Analysis of Histogram Based Shot Segmentation Techniques for Video Summarization

Anand P. Mankodia*[1] and Prof. Satish K. Shah[2]

[1]Ph.D. Student, U. V. Patel College of Engineering, Ganpat University
[2]Professor Retd.and Ex-Head,Electrical Engg. Dept.,MS University of Baroda

December 27, 2014

Abstract

Content based video indexing and retrieval has its foundations in the analyses of the prime video temporal structures. Thus, technologies for video segmentation have become important for the development of such digital video systems. Dividing a video sequence into shots is the first step towards VCA and content-based video browsing and retrieval. This paper presents analysis of histogram based techniques on the compressed video features. Graphical User Interface is also designed in MATLAB to demonstrate the performance using the common performance parameters like, precision, recall and F1.

Keywords : Histogram Difference, Weighted Difference, Histogram Intersection, χ^2 Histogram Matching, Block Based Histogram Comparison.

1 Introduction

There are many approaches for shot boundary detection like pixel based methods, block based methods, feature based methods, histogram based methods etc [1][2][3][4][5][6] [7] [8][9][10][11] [12]. Histogram is a graphical representation of the tonal distribution in digital image or frame. It plots the number of pixel for each tonal value. By viewing at the histogram for a specific frame viewer can judge the entire tonal distribution. We can generate histogram for entire frame or we can divide the frame into number of sections and then generate the histogram for each section. For gray scale image or frame, the histogram represents the plot of number of occurrence of gray levels in the frame against the gray level values. The histogram provides a convenient summary of the intensities in a frame. The histogram provides more insight about image contrast and brightness.

*mankodia@gmail.com

2 Histogram Matching Methods [13][14][15][16][17][18]

Histogram difference, weighted difference, histogram intersection, χ^2 Matching and Block Based Comparison are discussed in this section.

2.1 Histogram Difference

This method based on gray-level histograms. Images are compared by computing a distance between their histograms, as shown in the following equation:[19]. Detection if

$$\sum_{v=0}^{v} = |H(I_t, v) - H(I_{t-1}, v)| > T \tag{1}$$

Where T is Threshold fixed by user and I_t and I_{t-1} is the frames at time interval t and t-1. As clearly shown in equation 1 if the difference value is greater than threshold (T), then only detection can be possible.

A similar method using only 64 bins for color histograms (2 bits for each color component of the RGB space). Using the notation $H64(I_t, v)$, the detection is defined by [20], Detection if

$$\sum_{v=0}^{63} = |H_{64}(I_t, v) - H_{64}(I_{t-1}, v)| > T \tag{2}$$

Similarly for the RGB and HSV compute three histogram differences, considering separately the three color components of the RGB space. The highest value is compared to a threshold for shot change detection.

2.2 Weighted Difference

In color images, some color components may have a bigger influence than others. So it is possible to detect shot changes by weighting the histograms of each color component depending on their importance [21]. Detection if

$$\sum_{k=1}^{3} \sum_{v=0}^{v} = \frac{L(I_t, C_k)}{L_{Mean}(I_t)} |H(I_t, C_k, v) - H(I_{t-1}, C_k, v)| > T \tag{3}$$

Where,$L(I_t, C_k)$ and $L_{Mean}(I_t)$ are respectively the luminance for the K^{th} color component of the frame I_t and the average luminance of the frame I_t considering all the color components.

Zhao et al use weight values for weighted histogram difference calculation. First calculate original histogram difference and after use min-max optimization for select best weight which is used in weighted histogram difference. The shot detection process is given by following equation which uses 12 operations per histogram bin [22].

$$\sum_{k=1}^{3}\sum_{v=0}^{v} = W(k,v)|H(I_t,C_k,v) - H(I_{t-1},C_k,v)| > T \tag{4}$$

Where,$W(k,v)$ denote best weigh value.

2.3 Histogram Intersection

Similarity between two images can also be evaluated by histogram intersection. Histogram intersection is computed using different operators, for example a min function. Similarity ratio belonging to interval [0,1] is then compared to a given threshold. This comparison allows the detection of shot changes [13]. Detection if

$$\left(1 - \frac{1}{XY}\sum_{v=0}^{v} = min\left(|H(I_t,C_k,v), H(I_{t-1},C_k,v)\right)|\right) > T \tag{5}$$

Where, XY represents the number of pixels of frames processed.

Additional method for histogram intersection based shot detection is given by following equation [13]. Detection if

$$\left(1 - \frac{1}{XY}\sum_{v=0}^{v}\frac{min\left(|H(I_t,C_k,v), H(I_{t-1},C_k,v)\right)}{max\left(|H(I_t,C_k,v), H(I_{t-1},C_k,v)\right)}|\right) > T \tag{6}$$

2.4 χ^2 Histogram Matching

This is most efficient histogram based comparison method than the above discussed. Nagasakaet al have proposed a 64 bin histogram comparison based on χ^2 test. This comparison is then defined by[20]: Detection if

$$\left(\sum_{v=0}^{63}\frac{\left(H_{64}(I_t,v) - H_{64}(I_{t-1},v)\right)^2}{H_{64}(I_t,v)}\right) > T \tag{7}$$

With the assumption, $H_{64}(I_t,v) \neq 0$.
If this assumption does not hold, we use the following equation instead[20]: Detection if

$$\left(\sum_{v=0}^{63}\frac{\left(H_{64}(I_t,v) - H_{64}(I_{t-1},v)\right)^2}{H_{64}(I_{t-1},v)}\right) > T \tag{8}$$

With the assumptions $H_{64}(I_{t-1},v) \neq 0$ and $H_{64}(I_t,v) = 0$. This method is considered as more efficient than simple histogram comparison-based methods.

2.5 Block Based Histogram Comparison

Block sampling is used to increase the quality of shot boundary detection and decrease the calculation time. Block based method has advantages like relative insensitivity to noise and object motion. Color histograms computed on blocks of image noted $H(I_t, b, C_k, v)$ which gives shot boundary detection given by Swanberg et al.[23]

$$\left(\sum_{k=1}^{3}\sum_{b=1}^{3}\sum_{v=0}^{v}\frac{(H(I_t,b,C_k,v)-H(I_{t-1},b,C_k,v))^2}{H(I_t,b,C_k,v)+H(I_{t-1},b,C_k,v)}\right) > T \tag{9}$$

3 Performance Parameter Matrix [24][25][26][27][28][29][30]

i Correct(C):Number of shot changes that are correctly detected by algorithm.

ii Missed (M):Number of shot changes that actually present in video but algorithm fails to detect.

iii False-Positive (FP):Number of shot changes that are detected by algorithm but actually not present in video. Once we get the values of correct, missed, and FP, now we can derive performance parameter like Recall(R), Precision(P) and F1 measure from them. Definitions and equations of these parameters are as follow. Here, TH stands for Threshold Value; TF stands for Total number of Frames and D stands for Shot change detection.

iv Recall(R): It is defined as percentage of actual (desire) shot changes that are detected correctly by algorithm.

$$Rcall = \frac{Correct}{Correct + Missed} \tag{10}$$

v Precision (P): It is defined as percentage of total detected shot change by algorithm that is correct.

$$Precision = \frac{Correct}{Correct + FP} \tag{11}$$

vi F1 measure: It is defined as a measure that combines precision and recall; it is the harmonic mean of precision and recall.

$$F1 = \frac{2 \times R \times P}{R + P} \tag{12}$$

4 Simulation Results

Section presents simulation results for methods discussed in section 2 using MATLAB for three different types of video namely Animation video, NEWS video and Sport video using following procedure:

- Determine shot changes present in a video, by manually checking the frames of video using MATLAB programme to convert video into bunch of images.

- Apply different methods/algorithm to a video. Table 1 depicts the list of files developed for the methods.
- Comparing manual results with the algorithm result to obtain values of parameters like correct, missed, and False Positive (FP).
- Study effect of Variation of threshold values for each method on each video
- Use the values of parameter R, P, and F1 for comparative analysis.
- Tabulate the results (Tables 2, 3, 4, 5, 6, 7 are generated using MATLAB for the methods or algorithms).

Table 1: List of MATLAB Files Developed

Sr. No.	**File Name**	**Remarks**
1	hd.m	Histogram Difference Method
2	hi.m	Histogram Intersection Method
3	hm.m	Histogram Matching Method
4	wd1.m	Weighted Difference Method-1
5	wd2.m	Weighted Difference Method-2
6	bbhc.m	Block Based Hist. Comparison
7	sbd.fig	Graphical User Interface

Table 2: Histogram Difference Method

Type of video	**TH**	**TF**	**D**	**C**	**M**	**FP**	**R**	**P**	**F1**
Animation	15000	455	3	3	0	12	1	0.20	0.33
Animation	20000	455	3	3	0	0	1	1	1
Animation	25000	455	3	2	1	0	0.66	1	0.80
Animation	30000	455	3	2	1	0	0.66	1	0.80
NEWS	50000	750	5	5	0	24	1	0.17	0.29
NEWS	100000	750	5	5	0	0	1	1	1
NEWS	250000	750	5	5	0	0	1	1	1
NEWS	300000	750	5	4	1	0	0.80	1	0.89
SPORTS	100000	757	4	3	1	1	0.75	0.75	0.75
SPORTS	150000	757	4	2	2	0	0.50	1	0.67
SPORTS	200000	757	4	2	2	0	0.50	1	0.67
SPORTS	250000	757	4	1	3	0	0.25	1	0.40

Table 3: Histogram Intersection Method

Type of video	TH	TF	D	C	M	FP	R	P	F1
Animation	0.1	455	3	3	0	4	1	0.43	0.60
Animation	0.15	455	3	3	0	0	1	1	1
Animation	0.2	455	3	3	0	0	1	1	1
Animation	0.3	455	3	2	1	0	0.66	1	0.80
NEWS	0.4	750	5	5	0	1	1	0.83	0.90
NEWS	0.1	750	5	5	0	0	1	1	1
NEWS	0.3	750	5	5	0	0	1	1	1
NEWS	0.4	750	5	4	1	0	0.80	01	0.89
SPORTS	0.1	757	4	4	0	2	1	0.67	0.80
SPORTS	0.12	757	4	4	0	0	1	1	1
SPORTS	0.28	757	4	4	0	0	1	1	1
v SPORTS	0.3	757	4	3	1	0	0.75	1	0.86

Table 4: χ^2 Histogram Matching Method

Type of video	TH	TF	D	C	M	FP	R	P	F1
Animation	2500	455	3	3	0	1	1	0.75	0.86
Animation	3700	455	3	3	0	0	1	1	1
Animation	7500	455	3	3	0	0	1	1	1
Animation	10000	455	3	2	1	0	0.66	1	0.80
NEWS	5000	750	5	5	0	1	1	0.83	0.91
NEWS	7000	750	5	5	0	0	1	1	1
NEWS	80000	750	5	5	0	0	1	1	1
NEWS	90000	750	5	4	1	0	0.80	1	0.89
SPORTS	7000	757	4	4	0	5	1	0.44	0.62
SPORTS	10000	757	4	4	0	0	1	1	1
SPORTS	40000	757	4	4	0	0	1	1	1
SPORTS	50000	757	4	2	2	0	0.50	1	0.67

5 Graphical User Interface

Design Steps:
The three principal elements required to create a MATLAB Graphical User Interface as follows. Using these elements, the Screen shot of GUI created is shown in figure 1 for selection of technique.

1. *Components:* The types of components include graphical controls (pushbuttons, edit boxes, lists, sliders, etc.), static elements (frames and text strings), menus, and axes. Graphical controls and static elements are created by the function uicontrol, and menus are created by

Table 5: Weighted Difference Method-1

Type of video	**TH**	**TF**	**D**	**C**	**M**	**FP**	**R**	**P**	**F1**
Animation	2500	455	3	3	0	11	1	0.21	0.35
Animation	3700	455	3	3	0	0	1	1	1
Animation	5000	455	3	3	0	0	1	1	1
Animation	7500	455	3	2	1	0	0.66	1	0.80
NEWS	15000	750	5	5	0	8	1	0.38	0.56
NEWS	20000	750	5	5	0	0	1	1	1
NEWS	80000	750	5	5	0	0	1	1	1
NEWS	90000	750	5	4	1	0	0.80	1	0.89
SPORTS	15000	757	4	4	0	1	1	0.80	0.89
SPORTS	20000	757	4	4	0	0	1	1	1
SPORTS	35000	757	4	4	0	0	1	1	1
SPORTS	40000	757	4	2	2	0	0.50	1	0.67

Table 6: Weighted Difference Method-2

Type of video	**TH**	**TF**	**D**	**C**	**M**	**FP**	**R**	**P**	**F1**
Animation	2×10^6	455	3	3	0	3	1	0.50	0.66
Animation	3×10^6	455	3	3	0	0	1	1	1
Animation	4×10^6	455	3	3	0	0	1	1	1
Animation	5×10^7	455	3	2	1	0	0.66	1	0.80
NEWS	0.5×10^7	750	5	5	0	8	1	0.38	0.56
NEWS	0.7×10^7	750	5	5	0	0	1	1	1
NEWS	4×10^7	750	5	5	0	0	1	1	1
NEWS	4.5×10^7	750	5	1	4	0	0.20	1	0.33
SPORTS	1×10^7	757	4	4	0	3	1	0.57	0.73
SPORTS	1.2×10^7	757	4	4	0	0	1	1	1
SPORTS	2×10^7	757	4	4	0	0	1	1	1
SPORTS	2.5×10^7	757	4	2	2	0	0.50	1	0.67

the functions uimenu and uicontextmenu. Axes, which are used to display graphical data, are created by the function axes.

2. *Figures:* The components of a GUI must be arranged within a figure, which is a window on the computer screen. However, empty figures can be created with the function figure and can be used to hold any combination of components.

3. *Callbacks:* Finally, there must be some way to perform an action if a user clicks a mouse on a button or types information on a keyboard. A mouse click or a key press is an event, and

Table 7: Block Based Histogram Comparison

Type of video	**TH**	**TF**	**D**	**C**	**M**	**FP**	**R**	**P**	**F1**
Animation	5000	455	3	3	0	1	1	0.75	0.86
Animation	7500	455	3	3	0	0	1	1	1
Animation	10000	455	3	3	0	0	1	1	1
Animation	15000	455	3	1	2	0	0.33	1	0.50
NEWS	10000	750	5	5	0	4	1	0.56	0.71
NEWS	12000	750	5	5	0	0	1	1	1
NEWS	70000	750	5	5	0	0	1	1	1
NEWS	80000	750	5	4	1	0	0.80	1	0.89
SPORTS	15000	757	4	4	0	10	1	0.29	0.44
SPORTS	20000	757	4	4	0	0	1	1	1
SPORTS	40000	757	4	4	0	0	1	1	1
SPORTS	45000	757	4	3	1	0	0.75	1	0.86

the MATLAB program must respond to each event if the program is to perform its function. The code executed in response to an event is known as a callback.

Figure 1: Front Panel of GUI

6 Conclusion

F1 value near to 1 for a particular method at some particular threshold that means it is efficient method for that particular case. If lower threshold value is selected FP will increase and as we take higher threshold value missed shot will increase. Hence select the threshold value in such a way that F1 goes near to 1. It is seen that a particular method cannot work efficiently for all type of video and at all time. Different methods behave differently for different types of videos. Depending upon type of video content and how shot is taken, whether having more similarity or not, different method behaves differently at different threshold level. Each method has optimum threshold value in the range where it works efficiently.at certain threshold value. This can be verified form the table of simulation results. Every method has certain range of threshold over which it has F1=1, means optimum result.

If we take certain range of threshold, then there are two points at which results start getting worst. This two points are the boundary of the threshold range over which F1=1. By observing every methods at this points where results start getting worst, we can decide which method is better. More specifically, χ^2 (chi-square) histogram based method gives most efficient results for almost all types of video. In sport videos there are more similarity between frames due to limited content (i.e. ground and player), so normal histogram of frames are similar hence, Histogram Intersection method is better than χ^2 (chi-square) method.

7 Acknowledgement

It is a pleasure to acknowledge the assistance received from EC department, U. V. Patel College of Engineering, in terms of the laboratory resources. The authors are also thankful to the principal, U. V. Patel College of Engineering for providing us the access of various library resources and other infrastructure facilities.

References

[1] D. Brezeal and D. J. Cook, "Automatic video classification: A survey of the literature," *IEEE transactions on systems, man and cybernetics*, no. 0, pp. 1–16, 2007.

[2] R. H. Yoo H.W. and J. D.S., "Gradual shot boundary detection using localized edge blocks," *Multimedia Tools*, vol. 28, no. 3, pp. 283–300, 2006.

[3] X. L. Z. W. L. J. L. F. Yuan J., Wang H. and Z. B., "A formal study of shot boundary detection," *IEEE Transactions Circuits Systems Video Technology*, vol. 17, no. 2, pp. 168–186, 2007.

[4] L. H. Ling L., Chao L. and Z. X., "A general method for shot boundary detection," *Proceedings International Conference Multimedia Ubiquitous Engineering*, pp. 394–398, 2008.

[5] B. A. AbdelatiMalekAmel and MtibaaAbdellatif, "Video shot boundary detection using motion activity descriptor," *Journal of Telecommunications*, vol. 2, no. 1, pp. 54–59, 2010.

[6] L. L. X. Weiming Hu, NianhuaXie and S. Maybank, "A survey on visual content-based video indexing and retrieval," *IEEE Transactions on Systems, Man and CyberneticsPart C: Applications And Reviews*, vol. 41, no. 6, pp. 797–819, 2011.

[7] A. H. O. Xu Chen and S. Savarese, "Multimodal video indexing and retrieval using directed information," *IEEE Transactions on Multimedia*, vol. 14, no. 1, pp. 3–16, 2012.

[8] H.-J. R. D.-S. J. Hun-Woo Yoo, "Gradual shot boundary detection using localized edge blocks," *Multimedia Tools Applications*, vol. 28, pp. 283–300, 2006.

[9] M. C.-S. P.-F. A. D. A. A. G. Camara-Chavez, F. Precioso, "Shot boundary detection by a hierarchical supervised approach," *IWSSIPand EC-SIPMC,Slovenia*, pp. 209–212, 2007.

[10] D. Brezeal and D. J. Cook, "Automatic video classification: A survey of the literature," *IEEE transactions on systems, man and cybernetics*, no. 0, pp. 1–16, 2007.

[11] M. A. R. Lei Zhu, JunfengQu and W. Hong, "An integrated method for video shot boundary detection," *U.S. Government Work Not Protected By U.S. Copyright*, pp. 151–154, 2010.

[12] Z. Wei, "Research on video object segmentation method," *2nd International Conference on Electronic and Mechanical Engineering and Information Technology*, pp. 52–56, 2012.

[13] U. Gargi and R. Kasturi, "An evaluation of color histogram-based methods in video indexing," *International Workshop on Image Databases and Multimedia Search*, pp. 75–82, 1996.

[14] P.-H. Zhang Zong-Ping, Liu Kun, "Shot boundary detection based on histogram of mismatching-pixel count of fmb," *ICIEA*, 2006.

[15] S. G.G.LakshmiPriya, "Video cut detection using block based histogram differences," *RGB Color Space, International Conference on Signal And Image Processing*, pp. 29–35, 2010.

[16] K. D. R. Kanagavalli, "A study on techniques used in digital video for shot segmentation and content based video retrieval," *European Journal of Scientific Research*, vol. 69, no. 3, pp. 370–380, 2012.

[17] M. A. Awais Adnan, "Shot boundary detection using sorted color histogram polynomial curve," *Life Science Journal*, vol. 10, no. 4, pp. 1965–1972, 2013.

[18] V. H. Thakore, "Video shot cut boundary detection using histogram," *International Journal of Engineering Sciences and Research Technology*, vol. 2, no. 4, pp. 872–875, 2013.

[19] Y. Tonomura and S. Abe, "Content oriented visual interface using video icons for visual database systems," *Journal of Visual Languages and Computing*, vol. 1, no. 2, pp. 183–198, 1990.

[20] A. Nagasaka and Y. Tanaka, "Automatic video indexing and full-video search for object appearances," *IFIP Working Conference on Visual Database Systems*, pp. 113–127, 1991.

[21] R. A. A. Dailianas and P. England, "Comparison of automatic video segmentation algorithms," *SPIE Conference on Integration Issues in Large Commercial Media Delivery Systems*, vol. 2615, pp. 2–16, 1995.

[22] D. B.-K. S. N. N. W. Zhao, J. Wang and W. Chang, "Improving color based video shot detection," *IEEE International Conference on Multimedia Computing and Systems*, vol. 2, pp. 752–756, 1999.

[23] C. S. D. Swanberg and R. Jain, "Knowledge guided parsing and retrieval in video databases," *SPIE Conference on Storage and Retrieval for Image and Video Databases*, vol. 1908, pp. 13–24, 1993.

[24] D. P. R. D. Mr. Sandip T. Dhagdi, "Keyframe based video summarization using automatic threshold and edge matching rate," *International Journal of Scientific and Research Publication*, vol. 2, no. 7, pp. 1–12, 2012.

[25] L. H. X. Z. Xue Ling, Li Chao, "A general method for shot boundary detection," *International Conference on Multimedia And Ubiquitous Engineering*, pp. 395–397, 2008.

[26] Z. Guang-Sheng, "A novel approach for shot boundary detection and key frames extraction," *International Conference on Multimedia And Information Technology*, pp. 221–224, 2008.

[27] M. W. Y. C. Feng Hong-Cai, Yuan Xiao-Juan, "A shot boundary detection method based on color space," *International Conference on E-Business and E-Government*, pp. 1647–1650, 2010.

[28] J. C. ShujuanShen, "Abrupt shot boundary detection algorithm based on fuzzy clustering neural network," *IEEE transactions*, pp. 46–48, 2011.

[29] H. B. Ma Chunmei, Dong Changyan, "A new method for shot boundary detection," *International Conference on Industrial Control and Electronics Engineering*, pp. 156–160, 2012.

[30] T. G. A. Lus Varela, Rui P. Amaral, "Shot segmentation algorithm for online video event detection," *INES 2012, IEEE 16th International Conference on Intelligent Engineering Systems*, pp. 213–216, 2012.