Scene Cut Detection Using The Colored Pattern Appearance Model*

Kin-Wai Sze[1], Kin-Man Lam[1] and Guoping Qiu[2]

[1] Centre for Multimedia Signal Processing, Department of Electronic and Information Engineering, The Hong Kong Polytechnic University, Hong Kong
[2] School of Computer Science, The University of Nottingham, UK

ABSTRACT

In this paper, we propose to use the Colored Pattern Appearance Model (CPAM) as a content representation for video scene break detection. This model represents a scene by means of global statistics of the local visual appearance, and was originally motivated by studies in human color vision. The performance of this method is compared to several histogram-based approaches. An adaptive thresholding technique, namely entropic thresholding, is applied to determine the respective optimal threshold values for each of the approaches. In the experiments, the two video sequences in the MPEG-7 content set are used to evaluate the performances of the CPAM and the histogram-based methods. Experimental results show that our proposed model outperforms other histogram-based approaches in scene break detection.

1. INTRODUCTION

Content-based video indexing (CBVI) has been an extensively researched area in the computer vision community for the past decade, and many approaches have been developed and published. A general approach to CBVI [2, 3, 6] is to temporally segment a video into shots based on the extracted low-level visual features, and to use the visual features for indexing and providing a high-level understanding of the video. Low-level representation of multimedia signals has been commonly used in segmentation, indexing, retrieval, etc.

Temporal video segmentation plays a very important role in content-based video indexing. This process provides a fundamental understanding of indexing a video efficiently. Basically, it involves the detection of both abrupt and gradual transitions. In general, the detection of these kinds of transition involves two procedures: content representation and decision-making. In the content representation process, a video is represented by low-level features, such as DC image [7], edge image [8], monochrome histogram, color histogram in different color spaces, etc., which can be analyzed efficiently. In the decision-making process, thresholding technique is usually used to detect and identify the transitions. There are two ways to set the threshold: one is to pre-set it by experiments, the other is to set it adaptively. In this paper, we will evaluate the performances of different kinds of histogram-based low-level representation for automatic detection of abrupt transitions, so that the most effective and reliable one can be identified. There are many possible low-level visual features for representing video contents. With a particular representation, there may also be many ways to determine an abrupt transition. The entropic thresholding technique [4] is chosen in our analysis; it can provide an optimal and automatic solution in determining the threshold for detecting scene breaks.

To evaluate the performances of the different visual features with the entropic thresholding technique in detecting scene breaks, the two video sequences with more than 30,000 frames each from the MPEG-7 content-set are used. The frame numbers of the scene cuts in the videos have been marked manually. The CPAM approach [1] and a number of histogram-based low-level representations including monochrome histogram and color histograms in different color spaces are evaluated and compared. Experimental results show that the joint achromatic spatial pattern and chromatic spatial pattern based on CPAM achieves the best performance. The organization of the paper is as follows. Sections 2 and 3 present the CPAM representation and several low-level content representations, respectively. Section 4 describes the entropic thresholding technique used in our experiment. Section 5 presents the experimental results, and the conclusion is drawn in Section 6.

2. COLORED PATTERN APPEARANCE MODEL (CPAM)

The main problem with video segmentation is the existence of strong noises and motion in the video data. Therefore, it is not easy to have a single representation that is efficient, reliable and robust for scene cut detection. The colored pattern appearance model (CPAM) which has two channels capturing the characteristics of the chromatic and achromatic spatial patterns of small image regions has been used to compile content descriptors for content-based still image retrieval [1]. In this method, the visual appearance of a small image block is modeled by three components: the stimulus strength,

* This research work was supported by internal research grant from The Hong Kong Polytechnic University, Hong Kong.

the spatial pattern and the color pattern. In [1], the YCbCr space is used, and the stimulus strength S is approximated by the local mean of the Y component. The pixels in Y normalized by S form the achromatic spatial pattern (ASP) vector. Because C_b and C_r have lower bandwidth, they are sub-sampled. The sub-sampled pixels of C_b and C_r are normalized by S, and then concatenated together to form the chromatic spatial pattern (CSP) vector. In order to use the representation scheme in content-based temporal video segmentation, vector quantization (VQ) is used to estimate statistically the most representative feature vectors in the feature space. A 256-codeword quantizer for the ASP vectors and a 256-codeword quantizer for the CSP vectors are generated by means of the frequency sensitive competitive learning (FSCL) algorithm [1]. Therefore, each frame can be represented by a 256-bin ASP histogram and/or a 256-bin CSP histogram. The training samples are based on the MIT Media Labs VisTex image database, which consists of images of different texture appearance. The codeword generated based on this database can provide a general representation for different kinds of images/videos.

3. LOW-LEVEL FEATURE VIDEO CONTENT REPRESENTATION

An abrupt transition is usually induced by a camera break in a video. This change can be detected by computing the differences between the visual features of the consecutive frames. Many kinds of features for representing an image/frame have been proposed. Monochrome histograms and color histograms with different color spaces are the most commonly used methods for image representation. In MPEG-7, the color descriptors use different color spaces [5], such as monochrome, RGB, HSV, YCrCb, and HMMD. The opponent color space can also represent an image well for image indexing, and its transformation from the RGB color space is simple.

3.1 RGB Color Histogram

The RGB color space is the most common representation of color information. To generate a color histogram in RGB color space, the R, G and B components of each pixel in a frame are quantized into 256 color indices by vector quantization This 256-bin color histogram $H_{R,G,B}(k)$ is then formed as follows:

$$H_{RGB}(k) = \sum_{i,j} d\left(Q_{R,G,B}(R_{i,j}, G_{i,j}, B_{i,j}) - k\right), \quad (1)$$

for $0 \le k \le 255$

$$d(i-j) = \begin{cases} 1 & \text{for } i = j \\ 0 & \text{for } i \ne j \end{cases}$$

where i, j are the co-ordinates of a pixel, and $Q_{R,G,B}(\ )$ represents the color quantization function which quantizes a color to one of the 256 color indices.

3.2 HSV Color Histogram

The HSV color space can represent color information in the form most similar to human perception. RGB to HSV transformation [5] is a nonlinear but reversible process. In order to quantize the HSV space into 256-bins, fixed quantization is used. The H, S and V values are coded into 4 bits, 2 bits and 2 bits, respectively. The 256-bin HSV color histogram $H_{H,S,V}(k)$ is then formed as follows:

$$H_{HSV}(k) = \sum_{i,j} d\left(L_{H,S,V}(H_{i,j}, S_{i,j}, V_{i,j}) - k\right), \quad (2)$$

for $0 \le k \le 255$

where $L_{H,S,V}(\ )$ represents the color index of a pixel in the HSV color space.

3.3 HMMD Color Histogram

The HMMD color space is a new color space supported in MPEG-7 [5]. Three components, Diff, Sum and Hue, are used to describe a color in the HMMD space. The Hue (H) has the same meaning as in the HSV space. The Diff (D) and Sum (S) components are defined as the difference between max and min and the average of max and min, respectively, where max and min are the maximum and minimum among the R, G and B values. The H, D and S values are quantized into 4 bits, 2 bits and 2 bits, respectively, by a fixed quantization scheme according to MPEG-7 standard.

3.4 Opponent Color Histogram

The opponent color space [9] is a brightness-independent chromaticities space. This color space has the advantage of reducing the histogram dimensionality from 3-D to 2-D. The transformation of RGB to RgBy is simple. The opponent chromaticities are defined in terms of the r, g and b chromaticities:

$$(Rg, By) = \left(r - g, \frac{r+g}{2} - b\right) \quad (3)$$

$$\text{where} \quad r = \frac{R}{R+G+B}, \quad g = \frac{G}{R+G+B}, \quad b = \frac{B}{R+G+B}$$

A 2-D color histogram is then formed with 32 bins per color axis.

After extracting each of the low-level representations, the differences between successive frames of a video will be computed. The histograms f of successive frame differences can be formed as follows:

$$dH_i = \sum_{j=0}^{G-1} \left| H_{i-1}(j) - H_i(j) \right|, \quad \text{for } i = 1,2,...,L\text{-}1 \tag{4}$$

$$W = \max_{i=1.2,...,L-1} \{dH_i\} \tag{5}$$

$$f_k = \sum_{i=1}^{L-1} d\,(dH_i - k), \quad \text{for } 0 \le k \le W \tag{6}$$

where G is the total number of color levels in an image, L is the total number of frames in the video sequence, and dH represents the histogram difference between successive frames.

4. ENTROPIC THRESHOLDING

Thresholding technique is commonly used in segmentation and classification. With a selected threshold, a scene break is declared if the histogram difference is larger than this threshold; otherwise, no scene break occurs. The main problem here is how to determine the optimal threshold for different situations. Basically, there are two forms of the methods that can set the threshold. One is to preset the threshold by experimental results; the other is to set the threshold automatically based on the input data (video) itself. In temporal video segmentation, it is difficult to pre-set a fixed threshold because different directors may have different styles and the videos may have different natures. Adaptive thresholding plays an important role in determination of the threshold under different situations. One of the optimal approaches is called entopic thresholding, which finds the optimal threshold by applying information theory. The entropic thresholding method has been extended to find the optimal threshold for spatial and temporal video segmentation. Two entropies are obtained from two separate probability distributions: one is for the in-class; the other is for the non in-class. The threshold used for segmentation is selected in such a way that the total entropy is maximized. In our experiments, entropic thresholding was used to determine the optimal thresholds for the different low-level video representations. The threshold used for scene cut detection is calculated as follows:

$$P_{ns}(i) = \frac{f_i}{\sum_{k=0}^{T} f_k}, \quad 0 \le i \le T, \tag{7}$$

$$P_s(i) = \frac{f_i}{\sum_{k=T+1}^{W} f_k}, \quad T+1 \le i \le W. \tag{8}$$

$P_{ns}(i)$ and $P_s(i)$ represent the probability for the frames with the non-scene cut relationship with their successive frames and the probability for the frames with the scene cut relationship, respectively. The corresponding entropies for these two classes are:

$$E_{ns}(T) = -\sum_{i=0}^{T} P_{ns}(i) \log P_{ns}(i), \tag{9}$$

$$E_s(T) = -\sum_{i=T+1}^{W} P_s(i) \log P_s(i). \tag{10}$$

E_n $_s(T)$ and $E_s(T)$ represent the entropies for these two classes regions separated by a threshold T. The optimal threshold T_{opt} is chosen to satisfy the following criterion:

$$E(T_{opt}) = \max_{T=0,1,2,...,W} \left\{ E_{ns}(T) + E_s(T) \right\} \tag{11}$$

5. EXPERIMENTAL RESULTS

Many scene cut detection schemes have been proposed and evaluated using different video sequences. This makes it difficult to compare the performances of the different detection schemes. In our experiments, the two home video sequences, namely Lgerca_1.mpg and Lgerca_2.mpg, from the MPEG-7 content set were used. These two video sequences have strong noises and motion. Each of the sequences consists of 42 scene cuts, which have been marked manually by the Requirements Group of the MPEG-7 standard committee.

The objective of our experiment is to seek the best representative low-level feature for automatic temporal video segmentation. The histograms of monochrome, RGB, HSV, YCrCb, HMMD, RgBy, and CPAM were extracted from the videos, and their successive frame differences were then computed. For CPAM, the histograms based on the ASP, CSP, and joint ASP & CSP were considered. The entropic thresholding technique is then applied to each of the approaches such that their optimal thresholds for a video based on the frame difference values are selected. Correct Detection (CD), False Positives (FP), False Negatives (FN), Recalls and Precisions of each of the representations were then measured. The results for the two video sequences are shown in Tables 1 and 2.

Recall and Precision will be the basis of our analysis. The ranges of recall and precision are both between 0 and 1. When recall is equal to 1, no missing occurs. A higher value of precision represents a lower false alarm rate. For temporal video segmentation, it is difficult to have an algorithm that can provide perfect segmentation in terms of human perception. Nevertheless, the algorithm can help pre-segment the video, and so reduce the workload of the human operator. Therefore, an algorithm that can provide no missed detection and the minimum false alarms is highly desirable. These results show that the precisions of all the approaches are low due to the existence of strong noise and motion in the video sequences, and approaches based on the HSV color space and CPAM achieve the highest recall values. Figure 1 shows some of the false scene breaks due to the existence of strong noise and motions. More importantly, these two methods can achieve zero missing in video sequence 1.

From the results with video sequence 1, HSV and all representations using CPAM can obtain the highest recall rate, and the joint ASP & CSP method also achieves the highest level of precision. Therefore, it is clear that the joint ASP & CSP method outperforms other methods. From the results with video sequence 2, monochrome, HSV and joint ASP & CSP achieve the highest recall rate, while the joint ASP & CSP method also obtains the highest precision level. In other words, the joint ASP & CSP method achieves the best performance level in automatic scene cut detection.

Lgerca_1.mpg					
Low-level Representations	CD	FN	FP	Recall	Precision
Monochrome	41	1	73	0.9762	0.3596
Color Space					
256 RGB Color	23	19	88	0.5476	0.2072
64 RGB Color	21	21	125	0.5	0.1438
32x32 RgBy	39	3	51	0.9286	0.4333
HSV	42	0	66	1	0.3889
HMMD	41	1	44	0.9762	0.4824
CPAM					
ASP	42	0	115	1	0.2375
CSP	42	0	83	1	0.3360
Joint ASP & CSP	42	0	39	1	0.5063

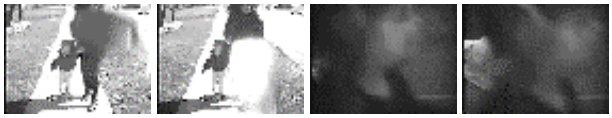
Table 1 Performances of different low-level representations for scene cut detection based on video sequence 1.

Lgerca_2.mpg					
Low-level Representations	CD	FN	FP	Recall	Precision
Monochrome	38	4	162	0.9048	0.1900
Color Space					
256 RGB Color	28	14	82	0.6667	0.2545
64 RGB Color	24	18	134	0.5714	0.1519
32x32 RgBy	35	7	320	0.8333	0.0986
HSV	38	4	177	0.9048	0.1767
HMMD	36	6	133	0.8571	0.2130
CPAM					
ASP	32	10	94	0.7619	0.2540
CSP	37	5	222	0.8810	0.1429
Joint ASP & CSP	38	4	150	0.9048	0.2021

Table 2 Performances of different low-level representations for scene cut detection based on video sequence 2.

6. CONCLUSION

In this paper, we have introduced the idea of using CPAM for scene representation and scene cut detection. In our experiments, the entropic thresholding technique was used to determine the optimal threshold, and the two MPEG-7 video sequences were used to evaluate the performances of the CPAM and several histogram-based low-level representations. The experimental results show that the CPAM method outperforms other methods in terms of scene break detection.



(a) (b) (c) (d)

Figure 1. (a) and (b) illustrate the false scene breaks due to the existence of strong noise. (c) and (d) illustrate the false scene break due to the existence of strong motion between two consecutive frames.

7. REFERENCES

[1] G. Qiu, “Indexing chromatic and achromatic patterns for content-based color image retrieval,” Pattern Recognition, Vol. 35, No. 8, pp. 1675-1686, Aug. 2002.

[2] S. Antani, R. Kasturi and R. Jain, “A survey on the use of pattern recognition methods for abstraction, indexing and retrieval of images and video,” Pattern Recognition, Vol. 35, No. 4, pp. 945-965, Apr. 2002.

[3] A. M. Ferman, A. M. Tekalp and R. Mehrotra, “Robust Color Histogram Descriptors for Video Segment Retrieval and Identification,” IEEE Transactions on Image Processing, Vol. 11, No. 5, pp. 497 –508, May 2002.

[4] J. Yu and M. D. Srinath, ”An efficient method for scene cut detection,” Pattern Recognition Letters, Vol. 22, No. 13, pp. 1379-1391, Nov. 2001.

[5] B. S. Manjunath,, J. R. Ohm, V. V. Vasudevan, and A. Yamada, “Color and Texture Descriptors,” IEEE Trans. CSVT, Vol. 11, No. 6, pp. 703-715, Jun. 2001.

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

[7] W. A. C. Fernando, C. N. Canagarajah, D. R. Bull, “A unified approach to scene change detection in uncompressed and compressed video,” IEEE Transactions on Consumer Electronics, Vol. 46, No. 3, pp. 769-779, Aug. 2000.

[8] S. W. Lee; Y. M. Kim; S. W. Choi, “Fast Scene Change Detection using Direct Feature Extraction,” IEEE Transactions on Multimedia, Vol. 2, No. 4, pp. 240-254, Dec. 2000.

[9] M. Swain and D. Ballard, “Color Indexing,” Computer Vision, Vol. 7, No. 1, pp. 11-32, 1991.