MVA'94 IAPR Workshop on Machine Vision Applications Dec. 13-15, 1994, Kawasaki

A Robust Approach for Camera Break Detection in Color Video Sequence [1]

John Chung-Mong LEE and Dixon Man-Ching IP

Department of Computer Science
Hong Kong University of Science and Technology
Clear Water Bay, Hong Kong

Abstract

A convenient and efficient indexing system is as essential to managing video information as it is to managing text. Creating an index requires partitioning the video into meaningful segments, assigning index terms to individual segments and combining the index terms to form a table of contents or an inverted index file. To date, video partitioning methods are still in the research stage. This paper introduces a new camera break detection algorithm, namely the Local Selective HSV Histogram Comparison. Color and spatial information are used in this new algorithm to complement the deficiency of the existing algorithms. The performance of four existing algorithms together with our proposed one are evaluated with image sequences of various scenarios, including camera movements, zooming, moving objects, deformable objects and video with degraded image quality. In addition, an adaptive thresholding technique is proposed to replace the general use of a single threshold for the camera break detection algorithms.

1 Introduction

Nowadays, we are seeing a revolution in the way information is produced, stored, and communicated. More and more information is becoming available in the market through video. Managing video information is now as important as managing textual information. Video has been used to record historical information, educational materials, medical examinations and even aerial time-sequence images taken from satellites. As the amount, complexity, and inter-relation of video information grows, the need for more intelligent video manipulating tools becomes evident. To date, the only interface available in the commercial market is a video cassette player or a laserdisc player which reads and decodes video signals, and displays the images (content) of the video on a television screen. Search of particular frames or video segments is limited to manual sequential search. For more efficient use of video sources, an intelligent system that allows systematic management and indexing of video sources is a must. Indexing a video involves partitioning it into meaningful segments of frames and assigning index terms to each segment according to its content. These index terms can then be put together to form a table of content (TOC) or an inverted index file. Users can browse through the TOC to obtain an overview of the video or search for a particular video segment via the indices.

Before the content of individual segments can be analyzed and assigned index terms, some sort of video partitioning methods must be developed. The partitioning process involves the detection of boundaries between uninterrupted segments (camera shots) of scenes. Segment boundaries can be classified into two categories: *gradual transition* and *instantaneous change* [1]. Our research focuses on the detection of instantaneous changes - *camera breaks*. They are perceived as instantaneous change from one shot to another. Currently, video partitioning is done manually. This is a tedious and time-consuming task which requires trained professionals. Ueda is developing a multimedia authoring system, called IMPACT [2] which aims to develop a system that can assist non-professional users to edit and create motion picture data. This system provides an automatic-cut-separation function which is also based on automatic video partitioning methods [3].

In this paper, a new approach for color video partitioning based on the ideas of human color perception is proposed. Section 2 describes some of the existing video partitioning algorithms. Section 3 discusses our proposed method in details. Video with various scenarios are selected to test all these algorithms together with our proposed one, the results are presented in section 4.

2 Related Work

Automatic video partitioning methods are not mature, and research in this area continues. Some of the proposed methods [3, 4, 5] detect camera breaks; however, they are not robust in the sense that they either produce too many false alarms or miss many real breaks. Some of these basic methods are: *Grey-Level Histogram Comparison*, *Color (RGB) Histogram Comparison*, *Pairwise Comparison*, and *Local Grey-Level Histogram Comparison*. Grey-Level Histogram Comparison makes use of grey-level histograms of frames. If the difference between the histograms of two consecutive frames exceeds a certain threshold, a camera break is identified between the two frames. The RGB method is similar to the Grey-Level Histogram Comparison, it however compares RGB histograms of two consecutive frames. In Pairwise Comparison, consecutive frames are examined point by point. Each pixel is compared with the corresponding pixel in the successive frame. If the overall difference exceeds

[1]This research was supported by the Sino Software Research Centre, Hong Kong

a certain threshold, a camera break is declared. These camera break detection algorithms can be classified into two categories: pixel-to-pixel comparison and histogram comparison. In general, methods based on pixel-to-pixel comparison are more sensitive to object movements and camera movements, while spatial information is lost in histogram approach. Our proposed approach is based on local histogram comparison, which can reduce false alarms introduced by camera movements and at the same time retain enough spatial information to produce accurate results.

3 Local Selective HSV Histogram Comparison

The accuracy of the camera break detection algorithms is highly dependent on the selection of thresholds. If the algorithms can give sharp peaks for the camera breaks and relatively low frame-to-frame differences for the non-break sequence, then we can easily select a threshold with high accuracy in the detection of camera breaks. As a consequence, we have designed the Local Selective HSV Histogram Comparison algorithm which aims to suppress the frame-to-frame differences of non-breaks and exaggerate those of real boundaries. Our proposed method compares images in the HSV color space to reduce the differences caused by change in intensity or shade from frame to frame of the same scene. Figure 1 gives an overview of our approach.

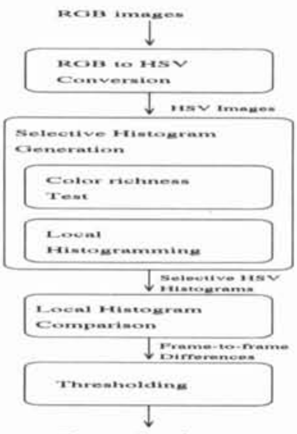

Figure 1: Overview of approach

3.1 HSV Color Space

Human is able to make good color judgements and to use intensity information, this implies that we use a transformed, non-RGB basis for color space. Actually, we perceive color as hue, saturation and value (intensity). The HSV color space is a hexcone (six-sided pyramid) model. The top of the hexcone corresponds to intensity, $V = 1$, which contains the relatively bright colors. Hue, H corresponds to the pure color pigment, and saturation S describes the purity of colors (red is highly saturated and pink is unsaturated.) Intuitively, HSV color space can be obtained by projecting the RGB color cube along the principle diagonal from white to black. In our algorithm, the RGB values of each pixel are transformed into the HSV values of the HSV color space by precise projections [6].

3.2 Selective HSV Approach

The use of the hue attribute makes our algorithm insensitive to shades since hue is independent of saturation and intensity. However, hue is unstable when the saturation or the intensity is very low. We need to separate the pixels with meaningful hue values from those contain only achromatic information.

In the creation of the selective HSV histogram, each pixel is first tested on the richness of color information. If it contains rich color information, that is, if the pixel has a certain degree of brightness (a high V) and saturation (a high S), we then classify the pixel into a discrete color based on its hue (H). Conversely, if a pixel fails in this test, it will be classified by its intensity value.

The selective histograms are formulated as follows.

$$B_{hue}(i,h) = \sum_{x=0}^{W}\sum_{y=0}^{H} I_{hue}(i,h,x,y)$$

$$B_{grey}(i,l) = \sum_{x=0}^{W}\sum_{y=0}^{H} I_{grey}(i,l,x,y)$$

$$I_{hue}(i,h,x,y) = \begin{cases} 1 & \text{if } HueLevel(i,x,y) = h \text{ AND} \\ & Saturation(i,x,y) > T_{saturation} \text{ AND} \\ & Value(i,x,y) > T_{value} \\ 0 & \text{otherwise} \end{cases}$$

$$I_{grey}(i,l,x,y) = \begin{cases} 1 & \text{if } (Saturation(i,x,y) \le T_{saturation} \text{ OR} \\ & Value(i,x,y) \le T_{value}) \text{ AND} \\ & GreyLevel(i,x,y) = l \\ 0 & \text{otherwise} \end{cases}$$

The frame-to-frame difference,

$$d(i,i+1) = \sum_{h=0}^{H-1} |B_{hue}(i,h) - B_{hue}(i+1,h)| + \sum_{l=0}^{G-1} |B_{grey}(i,l) - B_{grey}(i+1,l)|$$

where $B_{hue}(h)$ and $B_{grey}(l)$ denote the histogram value at a particular hue level h and grey level l respectively. $T_{saturation}$ and T_{value} are the thresholds used to determine whether the color information is rich enough to use.

Figure 3a, b, c and d show the results of applying four existing algorithms to a video sequence of a movie "A Better Tomorrow"; the frame-to-frame differences obtained from different algorithms are plotted against the frame numbers. The markers correspond to real camera breaks in the video sequence. Figure 3e shows the results of applying the selective HSV histogram comparison on the same video sequence. With this selective approach, the frame-to-frame difference of non-breaks is reduced in comparison with the other algorithms. However, the peaks which correspond to the camera breaks are not distinctive enough to be separated from the other non-break sequences by simple thresholding. Obviously, this is the result of using global histogramming. The problem can be illustrated clearly by figure 2 where two completely different images share an identical histogram. We address this problem by incorporating spatial information with a local histogramming technique.

3.3 Local Histograms

To obtain local selective histograms, a frame is divided into uniform, non-overlapping regions where selective HSV histograms are computed. These local histograms of corresponding regions from two consecutive

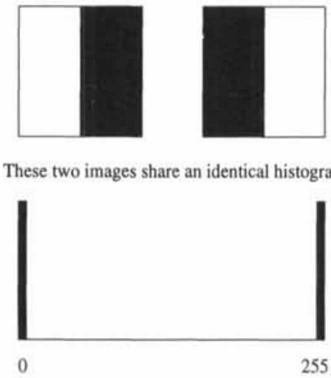

Figure 2: Global histogram fails to describe spatial differences

frames are then compared to obtain the frame-to-frame differences. The difference is computed as follows.

$$\begin{aligned} d(i, i+1) &= \sum_{p=0}^{P-1}\sum_{q=0}^{Q-1} H_{diff}(i, i+1, p, q) \\ H_{diff}(i, i+1, p, q) &= \sum_{h=0}^{H-1} |B_{hue}(i, p, q, h) - B_{hue}(i+1, p, q, h)| + \\ &\quad \sum_{l=0}^{G-1} |B_{grey}(i, p, q, l) - B_{grey}(i+1, p, q, l)| \end{aligned}$$

where P and Q are the width and height of the regions, and P x Q equals the total number of regions. Figure 3f shows the results of applying the Local Selective HSV Histogram Comparison to the same video sequence. The peaks corresponding to camera breaks are greatly enhanced such that a threshold can be selected easily to separate all the camera breaks from the video sequence.

4 Experimental Results

All the camera break detection algorithms have been implemented on Macintosh platform which consists of a Macintosh Quadra 700, a Pioneer LD-V8000 video laserdisc player and a 24-bit color RasterOps MediaTime video board. The laserdisc player is connected to the computer through a RasterOps MediaTime video board, which digitizes images from laserdisc player and display at a rate of 30 frames per second (fps).

We have tested these algorithms with various video segments including scenes with moving objects, camera zooming and panning, complex road scenes, fade-out (special effect), noisy scenes (eg. raining, gun flare), indoor/outdoors scenes, video with degraded quality (fluctuating in intensity) and very dark video images. Table 1 summaries the experimental results of applying different algorithms on various scenarios. The experimental results shows that our proposed Local Selective HSV Histogram Comparison always gives superior results in comparison to the other four algorithms. However, none of them provides a satisfactory performance with very dark video images.

The size of images used in the experiments is 160 x 120. The color conversions of RGB to grey levels and RGB to HSV take ≈ 5.5 seconds per frame and ≈ 7.1 seconds per frame respectively. Table 2 summaries the running time of individual algorithms (excluding the color conversions). It indicates that our proposed Local Selective HSV Histogram Comparison has the potential to be implemented in real-time, provided that we have the hardware to do the color conversions.

5 Concluding Remarks

In this paper, we introduce a new camera break detection algorithm for automatic video partitioning. The human perceptive HSV color space is exploited intelligently by selectively comparing the color of pixels of images according to their richness. From the experimental results, we conclude that our proposed Local Selective HSV Histogram Comparison gives superior results to the other four existing methods. However, none of the camera break detection algorithms provides a satisfactory performance on very dark video images. Further research in this area is required. Although we have many different camera break detection algorithms, selection of a reasonable threshold for each individual algorithm is very difficult. Our preliminary study on the selection of thresholds shows that the threshold selection method proposed by Zhang [7] fails in many scenarios. Their method derives a single threshold based on a statistical analysis of the frame-to-frame differences of a whole video, which means camera breaks can never be obtained in real-time. Moreover, it is obvious that the use of one single threshold for a whole video does not work, and we are currently working on an adaptive thresholding technique where adaptive threshold is derived from the frame-to-frame differences of adjacent frames instead of the whole video, which gives encouraging results.

References

[1] D. Bordwell and K. Thompson, *Film Art: an introduction, 2nd ed.* New York: Distributed by Random House, 1986.

[2] H. Ueda, T. Miyataka, and S. Yoshizawa, "Impact: An interactive natutal-motion-picture dedicated multimedia authoring system," *Human Factors in Computing Systems CHI'91 Conf. Proc.*, pp. 343–350, 1991.

[3] A. Nagasaka and Y. Tanaka, "Automatic scene-change detection method for video work," *Proc. 40th National Conf. Information Processing Society of Japan, 1Q-5*, 1990.

[4] R. Jain and H.H.Nagel, "On the analysis of accumulative difference pictures from image sequences of real world scenes," *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. PAMI-1, no. 2, pp. 206–214, 1979.

[5] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," *Visual Database Systems, II*, pp. 113–127, 1992.

[6] J. D. Foley, A. van Dam, S. K. F. and J. F. Hughes, *Computer Graphics: principles and practice, 2nd ed.* Addison-Wesley, 1990.

[7] H. Zhang, A. Kankanhalli, and S. W. Smoliar, "Automatic partitioning of full-motion video," *Multimedia Systems, ACM*, vol. 1, pp. 10–28, 1992.

Table 1: Experimental results of applying different camera detection algorithms on various scenarios

Scenes	Grey-Level Histogram Comparison	Local Grey-Level Histogram Comparison	Pairwise Comparison	Selective HSV Histogram Comparison	Local Selective HSV Histogram Comparison
Moving object	No false alarm	No false alarm	No false alarm	No false alarm	No false alarm
Zooming /w panning	No false alarm	No false alarm	No false alarm	No false alarm	No false alarm
Road	No false alarm	No false alarm	No false alarm	No false alarm	No false alarm
Moving Obj. /w panning	No false alarm	No false alarm	No false alarm	No false alarm	No false alarm
Fade-out	Many false alarm	Many false alarm	No false alarm	No false alarm	No false alarm
Raining	Many false alarm	Many false alarm	Many false alarm	Many false alarm	No false alarm
Outdoors	2 missing	1 missing	All 7 detected	1 missing	All 7 detected
Indoor (dialog)	No clear cut	No clear cut	1 missing	5 missing	All 13 detected
Fighting /w firearms	No clear cut	No clear cut	No clear cut	No clear cut	All 15 detected
Fighting in dark	No clear cut	No clear cut	No clear cut	Detected 3 out of 6	Detected 2 out of 6

Table 2: Running time of different camera detection algorithms (excluding color conversions)

	speed	for 5-min video
Grey-level Histogram Comparison	≈ 0.03 sec/frame	≈ 5 mins
Pairwise Comparison	≈ 0.04 sec/frame	≈ 6 mins
Local Grey-level Histogram Comparison	≈ 0.13 sec/frame	≈ 20 mins
Selective HSV Histogram Comparison	≈ 0.03 sec/30frames	≈ 9 secs
Local Selective HSV Histogram Comparison	≈ 0.27 sec/30frames	≈ 81 secs

Video sequence: Indoor scene (a dialog)

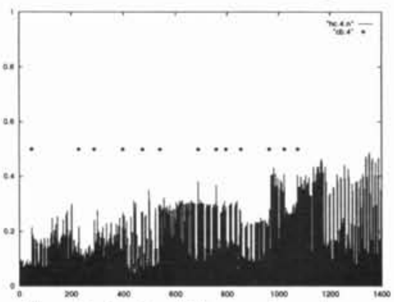

Figure 5a Grey Level Histogram Comparison

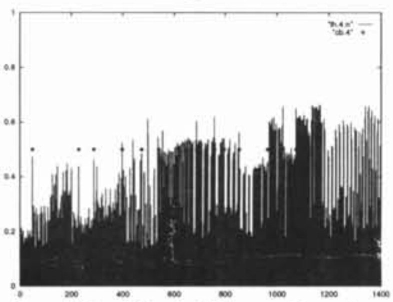

Figure 5b Local Grey-level Histogram Comparison

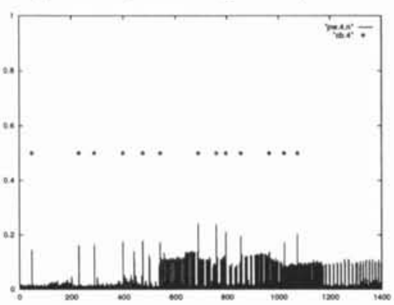

Figure 5c Pairwise Comparison

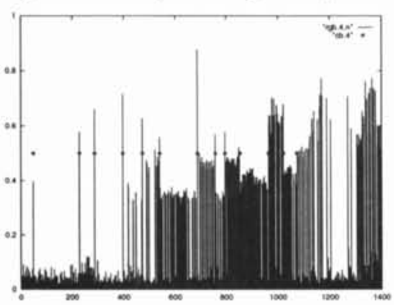

Figure 5d RGB Histogram Comparison

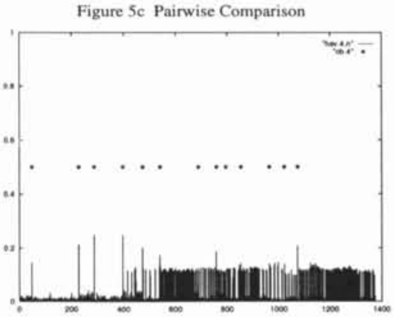

Figure 5e Selective HSV Histogram Comparison

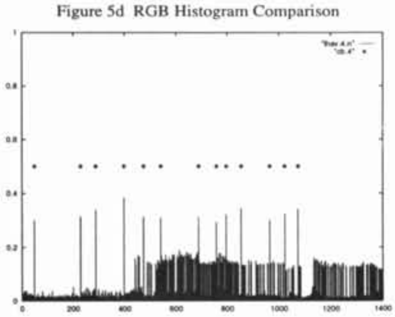

Figure 5f Local Selective HSV Histogram Comparison

Figure 3: Experimental results of applying difference camera break detection algorithms on a video sequence of indoor scene