A New Shot Boundary Detection Algorithm

Dong Zhang, Wei Qi, Hong Jiang Zhang

Microsoft Research, China
hjzhang@microsoft.com

Abstract. Shot is often used as basic unit for both analyzing and indexing video. In this paper, we present an algorithm for automatic shot detection. In our algorithm, we use a 'flash model' and a 'cut model' to deal with the false detection due to flashing lights. A technique for determining the threshold that uses the local window based method combined with reliability verify process is also developed. The experimental results of our method are very promising, improving the performance of shot detection.

1 Introduction

There are many shot detection methods already proposed in past decades [1-6]. The common way for shot detection is to evaluate the difference between consecutive frames represented by a given feature. Although reasonable accuracy can be achieved, there are still problems that limit the robustness of these algorithms.

One of the common problems in robust shot detection results from the fact that there are many flashlights in news video, which often introduce false detection of shot boundaries. Only some simple solutions to this problem have been proposed in [2]. Their main limitations are that they assume the flashlight just occur during one frame. In real world, such as news video, there will be many flashlights occur during a period of time and influence multiple consecutive frames.

Another problem that has not been solved very effectively well is threshold selection when comparing changes between two frames. Most of the existing methods use global pre-defined thresholds, or simple local window based adaptive threshold. Global threshold is definitely not efficient since the video property could change dramatically when content changes, and it is often impossible to find a universal optimal threshold across any video segments. The local window based adaptive threshold selection method also has its limitation because in some situation the local statistic are 'polluted' by strong noises such as big motions and flashlights

To address these two practical problems, we developed two new techniques: a robust flashlight detection technique and a new adaptive threshold selection technique. In the flashlight detection technique, we distinguish flashlights from real shot cut by applying a 'cut model' and a 'flashlight model' which are defined based on the temporal property of average intensity value across a frame sequence in a local window. For threshold detection, we proposed a strategy on whether to recalculate the threshold from the current local window or use the existing threshold based on the local shot activity statistic. In this way, our shot detection algorithm incorporates certain safeguards that ensure that thresholds are set in accordance with the received video content only if the content is relatively stable.

The rest of this paper is organized as follows. Section 2 gives a detail description of the two new algorithms. Section 3 presents experimental results, and we conclude this paper and discuss the future work in Section 4.

2 Proposed Algorithm

Our new algorithm is based on the framework of the twin-threshold method proposed by Zhang [3] that is able to detect both abrupt and gradual transition, such as dissolve and wipe, in video. We integrate our two methods for flashlight detection and threshold selection into this framework. A brief diagram of the framework is shown in Fig. 1. Our proposed two techniques are integrated into to the true cut decision and threshold selection functional blocks.

2.1 Metrics in Shot Detection

To partition the video, we should first define suitable metrics, so that a shot boundary is declared whenever that metric exceeds a given threshold. We use histogram difference as the first metric in our algorithm because histogram is less sensitive to object motion than other metrics, shown in equation (1). Another metric adopted in our algorithm is average intensity difference, shown in equation (4). We define the two metrics as bellow:

$$D_i = \sum_{j=1}^{Bins} | H_i(j) - H_{i-1}(j) | \tag{1}$$

$$AI_i = \frac{\sum_{j=1}^{Bins} j * H_i(j)}{\sum_{j=1}^{Bins} H_i(j)} \tag{2}$$

$$AI_{i-1} = \frac{\sum_{j=1}^{Bins} j * H_{i-1}(j)}{\sum_{j=1}^{Bins} H_{i-1}(j)} \tag{3}$$

$$AID_i = AI_i - AI_{i-1} \tag{4}$$

Where $H_i(j)$ indicates j-th bin of the gray-value histogram belonging to frame i. Generally, we choose 256 bins for gray level histogram. D_i denotes the histogram difference between frame i and its preceding frame (i-1). AI_i is the average intensity value of the frame i, and AID_i is the average intensity difference between frame i and (i-1).

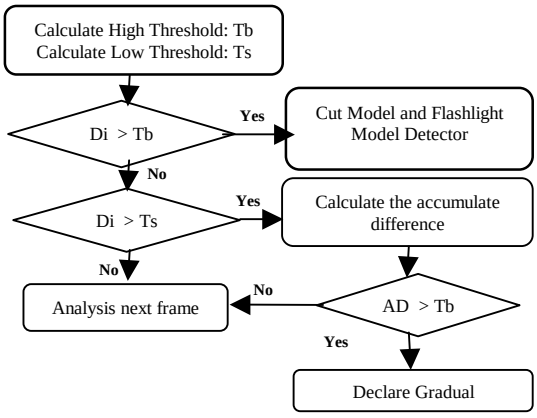

Fig. 1. Overview of the algorithm

2.2 Cut Model and Flash Model

Both the real shot cut and flashlight could cause a great change in histogram difference. So by only using the histogram difference to detect shot boundaries may cause a lot of false alarms because of flashlights. If the histogram difference of consecutive frames is larger than a particular threshold, shot boundary detector invokes an instance of flashlight detector to distinguish a potential flashlight condition from an actual shot cut boundary. In flashlight detector, we define two models based on the temporal property of flashlight and cut. If the histogram difference of adjacent frames is larger than an adaptive higher threshold: T_b, we can use either the histogram difference or average intensity difference as input of our models that are illustrated in Figure 2.

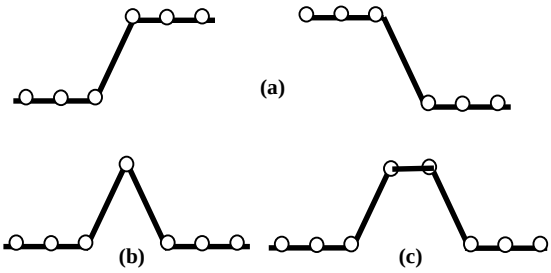

Fig. 2. "Flash Model" and "Cut Model"

Next, we use average intensity difference to describe our ideas. The ideal models for shot cut and flashlight based on average intensity are shown respectively in Fig. 2(a) and Fig. 2(b). Where the lines indicate the average intensity curve of a frame sequence. When a true abrupt transition occurs, the average intensity changes from one level to another, then the change will sustain in the next whole shot. When a flashlight occurs, the average intensity changes from one level to another, and then fall back to the original level after one frame. This is the ideal model for flashlight. In real world, the flashlight could last several frames and multiple flashlights could occur consequently, which could be represented by the Fig.2 (c). Then the problem of distinguishing true abrupt shot transition from flashlight is converted to the problem of designing an algorithm to distinguish the above two types of models. First, we define two 'heights' that will be used to classify these two models.

H1: the average intensity difference between current frame and previous frame

H2: the average intensity difference between frames in left sliding windows preceding the current frame and frames in right sliding windows after the current frame.

In ideal flashlight model, *H2* is zero because the average intensity of frames preceding the current frame and the average intensity of frames after the current frames are at the same level within the same shot. Meanwhile, in ideal cut model, *H2* is identical to *H1* because the average intensity of frames preceding the current frame is not at the same level with that of frames after the current frames. Thus, we use the ratio of H_1 and H_2 as the metric to distinguish these two models. We define the ratio as:

$$Ratio = \frac{H2}{H1}$$

Then we have the following rule:

$$Ratio = \begin{cases} 1 & \text{Cut Model} \\ 0 & \text{Flash Model} \end{cases}$$

As Ratio goes to a value of 1, flashlight detector concludes that the intensity change is due to a shot cut event and is, therefore indicative of a shot boundary. Deviations from a Ratio value of 1 are determined to be indicative of a flashlight event. Actually, we use a threshold T to make the decision. Threshold T should be larger than zero and less than one, we set it 0.5 in our experiment, and then the above rule is converted to:

$$Ratio = \begin{cases} > T & \text{Cut Model} \\ < T & \text{Flash Model} \end{cases}$$

When we calculate *Ratio* above, H_1 is easy to calculate, but H_2 is hard to define because of the complexity of practical video data as shown in Fig. 4. We use the following method to calculate H_2. First, we put the average intensities of 5-7 (the size of left sliding window) frames preceding current frame into left sliding window, at the same time put the average intensities of 5-7 (the size of right sliding window) frames after current frame into right sliding window. The reason that we choose 5-7 frames is based on the fact flashlight events do not, generally, last longer than 5-7 frames. Then, we use the average intensity of all frames within right and left sliding window to calculate *H2*. The advantages of using average intensity of all frames within right and left sliding windows as representative frames lie in two facts: One is that the flashlights always make the average intensity of frame larger. The other reason is that there are often several frames whose change is unstable within right or left sliding windows. The flow char is shown in Fig. 3.

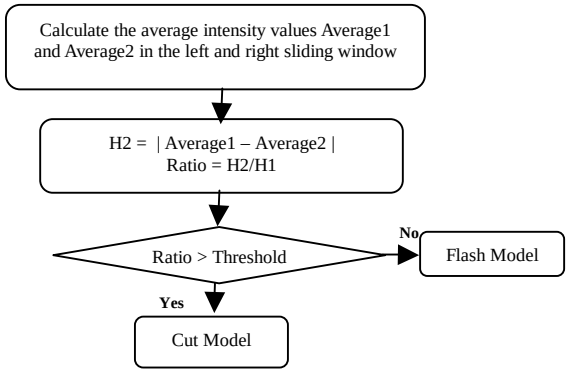

Fig. 3. Flow char of the Flashlight detector

According to one example implementation, this algorithm is applied to sequence 'CNN news', a segment with a lot of flashlight. The average intensity of frames from 277 to 700 is shown in Fig. 4. From left to right, there are a true cut and six occurrences of flashlight. We can observe that: in the average intensity diagram, a single or multiple pulses usually represent a flashlight. However, when a true cut occurs, the average intensity of frame is changed from one level to another.

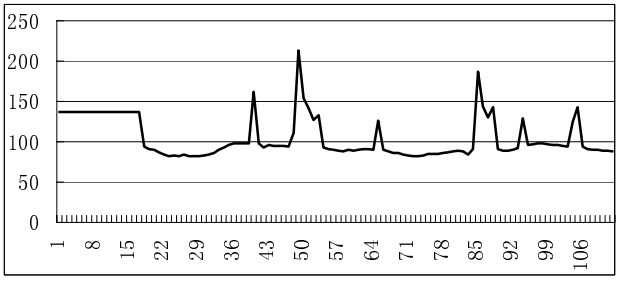

Fig. 4. **Average intensity diagram**

We can also use histogram difference in our 'flash model' and 'cut model' detector, where average intensity of all frames within sliding windows is replaced by the

average histogram of all frames within sliding windows. The average intensity difference is replaced by histogram difference.

2.3 Threshold Selection

The problem of choosing the appropriate threshold is a key issue in applying the shot detection algorithms. Heuristically chosen global thresholds is inappropriate because experiments have shown that the threshold for determining a segment boundary varies from one shot to another which must be based on the distribution of the frame-to-frame differences of shots. So adaptive threshold is more reasonable than global threshold. In our algorithm, we proposed the local window based threshold calculation method combined with reliability verify process. We build a sliding window preceding the current frame. Then we calculate the mean value: μ and the variance of histogram differences: σ within the sliding window. Next the average value is used to calculate the low threshold Ts (2 to 3 times of the mean value) and high threshold *Tb* (3 to 5 times of mean value) used in twin-threshold algorithm. Analyzing the variance within the sliding window can generate a statistical model of the light intensity of the frames within the window. So variance is used to analyze the reliability of calculated thresholds. If the variance is larger than a threshold T2, this is an indicator to threshold selection that the data is not a good source from which to develop thresholds. That is to say the distribution of histogram differences within the sliding window is dispersed. It means the video contents within the sliding window change dramatically, which is not a good source to develop thresholds. So threshold calculated using the sliding window is not reliable, we abandon the threshold and use previous threshold calculated using the previous sliding window. Vice verse, if the variance is smaller than the threshold *T2*, that is to say the distribution of histogram differences within the sliding window is concentrated. So threshold calculated using the sliding window is reliable, we adopt the threshold to judge whether current frame is shot boundary or not. Fig. 5 shows the illustration of the sliding window threshold selection.

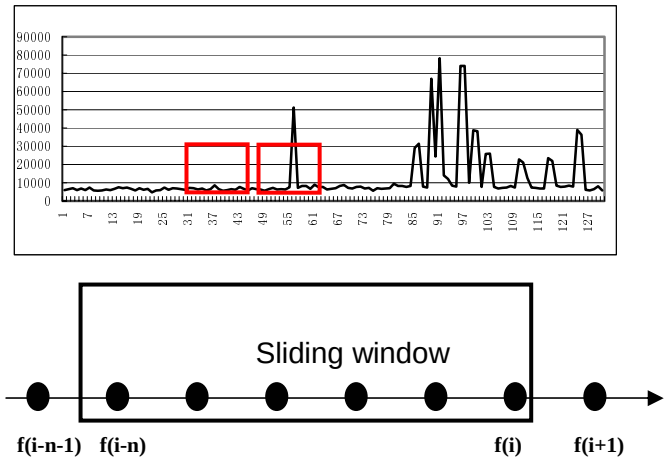

Fig. 5. Histogram difference diagram and illustration of sliding window

As shown in Fig. 5, the diagram is the histogram difference of consecutive frames in sequence 'CNN news'. In the left sliding window, there is little variation of histogram difference, we regard the thresholds calculated using the window is reliable; while in the right sliding window, the histogram difference has large variance due to camera motion and flashlight, and then we abandon the thresholds calculated using the

window and adopt pre-determined thresholds for use until the frames within the sliding window do not result in such a high variance.

3 Experimental Results

The algorithms described in this paper have been tested on a number of video sequences, especially news video in which a lot of flash occurs. The parameters used in our algorithm are fixed to all test video. We summarize all parameters in the following table.

Table 1. Parameters in experiment

Parameters	Size	Tb	Ts	T1	T2
Value	15	3.5	2.0	0.5	500

In table1, 'Size' is the frame number within the sliding window. In threshold selection, we put 15 frames preceding current frame into sliding window; and in flashlight detector we put 7 frames preceding current frame into left sliding window and 7 frames after current frame into right sliding window. It will be noticed that sliding windows of more or less frames could well be used, the size of 5-7 frames is chosen because flashlight events do not, generally, last longer than 5-7 frames. *Tb* and *Ts* are the high and low thresholds used in twin-threshold algorithm. *T1* is the threshold in flashlight detector. *T2* is the threshold used in adaptive thresholds selection. In table 2, we present the results of shot detection of 5-min "CNN" news and two10-min "SPORTS" news and a 2-min MPEG-7 test sequence "CUT15". They are representative material for testing our algorithm.

Table 2. Experiment Results. D: correct detections; M: missed detections; F: false alarms

Tested Video	Cut			Flash			Gradual Tran.		
	D	M	F	D	M	F	D	M	F
CNN	32	3	4	45	4	2	5	1	1
SPORT1	30	6	3	36	3	4	8	2	3
SPORT2	70	9	5	0	0	0	7	2	2
CUT15	4	0	1	0	0	0	16	2	2

Table 2 indicates that our algorithm can detect not only camera transition but also flashing lights with satisfactory accuracy. Approximately 87% of camera transitions and flashlight is detected. The miss camera transition mainly results from the fact that the differences between consecutive frames across a camera transition are lower than the given threshold because the histograms of the two frames are similar. Object movement is the source of false detection of camera transition, especially for gradual transitions. Flashlight detection fails are due to two main reasons. One reason is that some flash may be too weak to detect. The other reason is that the flash may sustain too many frames exceeding the size of the sliding window, so the detector mistake 'Flash Model' as 'Cut Model'.

4 CONCLUSION AND FUTURE WORK

This paper has presented an effective shot detection algorithm, which focus on two difficult problems solutions: false detection of shot transition due to flashing light and threshold selection. The main contributions of the presented work are to build two models and an adaptive threshold selection algorithm that uses the local window

combined with reliability verify process. Experiments show that the proposed algorithm is promising.

However the automatic video partition is still a very challenging research problem especially for detecting gradual transitions. Further work is still needed.

5 REFERENCES

[1] A. Nagasaka and Y.Tanaka, Automatic Video Indexing and Full-Video Search for Object Appearances, Visual Database Systems, II, pp.113-127, Elsevier Science Publishers, 1992.

[2] F. Arman, A. Hsu, and M.Y. Chiu, Image processing on compressed data for large video databases, Proceedings of 1st ACM International Conference on Multimedia, Anaheim, CA, pp.267-272, 1993.

[3] H. J. Zhang, A. Kankanhalli, S. W. Smoliar, Automatic Partitioning of Full-motion Video, ACM Multimedia System, Vol. 1, No.1, pp. 10-28, 1993.

[4] Hampapur, A., Jain, R., and Weymouth, T., Digital Video Segmentation, Proc. ACM Multimedia 94, San Francisco, CA, October, 1994, pp.357-364.

[5] B.L. Yeo and B. Liu, Rapid scene analysis on compressed video, IEEE Transactions on Circuits and Systems For Video Technology, 5,6, pp.533-544, 1995.

[6] Ishwar K. Sethi, Nilesh Patel. A Statistical Approach to Scene Change Detection. SPIE vol.2420, 329-338. San Jose, California: 1995.