Scene break detection and classification using a block-wise difference method

Mehran Yazdi and André Zaccarin
Department of Electrical and Computer Engineering
Laval University, Ste-Foy, Prov. Québec, Canada
Emails: {yazdi,zaccarin}@gel.ulaval.ca

Abstract

We introduce a new approach to the detection and the classification of effects in video sequences. We deal with the effects such as cuts, fades, dissolves, and camera motions. A global motion compensation based on block matching and a measure of block mean intensities are used to detect all the effects. We determine the dominant motion vectors to detect the camera motions for each shot and observe the ratio of the difference intensity variation of blocks in consecutive frames to detect change effects between video shots. The approach can handle the complex motions during gradual effects as well as the precisely detection of effect lengths. Both synthetic and real evidences are presented to demonstrate how this approach can efficiently classify effects in video sequences involving significant motions.

1: Introduction

The developments in visual information technology have enabled users to view large amount of video data over multimedia mediums like the internet. However, tools for facilitating search and retrieval of these video data are still limited.

A video sequence is a temporally evolving medium, where its content changes due to object or camera motion, cuts, and special effects. Video segmentation, which is the first step in content-based video analysis and retrieval, refers to breaking the input video into temporal segments with common characteristics. Manually partitioning an input video is an inefficient and inadequate process. Therefore, visual content descriptors that facilitate automatic annotation of input video need to be used.

Content based temporal video segmentation is mostly achieved by camera shot detections, where each shot refers to a sequence of frames generated during a single operation by the camera. Some effects are used to describe when a transition occurs from one shot to another. The *cut* is generally used to connect continuous activities. *Fades* have the effect establishing of the boundaries of individual scenes or sequences. A *fade out* corresponds to the endpoint of a scene, while the *fade in* signals the start of the new scene. A *dissolve* may be used to carry out the passage of the time or to bridge events occurring at separate times or places. Camera motion effects such as *zoom*, *pan*, and *tilt* are used to more efficiently capture the visual features of the scene during a long shot. So these effects can be used to classify the video shots.

Automatic Detection of the effects of a video sequence is a low-level feature description and is important toward firstly local feature extraction and finally the semantic description of the scene. To achieve this, many problems must be considered. It is difficult to detect all abrupt changes, gradual changes, and camera motions at the same time. Handling of more complex camera motions that occur in dissolves is also another problem. Any scene change detections should not be sensitive to the length of gradual changes. We present in this work an approach that can handle these problems.

Firstly, we discuss the related works on video effect detection. Then, we present our approach which is based on the analysis of vector motions resulting of block matching algorithm and overall mean intensity variations between blocks. We show how our approach can detect the effects on a simulated sequence even in present of complex camera motions and the different lengths of gradual changes. We present finally the current results on real video sequences and we discuss the limitation involving our algorithm.

2: Related works

The major techniques that have been used for scene breaks detections can be reviewed as intensity histogram comparison [3], color histogram differences [2], local feature based analysis [1], and compression factor for difference comparison [4]. These algorithms detect most every sharp transition, but they have more difficulties to correctly detect gradual transitions.

Some papers have proposed the transition models for detecting gradual effects. [5] has proposed an algorithm which first extracts the intervals of such effects by finding a local minimum in the histogram formed by the number of edge pixels between consecutive frames and then classifies them by the intensity model effect of transition. Their approach is extremely depended of correctly detecting the beginning and end frames of gradual transitions. The motion is also a major limitation of their algorithm.

Some improvements have been proposed by

extending classical histogram based approaches to handle dissolve [7]. They use a dual threshold and a motion compensation techniques to detect dissolve. Their method poorly handle false detection between scenes involving complex motion and can not correctly detect effect lengths.

A different alternative is proposed by [6], where an algorithm is presented which is based on computing the number of edge pixels that appear and disappear in a given frame as a feature for the comparison and detection of effects. Their algorithm is based on that during a cut or a dissolve, new intensity edges appear far from the locations of old edges. By counting the entering and exiting edge pixels, they detect and classify effects. The main problem of this algorithm is dealing with affine transformations of the contents of the image at low computational cost. Besides, in busy scenes, the separation of entering edges pixels from exiting edge pixels is not an easy task to do. Extraction of edges in the beginning and end frames of a long gradual transition where the pixel intensity values of appear or disappear scenes are so small, is also their another limitation.

Generally, an abrupt cut may be detected through a suitable metric which exhibits a comparably abrupt change in its value, whereas detecting special effects tends to require examining accumulate differences in the value of such a metric. However, simply detecting differences in successive video frames may result in many false breaks being detected. The false breaks are caused by large movements of objects within a shot or excessive camera movements while taking that shot, such as zooming, panning, and tilting. Thus, it is crucial to detect these changes and to distinguish them from special effects.

3: Motion compensation and classification

First, we compute a global motion of the scene between two consecutive frames before effect detection. There are the vast different algorithms reported in literature. We search for a translational motion between two frames. The variation of a large region is more significant than a smaller one, so we use a block matching algorithm for camera motion compensation which is sufficient and robust. While it is possible to handle affine or projective motions, they don't give additional information and don't necessarily perform better.

Each frame is first divided into blocks and we compute the motion vectors of each block by searching the best similar block in the search block around this block. The similarity is defined as the minimum of mean square error. Then, the motion vectors are quantified into four regions correspond to camera displacement directions and three regions correspond to camera speeds (Fig. 1-a).

Dominant motion vectors in one region, determine the type of motion. We also divide motion vectors into positive and negative types which correspond to the direction of motion vectors toward inside the frame and outside the frame respectively (Fig. 1-b).

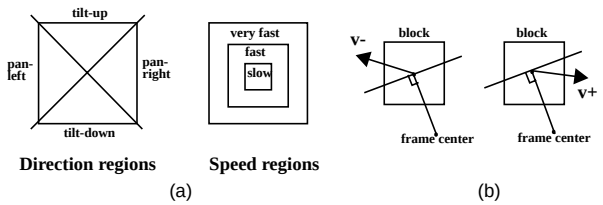

Fig. 1. a) Direction and speed vector quantification. b) Inside-outside vector quantification.

Dominant motion vectors in one type determine *zoom in* or *zoom out* effect. The algorithm is robust even in present of small camera movements. We use the original widely-known table tennis sequence which contains a fair amount of motion including zoom and pan plus a few cuts. Fig. 3 shows the results of the algorithm. On this sequence the motion shots are correctly detected and classified.

4: Detecting and classifying scene breaks

Once we know the global motion of the scene, we can distinguish motion from shot changes by computing a measure of overall intensity variations during shot changes. Conceptually, cuts are caused by sudden change of pixel intensities, while during a dissolve or a fade, pixel intensities tend to linearly increase or decrease. This leads us to distinguished dissolve and fades from other abnormal changes by using the computation of linearity between intensity variation of pixels. The measure of linear intensity differences of pixels give the exact detection of a dissolve and a fade. However, for a dissolve this measure is not reliable in present of motion or any variation of original scene intensities.

Meanwhile, although the concept of linearity does not hold for pixels intensities, the overall mean intensity of most blocks tend to maintain the regular (even linear) increasing or decreasing manner during all bunch of frames of a dissolve. Fig. 2 shows an example of a dissolve created by simulating two motion shots of tennis table sequence.

As we can see, even in present of motion, the overall intensity of two blocks tends to continuously decrease or increase. So based on this observation, we compute a measure of regular increasing and decreasing block intensities by using the number of blocks in which their mean intensity variation holds in increasing or decreasing manner during three consecutive frames. We need to verify at least three consecutive blocks to be sure that the regularity of increase or decrease variations holds. The measure is computed as:

$$R_{dissolve/fade} = \frac{B_v}{B_t} \quad (1)$$

B_v is the number of blocks which yield the similar

variation of mean intensity and B_t is the total number of blocks. This measure lets us detect any major regular changes during a bunch of frames.

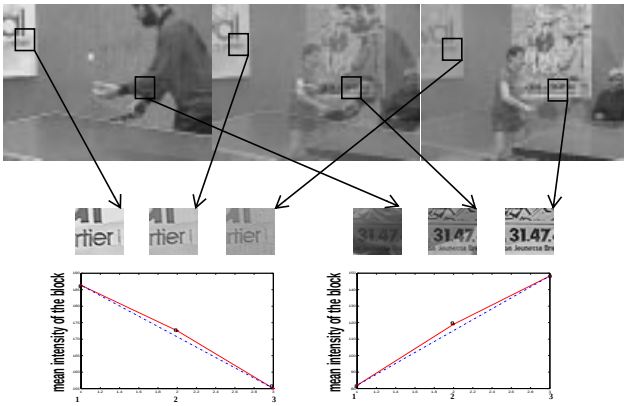

Fig. 2. Regular increase and decrease of the mean intensity of blocks during a dissolve.

Dissolve and fade cause the same effect in $R_{dissolve/fade}$. Typically, during a fade, images have their intensities linearly change from 0 to1 or from 1 to 0. So the examination of zero point of a scene break lets us separate the fade from dissolve and even detect fade-in and fade-out. It is self-evident that the cuts don't affect on this measure. Cut is characterized by sudden change of scene and thus we use a measure of number of blocks in which the mean intensity changes dramatically.

$$R_{cut} = \frac{B_s}{B_t} \qquad (2)$$

B_s is the number of blocks which yield the sudden variation of mean intensity and B_t is the total number of blocks. For additional efficiency, we have normalized $R_{dissolve/fade}$ and R_{fade} to have the values between 0 and 1.

Our algorithm is based on the measure of the number of blocks and their mean intensity instead of the value of pixel intensities them-selves. So the algorithm is less sensitive to any little variation of intensity values but it is robust to overall regular changes. Fig. 4 shows an example of scene break detection and classification. We use a simulated sequence by modifying table tennis sequence. We have added cuts and also dissolve and fades during where we use some mixture of motions in different lengths. On this sequence, $R_{dissolve/fade}$ and R_{cut} show clear peaks at the scene breaks and the algorithm correctly classifies them as well as precisely detects the effect sizes.

5: Experimental results

We tested our algorithm on various video sequences. Fig. 5 and Fig. 7 show two cases of scene break detection for two different types; Seinfeld sequence which is a busy scene with the complex motions and a news program which has many text changes. The images from dissolve are shown in Fig. 6-8. The algorithm detects the scene break effects in presence of motion as well as their length.

Our algorithm detects only the gradual changes which occur more than two frames which it is the general case in practice. Some false positives have been reported in present of rapid camera operation or significant object motion. This is mostly because that block matching algorithm can not correctly detect the motion. Speed motion detection of shots which is computed during motion compensation (Section 3), gives a good hint about if the motion compensation of block matching is reliable or not. Actually, during a rapid motion, the vector motions tend to have the values more than the maximum size of search blocks. Then, the dominant vectors are classified as very fast while this is not true for a cut where there is not a dominant vector. Once we know that, we can extended the search space for blocks to correctly detect motion or simply exclude these false positives from the scene break detection algorithm.

6: Conclusion and future works

In this paper, we presented a complete framework for scene breaks and shot motion detection. We have developed a scene break approach and an algorithm for classifying camera motions. Our algorithm can successfully classify motions. It detects and classifies scene breaks in the present of motion as well as their size. We are integrating this approach into an interface for video sequences which facilitates the user to search for scene breaks and motions. Our current research efforts deal with using this framework for a complete video segmentation by semantic object tracking.

7: References

[1] G. Pass and R. Zabih, "Comparing images using joint histograms", *ACM Journal of Multimedia Systems*, 1998.

[2] M. J. Swain and H. Ballard, "Color indexing", *International Journal of Computer Vision*, vol. 7(1), pp. 11-32, 1991.

[3] Y. Tonomura, K. Otsuji, A. akutsu, and Y. Ohba, "Stored video handling techniques", *NTT Review*, vol. 5, pp. 82-60, Mar. 1993.

[4] B. Yeo and B. Liu, "Rapid scene analysis on compressed video," *IEEE Trans. Circuits and Syst. for Video Tech.*, 1995.

[5] H. Yu, G. Bozdagi, and S. Harrington, "Feature-based hierarchical video segmentation", *IEEE International Conference on Image Processing*, vol. 2, pp. 498-501, Aug. 1997.

[6] R. Zabih, J. Miller, and K. Mai, "A feature-based algorithm for detecting and classifying production effects", *Multimedia Systems*, vol. 7, pp. 119-128, 1999.

[7] H. Zhang, A. Kankanhalli, and S. Smoliar, " Automatic partitioning of full-motion video" M*ultimedia Systems*, vol. 1, pp. 10-28, 1993.

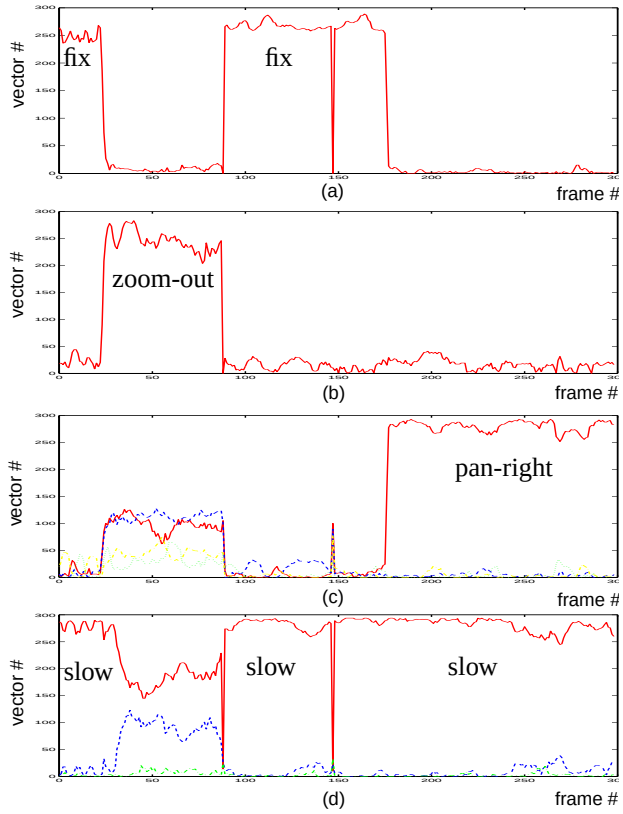

Fig. 3. Motion detection of the table tennis sequence. a) Histogram of dominant vectors with nul values. b) Histogram of positive-negative dominant vectors. c) Histogram of pan-right, pan-left, tilt-up, and tilt-down dominant vectors. d) Histogram of the speed of dominant vectors. We used a 16x16 block matching algorithm with 32x32 search blocks.

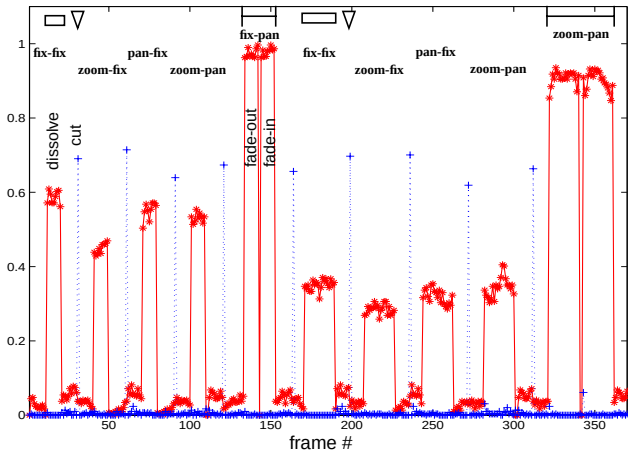

Fig. 4. An example of cut, dissolve, and fade detection for the simulated table tennis sequence in present of complex motions and different effect sizes. We used a 16x16 block matching algorithm with 32x32 search blocks.

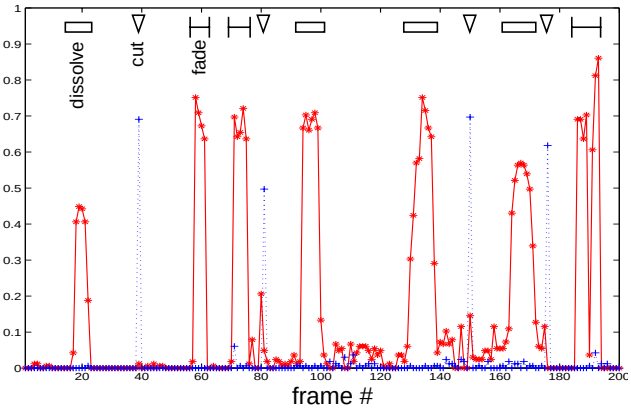

Fig. 5. An example of cut, dissolve, and fade measures for Seinfeld sequence. Frames are 96x128. We used a 8x8 block matching algorithm with 16x16 search blocks.

Fig. 6. Images from Seinfeld sequence detected as dissolves.

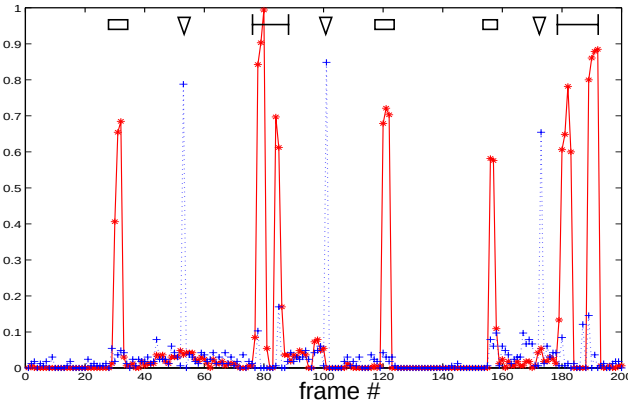

Fig. 5. An example of cut, dissolve, and fade measures for a news program. Frames are 96x128. We used a 8x8 block matching algorithm with 16x16 search blocks.

Fig. 8. Images from a news program detected as dissolves.