Fast Scene Change Detection Algorithm

*Khvan, Dmitriy **Ng, Teck Sheng ***Jeong, Jechang
Hanyang University
*dm_kvn@hotmail.com

Abstract

In this paper, we propose a new fast algorithm for effective scene change detection. The proposed algorithm exploits Otsu threshold matching technique, which was proposed earlier. In this method, the current and the reference frames are divided into square blocks of particular size. After doing so, the pixel histogram of each block is generated. According to Otsu method, every histogram distribution is assumed to be bimodal, i.e. pixel distribution can be divided into two groups, based on within-group variance value. The pixel value that minimizes the within-group variance is said to be Otsu threshold. After Otsu threshold is found, the same procedure is performed at the reference frame. If the difference between Otsu threshold of a block in the current frame and co-located block in the reference frame is larger than predefined threshold, then a scene change between those two blocks is detected.

1. Introduction

The scene changes in the video sequences mainly include sudden scene change, cross fade, fade in and fade out, the characteristics of cross fade, fade in and fade out is a gradual scene change process between two scenes. The sudden scene change switches from scene one to scene two immediately, the cross fade changes gradually from scene 1 until scene 2, fade in changes gradually from a monochromatic frame until another scene. Scene change detection is a key technique of video coding, it has an important impact on the video quality. Scene detection is essential to automatic content-based video segmentation. A video shot is a collection of video frames obtained through a continuous camera recording. Similar background and motion patterns typify the set of frames within a shot. Video data can be divided into different shots. A shot is a video sequence that consists of continuous video frames for one action. Scene change detection is a fundamental operation that divides video data into shots [1].

Nowadays, with development of signal processing a video scene change detection method is necessary. Many video applications demand the scene change detection to organize the video content .Video indexing techniques, which are necessary for video databases, rely on scene change detection.

There are many scene change detection algorithm recently proposed. These algorithms exploit different criteria for scene change detection. For instance, several methods use the low-level global features such as the luminance pixel-wise difference, another exploit luminance or colour histogram difference to compare two consecutive frames. Several authors use fixed thresholds for the similarity comparison between the current and the reference frames. These types of schemes define a threshold, typically a fixed one. If the value of the measure exceeds the threshold, a scene change is detected. However, a fixed threshold value cannot perform well for all videos mainly due to the diversity of their characteristics. The key problem is to obtain an optimal value for such fixed threshold.

So far, most of the developed video segmentation algorithms merely exploit visual information. However, several works such as [2] suggest that audio content of a sequence can be also used for scene change detection. A scene change is most often accompanied by a significant change in the audio characteristics. Despite the fact that audio content can be utilized as supplementary information, this work focuses on scene change detection analyzing only visual information.

In addition to different criteria for scene change detection, an algorithm can be either block-based or pixel based. Pixel based methods often produce a significant number of false detection and missed detections when high motion and brightness variations are present in the video. Mainly for this reason, in the proposed algorithm the block based approach was selected.

This paper is organized as follows. In section 2, the conventional method will be described in details. Section 3 will present the proposed algorithm. In section 4 experimental results will be shown. Finally, section 5 will conclude the paper.

2. Conventional method

The algorithm proposed in [3] was chosen as basis for the proposed one. This can be explained by effectiveness of Otsu thresholding approach. Moreover, this method is relatively simple and therefore fast. In this section we are going to describe Otsu method in details. Otsu method assumes that the distribution is bimodal and divides a distribution into two groups. For the two groups, it finds a boundary, i.e. a threshold that minimizes th within group variances. The variables for Otsu method are shown below.

$$
\begin{aligned}
q_1(t) &= \sum_{i=1}^{t} p(i) \\
q_2(t) &= \sum_{i=t+1}^{M} p(i) \\
u_1(t) &= \sum_{i=1}^{t} \frac{ip(i)}{q_1(t)} \\
u_2(t) &= \sum_{i=t+1}^{M} \frac{ip(i)}{q_2(t)} \\
\sigma_1^2(t) &= \sum_{i=1}^{t} \frac{[i-u_1(t)]^2\, p(i)}{q_1(t)} \\
\sigma_2^2(t) &= \sum_{i=t+1}^{M} \frac{[i-u_2(t)]^2\, p(i)}{q2(t)}
\end{aligned} \tag{1}
$$

where $p(i)$ denotes the histogram probability at the threshold level i. $q_1(t)$ is the probability for the group with values less than or equal to the threshold t and $q_2(t)$ is the probability for the group with values greater than t. $u_1(t)$ and $u_2(t)$ are the means for the first and second groups, respectively. $\sigma_1^2(t)$ and $\sigma_2^2(t)$ are the variances for the first and second groups, respectively. Using Otsu method, we can calculate the within-group variance, $\sigma_w^2(t)$, as shown in (2). In the automatic thresholding, we find the threshold t that minimizes $\sigma_w^2(t)$

$$\sigma_w^2(t) = q_1(t)\sigma_1^2(t) + q_2(t)\sigma_2^2(t) \tag{2}$$

Figure 1 shows an example of applying the automatic thresholding to two block-level histograms from the previous and current frames with a scene change. The figure shows the threshold levels for the two histograms. As shown in the figure, the two thresholds have significantly different values, which indicates that there is a scene change between the two corresponding blocks.

Utilizing aforementioned scheme, it is possible to detect local (part of a frame) scene change as well as global scene change. The ability to detect the local scene change yields significant improvement of Otsu algorithm's performance. As figure 2 demonstrates, the area enclosed in red box is the locally changed region.

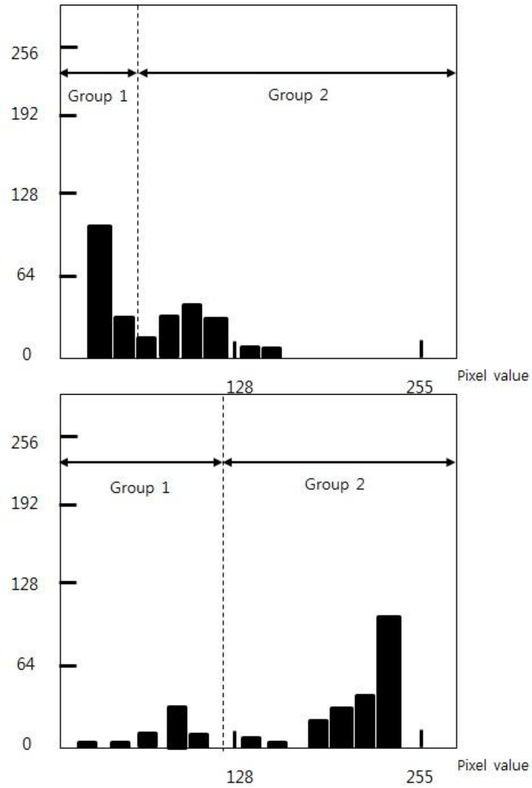

Figure 1. Bimmodal pixel histogram distribution.

Figure 2. Local scene change.

Figure 3. Global scene change.

3. Proposed method

The motivation for the proposed algorithms is to make the scene change detection method described above to operate with greater speed. The scene change detection is often used as an element of another algorithm, e.g. many well-known methods perform the scene change detection prior to interpolation. For this reason, fast implementation of the scene change detection algorithm explained earlier can considerably improve an algorithm's overall performance.

The proposed scheme is simple, but yet robust. Our goal was to achieve fast implementation without sacrificing the robustness of the conventional algorithm. The developed method can be described in three steps.

Step 1. First, we propose to check four blocks in the center of a frame on existence of scene change. The size of the block is chose to be 32x32, which contains 256 pixels. The reason to do so is to increase the area of a frame that will be checked for existence of scene change, while maintaining the calculating complexity as if it was 16x16 block.

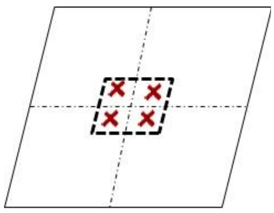

Figure 4. Step 1.

Step 2. After completing the step 1, those frames where the scene change was detected are further checked for existing of scene change in the way shown in the figure below. The larger emphasis is given to the bottom area of a frame because this is where subtitles usually occur.

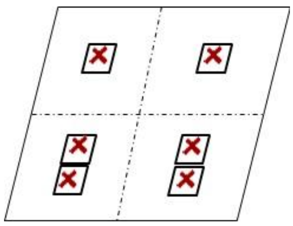

Figure 5. Step 2.

Step 3. Lastly, the final decision about whether the scene change exist or not is made according to results obtained in steps 1 and 2.

The proposed algorithm has detected the same number of scene changed as in Otsu algorithm, what gives us the right to claim its robustness.

4. Experimental results

Table 1. CPU time results.

Friends.yuv	Algorithm	#of scene changes	CPU time(sec)
	Otsu	4	0.89
	Proposed	4	0.56

Bigbang.yuv	Algorithm	#of scene changes	CPU time(sec)
	Otsu	4	0.90
	Proposed	4	0.71

Wilfred.yuv	Algorithm	#of scene changes	CPU time(sec)
	Otsu	3	1.16
	Proposed	3	0.68

Familyguy.yuv	Algorithm	#of scene changes	CPU time(sec)
	Otsu	5	0.99
	Proposed	5	0.66

5. Conclusion

The proposed scheme has considerably improved the performance of the previous work as it can be seen from the experimental results. It can be said that we have achieved the goal to maintain the robustness of the earlier method but reduce computation time. The proposed method can be possibly used as an element of more complex algorithm. The logical flow of the proposed method is very intuitive and therefore easy to comprehend.

Acknowledgement

This work was supported by the Industrial Strategic technology development program(10039886, Combined Modulator and Demodulator with Real Time Multi Codec Transcoder based on Hardware Processing) funded by the Ministry of Knowledge Economy(MKE, Korea).

References

[1] Nagasaka, A. and Tanaka, Y., "Automatic Bideo Indexing and Full-Video Search for Object Appearances," *Proc.of IFIP TC2/WG 2.6 Second Working Conference on Visual Database Systems II* (Amsterdam, The Netherlands, November 02-05, 1995)

[2] Kyperountas, M., Kotropolous, C. and Pitas, I., "Enhanced Eigen-Audioframes for Audiovisual Scene Change Detection," *IEEE Trans. on Multimedia*, vol. 9, no. 4, pp.785 - 797, June

2007.

[3] Kang, S.J., Cho, S.I. and Yoo, S.J., "Scene Change Detection Using Multiple Histograms for Motion-Compensated Frame Rate Up-Conversion," *IEEE Trans. on Consumer Electronics,* vol. 53, no. 3, pp.1117 – 1124, Aug. 2012.

[4] Shapiro, L.G. and Stockman, G.C. "Computer Vision," *Upper Saddle River, NJ:Prentice Hall.*.

[5] L. Alparone, M.Barni, F. Bartolini, V. Cappelini, "Adaptively weighted vector-median filters for motion-fields smoothing," *IEEE Trans. on Consumer Electronics,* vol. 4, pp. 2267-2270, May 1996.

[6] You-Young Jung, Byung-Tae Choi, Yung-Jun Park and Sung-Jea Ko, "An Effective De-interlacing Technique Using Motion Compensated Interpolation," *IEEE Transactions on Consumer Electronics,* vol. 46, no. 3, pp. 460-466, Aug. 2000,

[7] K. Sugiyama and H. Nakamura, "A method of de-interlacing with motion compensated interpolation," *IEEE Trans. Consum. Electron.*, vol. 45, no. 3, pp. 611 - 616, August 1999.

[8]Y.Y. Jung, B.T. Choi, Yung-Jun Park and Sung-Jea Ko, "An effective de-interlacing technique using motion compensated interpolation," *IEEE Trans. on Consumer Electronics,* vol. 46, no. 3, pp. 460-466, Aug. 2000.