Automatic Video Parsing
Using Shot Boundary Detection and Camera Operation Analysis

Mee-Sook Lee, Bon-Woo Hwang, Sanghoon Sull and Seong-Whan Lee
Center for Artificial Vision Research, Korea University
Anam-dong, Seongbuk-ku, Seoul 136-701, Korea
{mslee, bwhwang, sull, swlee}@image.korea.ac.kr

Abstract

In this paper, we present an efficient video parsing method using shot boundary detection and camera operation analysis technique. In the shot boundary detection, the local color information and an adaptive time window is used. The local spatio-temporal images and MLP(multilayer perceptron) are used for analyzing the camera operations.

In order to verify the performance of the proposed video parsing method, experiments with video database have been carried out. Experimental results demonstrate the efficiency of the video parsing technique.

1. Introduction

Content-based video indexing and retrieval are methods that find and manage the essential information of video[1, 2].

The ability of automatic video parsing is necessary for content-based video indexing and retrieval. Video parsing involves two tasks: video segmentation and video indexing. The video stream is segmented into the elemental units such as shots and scenes at the video segmentation stage. And each elemental unit is labeled based on its contents at video indexing stage. Shot is the basic unit for video manipulation, and there are many different transitions between shots: cuts, fades, dissolves, wipes, etc.

In this paper, We propose an efficient video parsing method. For the shot boundary detection, the local color information is used. In order to reduce the computation time, an adaptive time window is used. The local spatio-temporal images and MLP are used for analyzing camera operations. This method is reliable and fast because it utilizes the learning algorithm with spatial-temporal information in frames and does not process the entire image.

The remainder of this paper is organized as follows. Sections 2 and 3 describe the proposed shot boundary detection and the camera operation analysis method. Section 4 describes the experimental results and Section 5 concludes this paper.

2. Shot Boundary Detection

The histogram comparison is the most common method for detecting shot boundaries from raw video data. This method is quite simple, but ignores spatial information of a frame. And the error rate caused by abrupt luminance change is very high. In this paper, the mean values of color in local sub-blocks based on the Net comparison method[3] are used. This feature contains the spatial information of a frame and is robust when the abrupt luminance change occurs.

2.1. Local color comparison

Assume that each pixel in an image has same probability to change. The similarity S is defined as follow:

$$S(n, n+k) = 1 - \frac{N_C(n, n+k)}{N} \tag{1}$$

where N is the total number of sub-blocks in one frame, and N_C is the number of sub-blocks that are changed between two frames.

$$N_C(n, n+k) = \sum_{i=0}^{N-1} \varphi_{n,n+k}(i) \tag{2}$$

In equation 2, φ is the function defined as follow:

$$\varphi_{n,n+k}(i) = \begin{cases} 1 & if\ D^H_{n,n+k} > T_{sub} \\ & or\ D^S_{n,n+k} > T_{sub} \\ 0 & otherwise \end{cases} \tag{3}$$

where D^H and D^S represent the difference of mean value of H and S, between two sub-blocks, respectively.

$$D^H_{n,n+k}(i) = |E_H(n,i) - E_H(n+k,i)| \tag{4}$$

$$D^S_{n,n+k}(i) = |E_S(n,i) - E_S(n+k,i)| \tag{5}$$

where, E_H and E_S is the average value of H and S in each sub-block.

Table 1. Algorithm for shot boundary detection

1. Determine the number and the position of sub-blocks.
2. Compute the mean value of H and S at each sub-block.
3. If one of the D^H and D^S is greater than T_{sub}, determine that this sub-block has changed.
4. If the similarity S is not greater than the T_{frame} a shot boundary is declared.

2.2. Adaptive time window

It is time-consuming to apply shot boundary detection method to every frame, because almost all video shots are longer than dozens of frames. In order to reduce the computation time, we use the adaptive time sliding window whose size can be changed flexibly.

Table 2. Algorithm for adaptive time window

```
WindowSize = MaxWindowSize;
for(i=0;  i< TotalNumOfFrames;  i++){
    NextFrame = CurrentFrame + WindowSize;
    if(IsSimilar(CurrentFrame, NextFrame)){
        CurrentFrameNo = NextFrameNo;}
    else {
        if(WindowSize == 1){
            CurrentFrameNo = NextFrameNo;
            WindowSize = MaxWindowSize;
        }else {
            WindowSize = WindowSize/ScaleFactor;} }}
```

3. Camera Operation Analysis

In this section, we describe a method for analyzing camera operations using 2D spatio-temporal (2DST) images and multilayer perceptron. First, the 2DST images are generated for a given period. Then, the 2D Discrete Fast Fourier Transform and the power spectrums are applied in order to analyze the texture of the 2DST images. Finally, multilayer perceptron is used to analyze the camera operation.

3.1. Spatio-temporal image

2DST image is formed by stacking up the corresponding segments in time order. They have a special texture if pixels moved in the same direction. So, camera operations can be extracted by analyzing the texture of 2DST images. The 2D Discrete Fast Fourier Transform(2DFFT) and the power spectrum are used to analyze the textures in 2DST images. The direction θ_k of the 2DST image in each segment can be calculated from the equation 6

$$p(\theta_k) > p(\theta_i) \quad (i \neq k, \quad i,k = 0,...,\pi) \qquad (6)$$

where $p(\theta)$ is defined as $p(\theta) = \sum_r P(r,\theta)$. $P(r,\theta)$ is the power spectrum in polar coordinates form.

3.2. Camera operation analysis using multilayer perceptron

In this paper, two MLPs are used. One is for horizontal direction, and the other is for vertical direction. The architecture of two MLPs used to analyze the camera operations is same. This MLP has three layers. The input layer has 5x5 nodes and gets the direction extracted by analyzing each 2DST image as input data. The hidden layer has 10 nodes to maximize the performance of the MLP. The output layer has 6 nodes corresponding each camera operation, for example, in case of horizontal direction, each node represents the still, right, left, zoom in and out or ambiguous case. The data used for training are created by adding the random noises into the standard data. The example of standard training data for horizontal MLP are given in Figure 1.

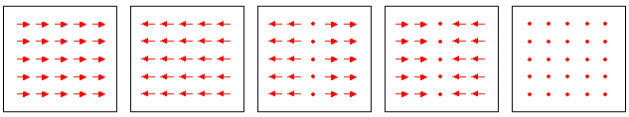

Figure 1. The standard training data

4. Experimental Results

The category, size and components of each video used in the experiments are given in Table 3. All video data were digitized at the size of 320x240 pixels at 15fps(frames per second).

Table 3. video data used for experiments

video category	video size (minutes)	# of shot boundary	# of camera operations
news	20	218	19
documentary	10.35	92	31
movie	10.48	71	17

4.1. Results of shot boundary detection

We compared the proposed method to the previous methods. The selected previous methods are pixel-wise comparison(PWC), gray-level histogram comparison(GHC), local block gray-level histogram comparison(LGHC), histogram comparison of difference image(HCOD) and edge image comparison(EIC)[4].

- $PWC(n, n+k) = \frac{\sum_{i=0}^{N-1} \Phi_{n,n+k}(i)}{N}$

Φ has 1, if the difference of gray value between two frame exceeds the threshold, otherwise has 0.

- $GHC(n, n+k) = \sum_{i=0}^{Q-1} |G_n(i) - G_{n+k}(i)|$

where $G_n(i)$ is the ith gray-level histogram value of nth frame, and Q is the size of histogram bin.

- $LGHC(n, n+k) =$

$$\sum_{b=0}^{N_b-1} \sum_{i=0}^{Q-1} |G_n(b,i) - G_{n+k}(b,i)|$$

where N_b is the number of local blocks in a frame, and $G_n(b,i)$ is the ith gray-level histogram value of b th local block in nth frame.

- $HCOD(n, n+k) = \frac{\sum_{i \notin [-T_{pix}, T_{pix}]} hod(i)}{N}$

where $hod(i) = G(f_n - f_{n+k}, i)$, f_n is the nth frame.

- $EIC(n, n+k) = max(\rho_{in}, \rho_{out})$

where ρ_{in} and ρ_{out} are the percentage of pixels that are newly created and disappeared, respectively.

Table 4 shows the comparison results of shot boundary detection. In Table 4, N_C, N_{FN} and N_{FP} represent the number of frames correctly detected, the number of false negatives and the number of false positives, respectively. And in X/Y/Z representation, X, Y and Z denote the result of news, documentary and movie, respectively.

Table 4. Comparison results

method	N_C	N_{FN}	N_{FP}
PWC	164/82/53	54/10/18	344/103/ 150
GHC	194/78/61	24/14/10	135/89/153
LGHC	202/87/63	16/5/8	303/104/128
HCOD	190/85/62	28/7/9	135/89/ 97
EIC	208/88/65	10/4/6	104/67/116
Proposed	202/87/63	16/5/8	181/89/111

According to the Table 4 , the EIC, LGHC and the proposed method show good performance.

Table 5. computing time(frames/sec)

method	video categories		
	news	documentary	movie
PWC	11.2	10.7	11.5
GHC	6.0	6.6	6.4
LGHC	7.2	6.9	7.0
HCOD	8.4	8.6	8.4
EIC	3.5	3.0	3.5
Proposed	10.8	10.2	10.5

Table 5 shows the computing time of each methods. The PWC and the proposed method is faster than others. The EIC shows the worst performance, because it needs very complex procedures. The PWC and the GHC can be recommended for applications where the ratio of false positive is not important. Therefore, the proposed method seems to be the best algorithm in term of the performance and computing time.

4.2. Results of camera operation analysis

We classify the basic camera operations as three classes, horizontal movement, vertical movement and zoom in and out, because it is difficult to discriminate correctly in video.

Table 6 shows the results of camera operation analysis of the proposed method(CO denotes the camera operation). The composition type represents the data that consist of two or more simple camera operations such as panning with zooming in, tilting with zooming out, and panning with tilting and zooming in, etc.

Table 6. Results of the analysis

type of CO	# of CO	N_C	N_{FN}	N_{FP}
horizontal	25	19	6	75
vertical	9	6	3	0
zoom	10	7	3	21
composition	23	11	12	57

The documentary video has been used for this experiments. The proposed method detected correctly the simple camera operations, but it showed poor performance in composition type. In the case of horizontal movement, the error rate caused by moving objects is high, so the number of false positives is large.

Acknowledgement

This research was supported by the Systems Engineering Research Institute under the contract: 97-19-P00241.

References

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

[2] P. Aigrain, H. J. Zhang and D. Petkovic, "Content-Based Representation and Retrieval of Visual Media: A State-of-the-Art Review," *Multimedia Tools and Applications,* Vol. 3, pp. 179-202, 1996.

[3] W. Xiong, J. C. M. Lee and M.-C. Ip, "Net comparison: a fast and effective method for classifying image sequences," *Proc. of SPIE - Storage and Retrieval for Image and Video Databases II,* San Jose, CA, Vol. 2420, pp. 318-328, 1995.

[4] Y. Wu and D. Suter, "A Comparison of Methods for Scene Change Detection in Noisy Image Sequence," *Proc. of the First International Conference on Visual Information Systems,* Melbourne, Australia, pp. 459 - 468, 1996.