13] Z. Cernekova, C. Kotropoulos, and I. Pitas " Video shot segmentation using singular value decomposition". In Proc. 2003 IEEE Int. Conf. on Multimedia and Expo,volume II, pages 301 – 302, Baltimore, Maryland, USA, July 2003.
[14] .Alan Hanjalicb"Shot-boundary detection: Unraveled and resolved?" IEEE Trans. on Circuits and Systems for Video Technology, 12(2):90 – 105, February 2002