IJESRT

INTERNATIONAL JOURNAL OF ENGINEERING SCIENCES & RESEARCH TECHNOLOGY

Video Shot Cut Boundary Detection using Histogram

Vrutant Hem Thakore

Department of Electrical and Electronics Engineering. Veermata Jijabai Technological Institute, Mumbai – 400 019. Maharashtra, India

vrutantthakore@gmail.

Abstract

In this paper we will be discussing about Video shot boundary detection using histogram differences. We can use color histogram as well as gray scale histogram. This technique is able to differentiate abrupt shot boundary detection. The aim of this method is to provide a simple and fast algorithm to work in real time with high performance in video indexing tool. In this paper we give the overview of our approach and summary of the result.

Keywords: **S**hot boundary detection, gray-scale histogram, color histogram, accumulating histogram differences.

Introduction

Shot Boundary Detection (SBD) appeared in the 1990's. It has been deeply studied in recent years and has found applications in many domains such as video indexing, video compression, video access and others. It is a basic task of video temporal segmentation. It is closely related to the video making method. It is a way to divide video in many different parts, which are easy to manage.

Video Shot Boundary Detection algorithms have to challenge the difficulty of finding shot boundaries in the presence of camera, motion of the objects and illumination variation. Moreover, different shot boundaries may present very different appearances like abrupt temporal changes or smooth temporal transitions.

Shot Boundary Detection has been an area of active research. Many automatic techniques have been developed to detect frame transition in video sequences. There are all together many ways of detecting shot boundary detection. The simplest way is by comparing the pixel values of 2 corresponding frames [1]. The way alternate to pixel matching is using a gray-scale or color histogram [2]. Another way of detecting shot boundary is by edge change [3]. Other methods use predefined models, objects, regions or spatio-temporal sub-sampling to detect camera breaks [4].

In this paper we present a simple and a fast method for detecting video shot boundary based on gray-scale and color histogram. The algorithm detects abrupt shot changes both using gray-scale as well as color frames. The simplicity of this method is due to its low complexity of computation and that all the further analysis is based on 1D signal.

The paper is organized as follows. After this section the section II will describe the methods of detecting abrupt video shot boundary. Section III will give the overview of the results obtained with the methods discussed in the paper. Section IV will comment some conclusions about the result and some future improvements.

Shot Boundary Detection

Introduction

Visual Discontinuities are the basis of any shot boundary detection method. Measuring the degree of similarity between frames in a given shot whose visual features has to be extracted in the detection process. This measure denoted as g (n, n+k), is related to the difference or discontinuity between the n'th and the (n+k)'th frame where k>0. There are several methods to compute the value of g (n, n+k) in a video sequence the difference between frames is one of the simplest.

$g(n, n+k) = \sum | I_n(x, y) - I_{n+k}(x, y) |$

where I (x, y) is the intensity value of the image at the x and y position. Methods based on absolute difference compare this value with a given threshold in order to determine the occurrence of a significant change in the image sequence. There is a high probability of false alarm because the measure of discontinuity g (n, n+k) changes considerably when the object or camera is in motion or any luminance changes.

An alternate method to pixel matching is by statics of global color. Histograms capture the color distribution of the image. However, the method based on color histogram is sensitive to panning or zooming.

Histogram Features

In this paper we are dealing with two types of histograms. One is the gray-scale histogram and the other is the color histogram. The gray-scale histogram gives us video shot change detection by converting digital color images into gray-scale ranging from 0-255.

Usually, digital images are represented in RGB color space.

In our work we have 24 bits/pixel (8 bits/pixel for R, G and B respectively). We define the color histogram as an array of M elements, where M is the number of different possible colors within the color space.

Alternately, we can also use HSV to evaluate color histogram due to its similarity and perceptibility characteristics. This color space is defined according to the human color perception. Perceptibly similar color are situated within close quantization levels. In addition, similarity between two colors can also be found out by evaluating the distance between their color spaces. Thus we have to transform RGB to HSV using our own transformation algorithm as there is no direct transform available.

Detection Principle

The flow chart shown depicts the measures for the procedure adopted for the color to gray transformation and shot boundary detection.

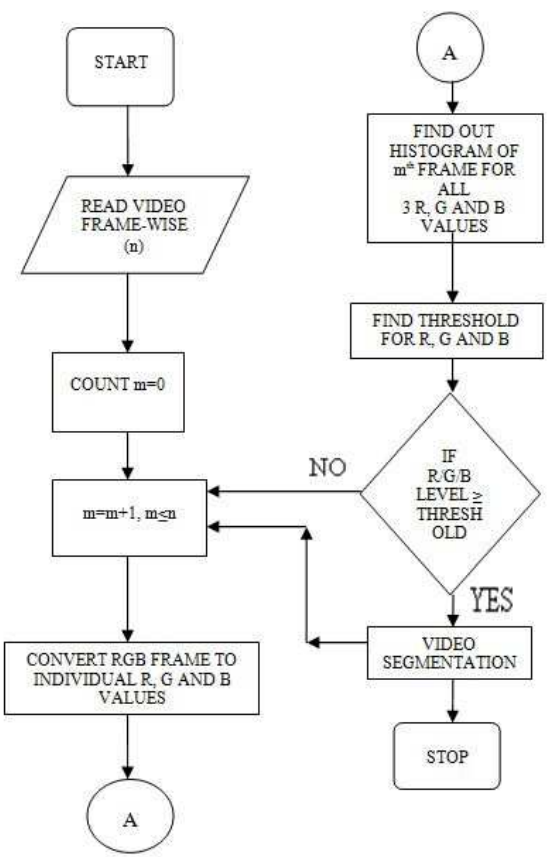

Fig. 1. Flowchart for Gray Scale Histogram

Histogram Difference

The method presented in the paper for video shot boundary detection is based on the computation of differences of gray-scale as well as color histograms between frames as a measure of discontinuity. This difference can be computed as the absolute difference between the bin values in either case.

$$dRGB\ (X, Y) = \sum | hx\ (i) - hy\ (i) |$$

where *h*x is color histogram of image X which contains M different bins.

In our paper the evaluation of color histogram in the RGB space with M bins per histogram is considered. As we have mentioned in the earlier part regarding HSV we have observed that the results of both these methods are almost the same so we do not use the HDV method because it just increases the computation cost and also an extra algorithm is required to convert RGB to HSV.

The shot boundary detection method is based on the difference between the color histograms of frames belonging to a video sequence.

The difference is computed as

$$Histdiff\ [i] = \sum | hi\ (j) - hi\text{-}1(j) |$$

where *h*i is the color histogram with M bins of frame *i* corresponding to the video sequence.

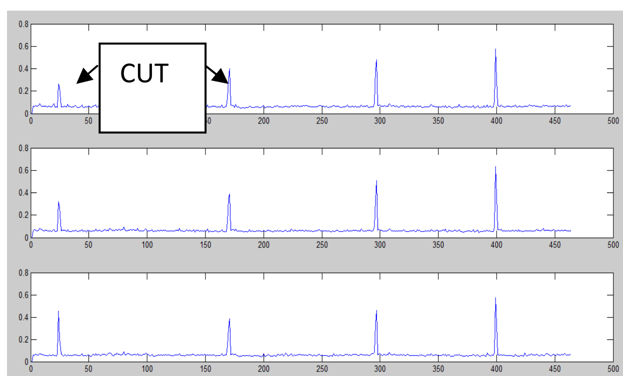

Fig. 2. Color Histogram difference signal of a frame sequence with cut boundaries

The above figure shows the result of computing the color histogram difference for a continuous video sequence. It shows that the video sequence is for **10 seconds.** As we can see that the above figure is divided into 3 subplots each plotting histogram difference for Red, Green and Blue individually.

We can see that the peaks that appear in the figure can be of different sizes. The main reason for a peak to appear is because of a large discontinuity

 ISSN: 2277-9655

occurring between histograms. This peak can be associated to an abrupt transition or cut. These could be easily recognized from other video effects because they always present big amplitude. Ideally, an abrupt change can be represented as a delta function

$$HistDif\ \mathrm{cut}\,[i] = \alpha i \,.\, \delta(i - i\ cut)$$

where αi represents the amplitude of the delta function and *icut* is the frame number where the cut occurs.

Boundary Detection

Now that we have found out the histogram differences signal *HistDif [i]* of a video sequence the next step is to convolve the signal with a rectangular window of width *W*

$$HistDifconv[i] = HistDif\,[i] * 1/W \,.\, \mathrm{rect}(i/W)$$

With this signal processing operation we are smoothing *HistDif [i]* signal so that small variations due to histogram difference computation are eliminated. After the convolution process the cut signal have been modified. In case of cuts, after convolution we obtain a rectangular shaped signal where the middle of the rectangle is considered the point or frame of abrupt transition.

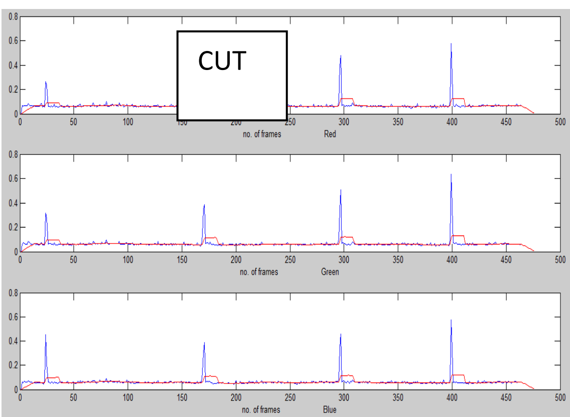

Fig. 3. Color Histogram difference signal (blue) and convoluted signal with a rectangular window W width (red)

In case of camera/object movement close to the cut, the signal *HistDifconv[i]* loses the rectangular shape and increases the difficulty of detection.

The next step is to detect the cuts now. For cut detects, we aim to detect and identify signals with rectangular pattern in the convolved signal of the color histogram difference *HistDifconv[i]*.An effective method that reduces the distortion produced by object/camera is to apply the first derivative to *HistDifconv[i]* signal

$$HistDif(deriv)conv[i]=[1,-1]*HistDifconv[i]$$

As we can see in the figure below, applying the first derivative a positive peak followed by a negative peak is obtained. The distance between these two peaks is *W*, the width of the rectangular window used during convolution.

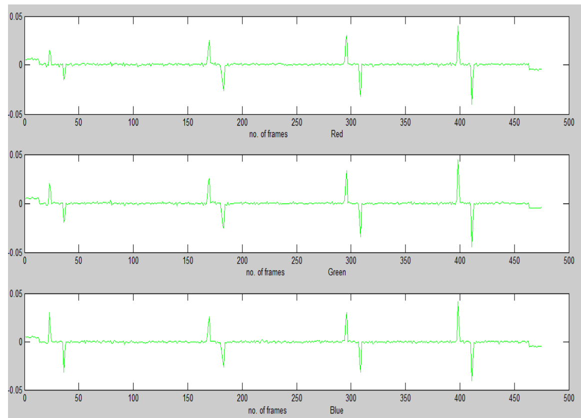

Fig. 4. First derivative of convolved color histogram difference signal (green)

A frame that fulfills this condition, a positive peak and a negative peak shifted *W* samples and the peak values exceeding a certain threshold is considered as a cut within the video sequence. Specifically the middle frame between the positive and the negative peak is to be considered as the beginning of a new shot, which coincides with the center of the rectangular shape signal found in the convoluted signal. The selection of the threshold is not difficult because cut signals appear well distinguished from noise. The threshold value for cut detection is used in the first derivative signal is obtained from the percentage of the maximum possible value of the color histogram difference. In this way, a cut is only identified when an abrupt variation overcomes a certain percentage of color change in a video sequence.

The advantage of the cut detector method based on color histogram difference is that it takes into account the global variation of the image, being therefore less sensitive to camera and object motion.

Experimental Results

Experimental results were obtained from processing a matlab code. First, we applied the cut boundary detection method. After the computation of the color histogram difference and convolution with rectangular window of *W*=13 samples, and we tried to reduce the effect of camera and object motion over the detection of abrupt shot change. The obtained results are shown below in the table where we show results of different video sequences with different threshold level for cut detection.

$$Precision = \frac{Correct}{Correct+False\ Positive}$$

Video sequence	Runs	Precision		
		R	G	B
1 (ddlj)	1	0.8750	0.8750	0.8750
	2	0.4286	0.4286	0.2857
2 (3 Idiots)	1	0.6154	0.4074	0.4231
	2	0.3486	0.1852	0.1923
3 (rajneeti)	1	0.5385	0.4091	0.7097
	2	0.5385	0.3000	0.2069

Table 1. The Precision values for the test data

With the method used above we can obtain good overall results for cut shot boundary detection but not for gradual shot boundary detection due to the use of easy method. Another point to improve is the behavior of the motion in the presence of motion and a low variation of color in the scene that leads to false alarm.

Conclusion

In this paper we have shown shot boundary detection method using color histogram. Usually the detection of cut boundaries is simple in front of a camera and moving object. Whereas for smooth shot boundary detection we know that we have to apply a complimentary method which takes into account the linear color variations in the image.

In the future we count on using more accurate techniques for shot boundary detection and motion analysis methods to overcome missed shots and false alarms.

References

[1] Jordi Mas and Gabriel Fernandes, “Video Shot Boundary Detection Based on Color Histogram”, Digital Televison Center (CeTVD), La Salle School of Engineering, Ramon Llull University Barcelona, Spain.

[2] W.A.C. Fernando, D.R. Bull, C.N.Canagarajah, “Wipe Scene Change Detection in Video Sequence”, Image Communications Group, Centre for Communication and Research, University of Bristol, United Kingdom.

[3] Quing-Ge Ji, Jiang-Wei Feng, Jie Zhao, Zhe Ming Lu, “Effective Dissolve Detection Based on Accumulating Histogram Difference and the support point.”

[4] Zhang, H.J., Kankanhalli, A., and Smoliar, S.W., "Automatic Partitioning of Full-motion Video", Multimedia Systems (1993) Vol. 1, No. 1, pp. 10-28

[5] E. Garcia, “Implementing a Histogram for Image Processing Applications," Xilinx Xcell Magazine, (38):40-46, 2000.

[6] Ueda, H., Miyatake, T., and Yoshizawa, S., "IMPACT: An Interactive Natural-motion-picture Dedicated Multimedia Authoring System", in proceedings of CHI, 1991 (New Orleans, Louisiana, Apr.-May, 1991) ACM, New York, 1991, pp. 343-350.

[7] Little, T.D.C, Ahanger, G., Folz, R.J., Gibbon, J.F., Reeve, F.W., Schelleng, D.H., and Venkatesh, D., "A Digital On- Demand Video Service Supporting Content-Based Queries", Proc. ACM Multimedia 93, Anaheim, CA, August, 1993, pp. 427-436.

[8] A Hanjalic, “Shot Boumdary Detection: Unraveled and Resolved?”, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 12, No. 2, pp.90-104 February 2002

[9] Kasturi, R. and Jain R., "Dynamic Vision", in Computer Vision: Principles, Kasturi R., Jain R., Editors, IEEE Computer Society Press, Washington, 1991.

[10] J. S. Boreczky and L. A. Rowe, “Comparison of video shot boundary detection techniques,” in Storage and Retrieval for Image and Video Databases (SPIE), 1996, pp. 170–179.

[11] S.H. Han, K.J. Yoon, I.S. Kweon, “A New Technique for Shot Detection and Key Frames Selection in Histogram Space”, Robotics and Computer Vision Lab., Republic of Korea.