INFORMATICA, 2003, Vol. 14, No. 1, 19–36

Scene Change Detection Schemes for Video Indexing in Uncompressed Domain

Eyas EL-QAWASMEH
Computer Science and Information Systems Dept., Jordan University of Science and Technology
P.O. Box 3030, Irbid 22110, Jordan
e-mail: eyas@just.edu.jo

Received: December 2002

Abstract. There are a growing number of applications, which extensively use the visual media. A key requirement in those applications is efficient access to the stored visual information for the purposes of indexing, fast retrieval, and scene analysis. To support this vision, many scene change detection algorithms have been developed for both sudden and gradual scene change in uncompressed domain. This paper presents a comparison of shot boundary detection and classification techniques and their variations including histogram, pixel-differences, likelihood ratio, and motion vector. In addition, the pros and cons of each approach with many clarifying comments are presented.

Key words: audio segmentation, scene change detection, shot change detection, video databases, video indexing.

1. Introduction

Recently, multimedia information has been made overwhelmingly accessible with the rapid advances in communication and multimedia computing technologies. The requirements for efficiently accessing mass amounts of multimedia data are becoming more and more important (Chen *et al.*, 2001). The video scene change detection is a fundamental operation used in digital video transition and storage technologies (including video-on-demand services, multimedia-capable computer networks, and digital libraries), that must be performed prior all other processes.

One of the challenges of this new technology is video indexing that organizes all this footage, so that video clips relevant to the needs of the users can be retrieved efficiently. An effective video indexing system must be able to do indexing automatically in reasonable time, and scale to support terabytes of video information. It must support a wide range of queries based on audio and speech content, image content, action and camera motion. Finally, it must allow users to retrieve and browse individual video clips, frames, and shots.

Video indexing requires the parse process for video clips. This process will extract meaningful features of the video. An important operation in this process is the scene and shot segmentation. This is because shots are the elementary units, which constitute

a video. The detection of shot boundaries allow the recovering elementary video units, which in turn provide the ground for nearly all existing video abstraction and high-level video segmentation algorithms. Most video' programs are created by editing together a series of shots. The edits may consist of abrupt cuts or more subtle transitions. These edits must be detected in order to break down the video into chunks suitable for presentation, and to reveal its logical structure.

Currently there are many image statistics and multi-time scale comparison algorithms to detect and distinguish between cuts and dissolves, and to reject transients such as flashbulbs, and continuous motion. These algorithms are designed to work either in the compressed domain, which impose some restrictions, or uncompressed-domain. In this paper, we are interested in an uncompressed-domain only.

Using image-processing techniques, detailed image analysis algorithms are too slow to be applied to every frame of a video sequence. Thus, a relatively small number of key frames must be extracted from each shot for further processing (key frames can also be used as a storyboard to give users an overview of the video content).

The choice of the number and position of key frames is of great significance to the speed and accuracy of a video indexing system; thus, we are investigating algorithms for shot break detection. These shots are the basic units that are used to extract key frames for further processing in image indexing. This has a great significance on the speed and accuracy of any video indexing system.

In this paper, a comprehensive survey will be conducted on these techniques. In addition, the pros and cons of each one will be explained. The organization of this paper will be as follows. Section 2 is video structure. Section 3 is shot detection. Section 4 is the sudden cut detection. Section 5 is gradual effect detection techniques. Section 6 is discussion, and Section 7 is the conclusions.

2. Video Structure

Video is a rich source of information. It provides visual information about scenes. However, this information is implicitly buried inside the raw video data, and is provided with the cost of very high temporal redundancy. While the standard sequential form of video storage is adequate for viewing in a "movie mode", it fails to support rapid access to information of interest that is required in many of the emerging applications of video (Irani *et al.*, 1998). In this context, video indexing plays a major role in organization of this information for fast access and retrieval of videos.

One of the most popular approaches to represent video contents is the structure modeling approach. In this structure, the video sequences are first divided into shots, where each *shot* is a continuous sequence of video frames that have no significant inter-frame difference in terms of their visual contents. A *shot* is obtained by a continuous camera recording. Thus, a movie sequence that alternated between views of two people would consist of multiple shots (Boreczky, 1996). A shot usually represents a primitive concept as in hypertext and is used as a basic unit of manipulation. The discontinuous gap between shots is called a *cut*. A *scene* is defined as a collection of one or more adjoining

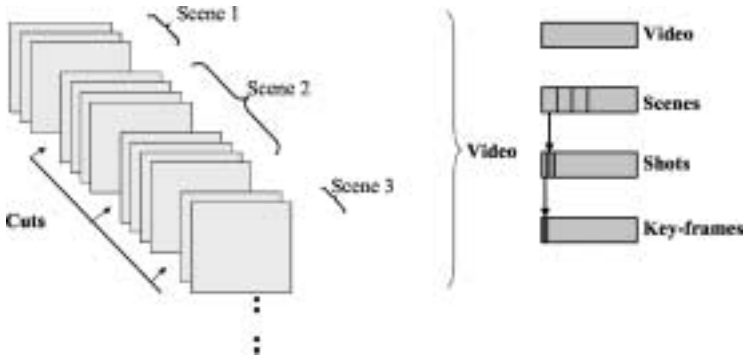

Fig. 1. Structure model of a video.

shots that focus on an object or objects of interests. For example, a person walking down a hallway into a room would be one scene, even though different camera angles might be shown. Three camera shots showing three different people walking down a hallway might be one scene if the important object was the hallway and not the people. Fig. 1 describes the structural model and the terms that we use in video indexing.

The most common approach to video segmentation is shot transition detection where each shot represents a meaningful event or a continuous sequence of action. Once shots have been identified, key frames of each shot must be selected. Several techniques have been proposed for key frames selection. Then, when the storyboard has been created, the next step is to index each still image (key image). Image indexing and retrieval methods based on automatically derived features such as color, texture and shape starts to be available in the marketplace.

3. Shot Detection

Shot boundaries are caused by a change of camera position. This causes a different view on the scene. To determine shot boundaries one has to measure the change in visual contents. The shot detection work in general as follows: 1) it extracts features from frames; 2) it computes the difference for consecutive frames; 3) it compares the difference to a threshold. Fig. 2 shows a shot detection.

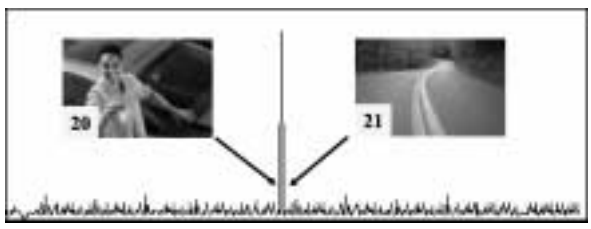

Fig. 2. Shot detection.

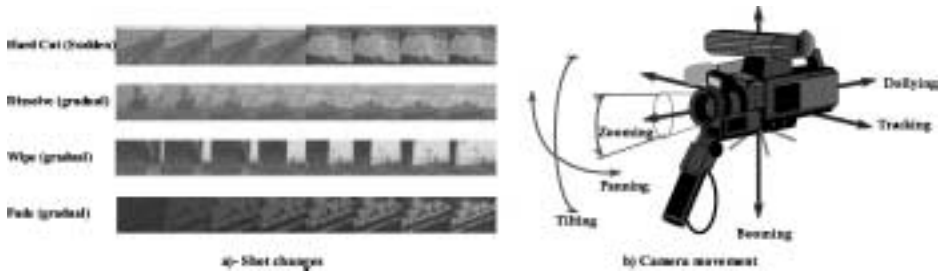

Fig. 3. Examples of shot changes and camera movement.

Shot transitions can be divided into two categories: abrupt transitions and gradual transitions. Gradual transitions include camera movement (panning, tilting, zooming), and video editing special effects (fade, dissolving, wiping). Segmentation into shots is the first phase in indexing video databases. Fig. 3 shows an example of gradual transition. In general, hard cut is the easiest to detect among other kinds. More details of each one will be addressed later.

Automatic shot boundary detection is a difficult problem since any kind of shot transition can be easily confused with camera and object motion, which occur in video anyway. A shot with much object motion throughout the frame such as a sports or action shot or a clip from a music video, can cause the false recognition of a shot boundary. To further complicate matters, a camera can have a variety of movements such as panning, tilting, tracking, zooming in or out, or a combination of these (Browne *et al.*, 2000).

Boundary detection techniques can be grouped into two categories. The first category consists of cut detection methods that use video in uncompressed form. The second category consists of methods that directly detect cuts in compressed domain. The major techniques that have been used for the first category are pixel-differences, statistical differences, histogram comparisons, edge differences, and motion vectors. Early techniques of automatic content extraction of videos focused on cut detection and more recent work has focused on detecting gradual transitions.

4. Sudden (Abrupt) Detection

Abrupt shot transitions, which are also known as hard cuts, are very easy and common to detect since the two frames that we compare are unrelated. A hard cut occurs in a single frame, thus, if a frame contains two consecutive shots I_1 and I_2 then a similarity measure computed in the followimg equation

$$D(I_1, I_2) = \sum_{i=1}^{X} \sum_{i=1}^{Y} |I_1(i,j) - I_2(i,j)| \tag{1}$$

is used to determine the hard cut by comparing it against a certain threshold.

Hard cuts are reliable and easy to detect. A hard cut generates a temporal visual discontinuity in the video stream presented by isolated peaks in the feature time series. Current hard cut detection algorithms are different in the features used to measure the discontinuity and in the schemes used to detect the discontinuity (Lienhart, 2001). For example, some use shot intensity function and pixel spatial distributions (i.e., the position of a pixel as well as its neighbors) for their features.

Most previous work on detecting a sudden scene change is based on the entire images and uses different metrics to evaluate the changes between successive frames. The most common schemes are:

4.1. *Pixel-Differences*

In this technique, the absolute intensity difference between two corresponding pixels of consecutive frames is computed. This number is compared against a threshold to determine if a shot boundary has been found (see Fig. 4).

This method is sensitive to camera motion, object movement, and changes in the background. In this technique, it is possible to produce false alarms due to fast camera and/or object movement. The high cost of pixel-differences and the sensitivity to camera operation like zooming makes it a very robust approach (Aslandogan *et al.*, 1999).

Zhange, Kankanhalli, and Smoliar implemented this method with the additional step of using a 3×3 averaging filter before the comparison to reduce camera motion and noise effects (Zhang, 1993). They obtained good results by selecting a threshold tailored to the input sequence, although the method was somewhat slow. It was reported that adjusting the threshold manually is unlikely to be practical (Boreczky, 1996).

Shahraray divided the images into 12 regions, and found the best match for each region in a neighborhood around the region in the other image. This matching process duplicates the process used to extract motion vectors from an image pair. The pixel-differences for each region were sorted, and the weighted sum of the sorted region differences provides the image differences measure. Gradual transitions were detected by generating a cumulative difference measure from consecutive values of the image differences (Shahraray, 1995).

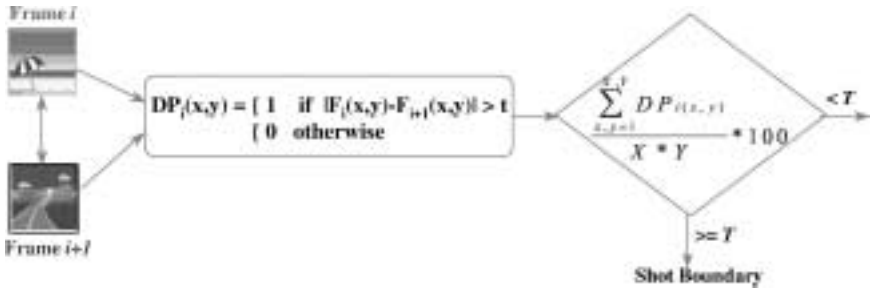

Fig. 4. Pixel-differences approach.

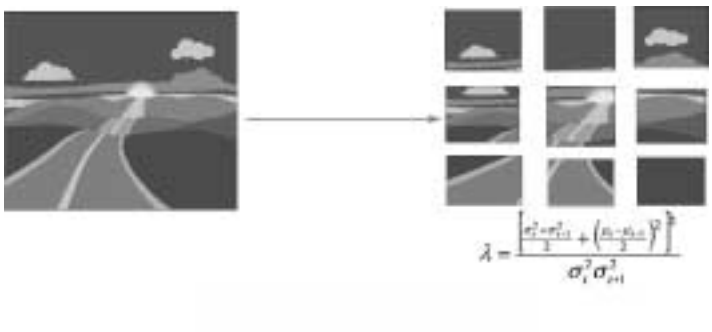

Fig. 5. Likelihood ratio.

4.2. *Likelihood Ratio*

This algorithm works on the level of blocks of pixels instead of pixels. In this algorithm, the frames are subdivided into a set of blocks, which are then compared on the basis of the statistical characteristics of their intensity levels. The formula that calculates the likelihood function λ is listed in Fig. 5.

In Fig. 5, μ_i and μ_{i+1} are the mean intensity values for a given region in two consecutive frames and σ_i^2 and σ_{i+1}^2 are the corresponding variances. The camera break is assumed when a sufficient fraction of blocks satisfy $\lambda > t$ where t represents an appropriate threshold (Zhang, 1993). A subset of the blocks can be used to detect the difference between the images so as to expedite the process of block matching. Note that it is possible for two different blocks to have the same density function, and in such a case, no change will be detected.

This approach is better than the pixel-differences approach as it increases the tolerance to noise associated with camera and object movement (Fernando, 2001). On the other hand, its disadvantage is the high computational complexity it needs.

The sensitivity to camera and object motion can also be reduced by comparing the histograms of the two frames. This is because any two frames whose backgrounds differ slightly and have the same amount of object motion will have almost the same histograms. This is explained in the next sub-section.

4.3. *Histogram Comparison*

The basic idea in the histogram is that the color content does not change rapidly within shots. Thus, hard cuts and other short lasting transitions can be detected as single peaks in the time series of the differences between color histogram of contiguous frames or of frames at certain distance k apart (Lienhart, 1999).

The histogram is given by the number of pixels belongs to each level in the frame. The simplest histogram method computes gray level or color histograms of the two images. If the bin-wise difference between the two histograms is above a threshold, a shot boundary is assumed (see Fig. 6).

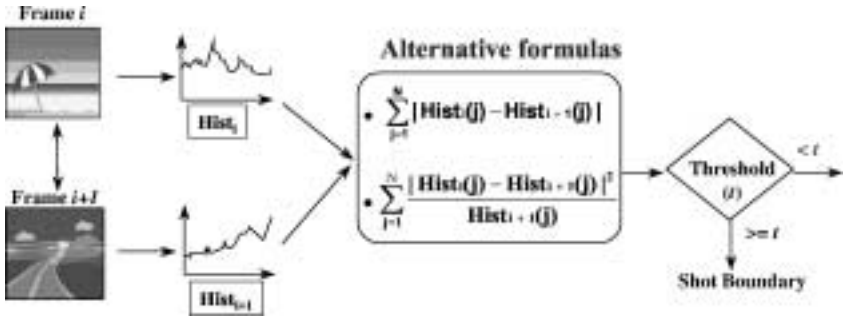

Fig. 6. Histogram technique.

Histograms are the most common and reliable method used to detect shot boundaries (Fernando *et al.*, 2001). Histograms are less sensitive to object motion than other metrics (Zhang *et al.*, 2001). To be more specific, they are more sensitive to global changes, such as panning and zooming than to local changes due to object motion (Lee *et al.*, 2000). The less sensitivity to object motion is due to the fact that they ignore the spatial changes in a frame.

Nagasaka and Tanaka compared several simple statistics based on gray level and color histograms. They found the best results by breaking the images into 16 regions, using a χ^2-test on color histograms of these regions, and discarding the 8 largest differences to reduce the effects of object motion and noise (Boreczky, 1996). The χ^2-test they use for the area between the distributions is given in Eq. 2.

$$\sum_{t}^{N} \frac{\left(H_{t+1}\left(j\right) - H_i\left(j\right)\right)^2}{H_{t+1}\left(j\right)}, \tag{2}$$

where N is the number of gray levels, j is the gray/color value, t is the frame at time t, and $H_t(j)$ is the value of the histogram for the gray/color level at time t.

Swanberg, Shu, and Jain used gray/color level histogram differences in regions; weighted by how likely the region was to change in the video sequence. This worked well because their test video (CNN Headline News) has a very regular spatial structure. They did some simple shot categorization by comparing shots with the known types (e.g., anchor person shot) in a database. They were also able to group shots into higher-level objects such as scenes and segments by matching the shot types with the known temporal structural (Boreczky, 1996; Swanberg *et al.*, 1993).

Zhange, Kankanhalli, and Smoliar compared pixel-differences, statistical differences and several different histogram methods and found that the histogram methods were a good trade-off between accuracy and speed (Zhang *et al.*, 1993). In order to properly detect gradual transitions such as wipes and dissolves, they used two thresholds. If the histogram difference fell between the thresholds, they tentatively marked it as the beginning of a gradual transition sequence, and succeeding frames were compared against the first frame in the sequence. If the running difference exceeds the larger threshold, the sequence was marked as a gradual transition. To reduce the amount of processing needed,

they compared non-adjacent frames and did finer level comparisons for possible break detection (Boreczky, 1996).

Histogram and statistics-based metrics such as pixel-differences are sensitive to lighting changes such as light flickers between frames of the same shot. These variations alter the shape of the histogram and also the mean and variance of the gray/color level. As a result, a large metric values and false positives are produced. The advantage of these metrics is that they are invariant to large changes in object motion. The converse is true of pixel-difference comparisons: they are more robust with respect to lighting changes but are sensitive to large interframe changes due to motion, camera zooming, and panning (Fernando *et al.*, 2001).

4.4. *Motion Vector Counts*

Hard cuts are also accompanied by motion discontinuity. The simplest measure of motion is the pixel-wise frame difference (see Fig. 7). At hard cut locations, the pixel-wise difference is usually large (Zhang, 1993). Unfortunately, this simple motion measure is very susceptible to object and global camera motion. Even if global camera motion is compensated, object motion still poses a significant challenge to this feature.

More sophisticated motion features calculate the optical flow and use the number and distribution of motion vectors and the strength of the residual derived by block matching as features. However, the performance of block matching based methods is below the performance of intensity/color histogram based algorithms. The core problem with all motion features arises from the fact that reliable motion estimation is more difficult than detecting visual discontinuity, and thus less reliable. A simple detection problem is basically replaced by a much more complicated one.

Some researchers used the motion vectors determined from block matching to detect whether or not a shot was a zoom or a pan (Ueda *et al.*, 1991; Zhang *et al.*, 1993). Shahraray used the motion vectors extracted as part of the region-based pixel-differences computation described in order to decide if there is a large amount of camera or object motion in a shot. Because shots with camera motion can be incorrectly classified as gradual transitions, detecting zooms and pans increases the accuracy of a shot boundary detection algorithms.

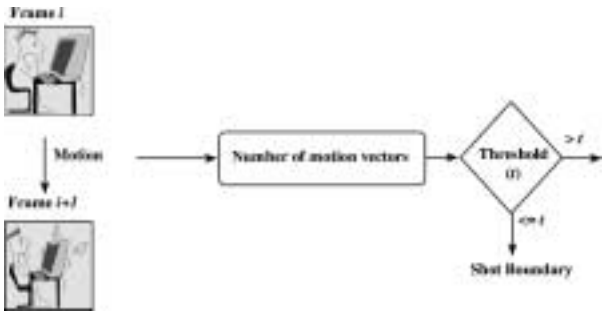

Fig. 7. Motion vector based technique.

4.5. *Three-Frames Approach*

The problem of scene change detection in image sequences has been studied by Sethi, and Patel (Sethi *et al.*, 1995). Their study was motivated by their desire to establish correspondence of objects across discontinuities in the camera motion. They suggested the use of three-frame approach to locate scene changes. Let k, l, and m be three consecutive frames and D_{kl}, D_{lm} be the measure of frames dissimilarities; the Observer Motion Coherence (OMC) is defined by:

$$OMC(k,l,m) = \left| \frac{D_{kl} - D_{lm}}{D_{kl} + D_{lm}} \right|. \tag{3}$$

Eq. 3 returns a value close to one if there is no scene change in the three-frames under consideration, and a value close to zero otherwise (Patel *et al.*, 1997; Sethi *et al.*, 1995). To measure similarity between a pair of frames, Sethi, Salari, and Vemuri suggested the use of either the normalized energy (NDE) measure or the absolute difference to summation.

4.6. *Incremental Ratio*

The basic idea here is that consecutive frames belonging to the same shot are in some way more similar than frames belonging to different shots, even in presence of rapid moving objects or camera pans. As a result, a cut exists when the difference between two frames is much larger than the standard differences between frames belonging to the same shot (Corridoni *et al.*, 1995). This idea of a relative difference between frames belonging to the same shot was expressed by the incremental ratio measure. The incremental ratio is computed as follows:

$$\delta = \frac{D\left(f_{t+1}, f_t\right)}{D\left(f_t, f_{t-1}\right)}. \tag{4}$$

Using the three statistic moments of the color histograms, the difference between any two successive frames is obtained by dividing each frame into n^2 sub-regions each of them originating a difference value D_i. Thus, a cut is detected whenever δ overcomes a predefined threshold and an index is set in the video accordingly. The difference value D_i is computed as follows:

$$D_i^{color}\left(f_t, f_{t+1}\right) = \sum_i a_i \left| m_i^{color}\left(t+1\right) - m_i^{color}\left(t\right) \right|, \tag{5}$$

where $m_i^{color}(t)$ represents the i-th moment of the histogram of the color (red, green, blue) component for the i-th frame, and **a** is a set of parameters experimentally tuned. The between frames difference is computed by adding the three-color differences for each region, discarding the k largest values, and finding the average of the remaining ones (Corridoni *et al.*, 1995; Brunelli *et al.*, 1999).

4.7. *Yakimovsky Likelihood Ratio*

This measure was originally proposed to detect the presence of an edge at the boundary of two regions (Sethi *et al.*, 1995). It is based on the computation of maximum likelihood which estimates the probabilities P_0 and P_1 that the following hypothesis H_0 and H_1 respectively, are true. The hypotheses are:

H_0: The two histograms are taken from the same scene, i.e., the same distribution.
H_1: The two histograms are taken from different scenes.

The estimated probabilities are computed under the assumption of Gaussian distribution. The expression of the Yakimovsky ratio, for frames of equal size is:

$$y = \Big(\frac{\sigma_0^2}{\sigma_{t-1}^2}\Big)\Big(\frac{\sigma_0^2}{\sigma_t^2}\Big), \tag{6}$$

where $\sigma_{t-1}^2, \sigma_t^2$ represent the variances of the past and current frames while σ_0^2 is the variance of the pooled data from both the histograms (Patel *et al.*, 1997). A scene change is detected between the previous and current frames if the Yakimovsky likelihood ratio y exceeds a certain threshold.

4.8. *Kolmogorov-Smirnov Test*

This test is based on calculating the cumulative distributions $C_1(x)$, $C_2(x)$ of the pixel luminance in two successive frames of two sets of data and measuring the maximum absolute difference between them. If the distributions are approximated by histograms, a defective estimate of the distance is obtained by:

$$D = \max_j |C_{1j} - C_{2j}|, \tag{7}$$

where j denotes the j-th bin (Brunelli *et al.*, 1999; Sethi *et al.*, 1995).

A high value of D indicates a scene change between two frames. We should remember here that the probability of a scene change occurrence can be calculated and used instead.

4.9. *Edge Change Ratio*

The edge detection scheme detects the appearance of intensity edges in a frame that are fixed distance away from the intensity edges in the previous frame. Thus the Edge Change Ratio (ECR) can be used to detect hard cuts. The ECR is defined as follows. *Let σ_n be the number of edge pixels in frames n, X_n^{in} and X_{n-1}^{out} the number of entering and exiting edge pixels in frames n and $n-1$ respectively. Then*

$$ECR_n = \max\big(X_n^{in}/\sigma_n,\ X_{n-1}^{out}/\sigma_{n-1}\big)$$

gives the edge change ratio (ECR_n) between frames $n-1$ and n. It ranges from 0 to 1. The edges are calculated by the Canny edge detector (Lienhart, 1999). Before extract

edge information from the consecutive frames, the frames are aligned to minimize the effects of camera motion. This is achieved by not considering edge pixels in one shot that have edge pixels in nearby of another shot as entering and exiting edge pixels (within 6 pixels's distance). In addition, a global motion compensation that is based on Hausdroff distance is used before the calculation of the ECR (Lienhart, 1999).

A drawback of this approach is that it requires the computation of motion compensation for a video. Since fast camera panning or zooming or fast object movements lead to a high ECR for successive frames even if there is no cut. Object motion or camera operations can be distinguished from a hard cut since they always last for several frames. Hard cuts, fades, dissolves and wipes demonstrate a characteristic pattern in the ECR time series. Hard cuts are detected as isolated peaks. An observation about fade in/fade out is that the number of incoming/outgoing edges is predominated. For a dissolve occurrence, initially the outgoing edges of the first shot protrude before the incoming edges of the second shot start to dominate the second half of a dissolve (Lienhart, 1999).

It should be noted that all the above methods require a comparison with appropriate thresholds to detect the camera breaks. The problem of choosing the appropriate threshold is a key issue in applying the shot detection schemes. Heuristically chosen global thresholds is inappropriate because experiments showed that the use of threshold for determining segment boundary varies from one shot to another which must be based on the distribution of the frame-to-frame differences of shots (Zhang *et al.*, 2001). Therefore, adaptive threshold is more reasonable than global threshold. A technique of choosing these thresholds is presented in (Zhang *et al.*, 2001). The selection of the threshold in this technique uses the local window based threshold calculation method combined with reliability verify process. In this technique, a sliding window was built so that it precedes the current frame. The mean and the variance of the sliding window are computed. Later the variance is analyzed within the sliding window. In other words, variance is used to analyze the reliability of calculated thresholds. Multi-pass approaches are also introduced by Zhange and others in order to reduce the computation time. A first scan of video material is done using a reduced threshold for camera breaks and under sampling in time the available frames. The video stream is then examined in correspondence of the candidate breaks at full temporal resolution.

Some of the previous mentioned algorithms assess the performance with respect to edit detection while others with their ability to classify correctly the type of the edits and its temporal extents. In practice 99% of all edits fall into one of the following 3 categories: 1) hard cuts; 2) fades; or 3) dissolve (Lienhart, 1999).

5. Gradual Effects Detection

As a result of the increased role of computer technology in video production, several types of complex gradual scene changes have begun to appear in video clips. These gradual transitions are used to enhance the quality of the video production. However, gradual transitions are more difficult to detect due to the difficulty of modeling of gradual shot

boundaries. Therefore, comparison based on successive frames alone is not adequate for the detection of gradual transitions (Fernando, 2001).

Gradual transition is an important class of effects and includes dissolving, fading in, fading out, mattes, and translates. A *fade* is a gradual transition between a scene and a constant image (fade out) or between a const image and a scene (fade in). During a fade, images have their intensities multiplied by some value α. During a fade in, α increase from 0 to 1, while during a fade out, α decrease from 1 to 0. The speed with which α changes controls the fade rate. A dissolve is a gradual transition from one scene to another, in which the first scene fades out and the second scene fades in. Typically, fade out and fade in begin at the same time, and the fade rate is constant. Another common scene break is a wipe, in which a line moves across the screen, with the new scene appearing behind the line (Zabih *et al.*, 1995). The presence of specific effects can then be used as a clue for detecting scene changes as opposed to simple shot detection. A *matte* is a progressive obscuration of visual field, due to a mask that invades the screen. A *translate* is a spatial edit: the first shot translates out, uncovering the shot that follows the edit. Different kinds of transitions are due to panning and zooming but these are not usually related to shot changes. Following is a survey of the common algorithms that have been proposed to detect gradual transitions.

5.1. *Twin Comparisons*

This scheme takes into account the cumulative differences between the frames and requires two cut-off thresholds: a higher threshold (T_h) for detecting abrupt transitions and a lower one (T_l) for gradual transitions. In the first stage the higher threshold is used for detecting abrupt transitions. In the next stage the lower threshold is used and any frame for which this threshold is exceeded is declared to be a potential transition start frame. However, for most gradual transition, the frame differences fall below the lower threshold. Therefore, such transitions can not be detected using the twin-comparison technique. Furthermore, this scheme is not suitable for real-time processing or for classifying gradual transitions. A comparison based on successive frames alone is not adequate for detecting gradual transitions because changes are small in this case (Fernando, 2001). One alternative is to use every k-th frame instead, i.e., to perform temporal subsampling. However, the larger separation between two frames used for comparison implies a significantly larger difference in statistics within a shot. Such an effect is especially pronounced in the case of camera/object motion.

Another scheme which will be described in the following sub-section is when every frame is used and compared to the k-th following frame.

5.2. *Plateau Detection*

The Plateau research group headed by Professor L. Rowe introduced this approach. It was noted that comparison based on successive frames alone is not adequate for the detection of gradual transition because changes are small (Brunelli *et al.*, 1999). Plateau compares

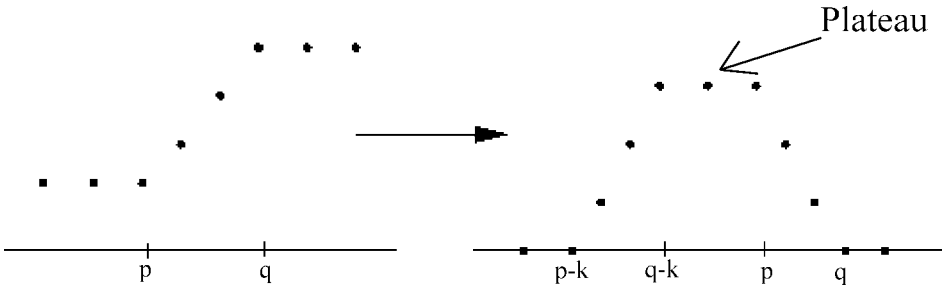

Fig. 8. The Plateau method of detecting gradual transitions.

the features of the of the i-th frame with that of the $(n-k)$-th frame instead of adjacent ones. Then the distance measure plotted as a function of time shows a plateau-like graph and plateau detection is performed to detect the shot boundaries. In this algorithm, abrupt changes and gradual transitions are detected separately. An abrupt scene change is declared if there is a sharp peak in the D_i plot, where D_i is simply the difference between frame i and frame $i+1$. A robust detector for gradual transition in this scheme is obtained as follows: it compares every i-th frame to the $(n-k)$-th frame using the sum of absolute difference between the two frames as difference metric. The plot of these deferred differences yields a plateau if k is greater than the length of the gradual transition. If the transition occurs from frame i to frame j, then a plateau should appear from $j-k$ to i (k should be of at least slightly greater than $j-i$ for the plateau to be present) in the deferred difference plot (see Fig. 8). This algorithm suggests two criteria to detect the points on the difference plot. The first criterion test is to see that the points (within a variation between 10 – 20%) around a point on the plateau are within the neighborhood. The second criterion tests to see if the plateau stands out by the following:

$$D_i^k \geqslant l \times D_{i-\lfloor k/2 \rfloor - 1}^k \quad \text{or } D_i^k \geqslant l \times D_{i+\lfloor k/2 \rfloor + 1}^k, \quad l \gg 1, \tag{8}$$

where D_i is the difference between frame i and frame $i+k$, and l is the threshold parameter (Brunelli *et al.*, 1999).

The quality of performance is measured by distinguishability of plateau's of gradual transitions. However, due to the fact that such plateau's can appear also by strong object or camera motion, it is necessary to distinguish between these effects to avoid false detections in high-action scenes (Brunelli *et al.*, 1999; Lienhart, 2001).

5.3. *Detection by Modeling*

Another important approach to the detection of video effects is based on effect modeling (Corridoni *et al.*, 1995; Hampapur *et al.*, 1995). Mathematical models of video based on video production techniques are formulated. Hampapur *et al.* (1995) provides an explicit model of video, which is called the production model based classification approach to segmentation. The proposed model contains three components where the chromatic scaling is one of them. In their model, the chromatic scaling was used to detect a variety of

scene breaks. Their approach to dissolves is novel because it does not involve the use of histograms. Instead, they compute a chromatic image from a pair of consecutive images. Its value at each pixel is the change in intensity between the two images divided by the intensity in the later image. Ideally, the chromatic image should be uniform and non-zero during a fade. They proposed dissolves detection by measuring the uniformity of the chromatic image. Hampapur and his colleagues noted that in their discussion of dissolves that their measure "is applicable if the change due to the editing dominates the change due to motion", and describe both object and camera motion as causes of false positives for their model.

A dissolve operation from scene X to scene Y is a sequence of frames represented by:

$$E(t) = \left[\frac{t}{L_2}Y\right]_{t\in[T_2,T_2+L_2]} + \left[\left(1-\frac{t}{L_1}\right)X\right]_{t\in[T_1,T_1+L_1]}, \tag{9}$$

where $T1,\ T2$ represent the times at which the scaling of $X,\ Y$ begin and $L1,\ L2$, are the duration from which the scaling last (Brunelli *et al.*, 1999). Note that a fade in is a special case when X equals to 0 and a fade out is also a special case characterized when Y equals to 0. If we look at a fade out, then under the assumption that the only source of variation is the scaling, the following holds:

$$F(t) = \frac{\delta E/\delta t}{X(x,y,t)} = -\frac{1}{L}, \tag{10}$$

where L represents the length of the dissolve expressing in frames (Brunelli *et al.*, 1999). The different possible dissolves can be classified according to their start/end times. By looking at the resulting categories, it can be seen that using (10) to detect them is appropriate except when very similar sequences are being dissolved with precisely equal fade rates over the dissolve. This particular situation is rarely encountered in most commercially produced videos. A similar approach can be used to detect spatial edits, such as translates. As shown by Hampapur and others, assuming that there is no scene action in progress during the edit, the following holds when the edits is a pure translation in the x directions:

$$T(x,y,t) = \frac{\delta E/\delta t}{\delta E/\delta x}, \tag{11}$$

where $E(t)$ represents the sequence of edited frames (Brunelli *et al.*, 1999). Again, the effect can be detected by looking for a constant image $T(x,y,t)$. One limitation of using chromatic scaling is that chromatic translations and rotations are difficult to detect.

A method for detecting mattes is proposed in (Corridoni *et al.*, 1995). They are similar to fades, apart from the fact the luminance varies over the frames following a geometrical law. Once a fade has been detected, the central frames are considered: if the luminance has a big discontinuity, due to the presence of a black mask that partially covers the frame, a matte is detected.

5.4. *Feature Based Detection*

Another interesting approach to the detection of gradual effects as well as hard cuts has been presented in (Zabih *et al.*, 1995). During a cut or a dissolve, a new intensity edges appear far from location of old edges. Edge pixels that appear far from existing edge pixels are considered as entering edge pixels, while edge pixels that disappear far from an exiting edge pixel as an existing edge pixel. Cuts, fades and dissolves can be detected by counting the entering and exiting edge pixels, while wipes can be detected by looking at their spatial distribution. The algorithm is based on the following steps:

- align frames F_t and F_{t+1} by using a global motion compensation algorithm;
- compute edges by applying Canny algorithm to a smoothed version of the frames;
- dilate the binary edge maps radius r, so that the condition on the mutual distance of edge pixels can be easily verified by set intersection;
- compute the fraction of entering edge pixels ρ_{in} and exiting edge pixel ρ_{out}.

Breaks are simply detected by looking at the edge change fraction $p = \max(\rho_{in}, \rho_{out})$. A cut will lead to a single isolated high value of ρ while the other scene breaks will leads to an interval where $\rho' s$ value is high. During a fade in the value ρ_{in} will be much higher than ρ_{out}. The reverse happens for fade outs. A dissolve is characterized by a predominance of ρ_{in} during the first phase and of ρ_{out} during the second phase. The technique works properly also on heavily compressed image sequences.

This algorithm does not only detect shot transitions, but also classifies the types of shot transitions using the change of fraction of entering/exiting edges. The experimental results showed that the performance of this algorithm is very good, but this algorithm requires too much processing time. In addition, the feature-base detection algorithm has two limitations; first its detection does not handle rapid changes in overall scene brightness, or scenes, which are very dark or very bright. Second, it does not handle multiple rapidly moving objects particularly well (Zahib *et al.*, 1995). In general, it's robustly tolerates motion, as well as compression and artifacts.

There was a study conducted by Paul and others to compare three different shot boundary detection methods with a fixed threshold and dynamic threshold (Browne *et al.*, 2000). They found that using a fixed threshold is a drawback in detecting boundaries. The reason they explained is that the static threshold did not take into account the variable program types (They used a collection of 8 hours of TV broadcasting). Their results showed that using dynamic or adaptive threshold (where the threshold used will rise and fall depending on the type of the program being analyzed) value is better. They developed an approach to dynamically adjust the threshold by looking at the visual noise across a future window of frames and from this raising or lowering the threshold as appropriate. Later in their work, they were thinking to use combined shot boundaries detection algorithms; however, they decided to use what they call "weighted boolean logic" to combine these methods. This improves the precision. Their results verified that using more one scheme will improve the detection process.

6. Discussion

The solution to identifying abrupt cuts was done by looking for rapid changes in color histogram or image intensity mostly. The gradual cuts have also been developed to successfully detect fades, dissolves, and page translate edits. Once shots have been identified, key frames which characterize the shots can be selected, for example, by selecting the motion of objects within the shot. This will help us in combining information from multiple frames of an image sequence to create a "salient video still" which characterize the shot in some way or some other forms of visual summaries. These methods vary considerably in their computational complexity and effectiveness for different video sources, but each has its merits.

The detection of shot transitions can be trivial or complex depending on the video content being combined and the type of transition used. For example, when video from two very different sources are spliced together with zero frames of transition, it is easy to detect the scene change. One the other hand, if two very similar shots are combined with a gradual cross fade; the visual changes may be much smaller than we might expect in a video with moderate object motion. Thus, it is very likely that any automated image-based shot detection algorithm will miss some fraction of the shot boundaries. Fortunately, this does not impact the quality of the scene detection greatly because shot transitions which are gradual are often chosen by procedures since the two shots are actually related and should remain in the same scene.

7. Conclusions

A powerful scene detection algorithm is required in order to characterize video sequences completely for indexing and retrieval purposes. In this paper, several scene detection algorithms in uncompressed domain have been discussed. These algorithms support user annotation and automatic image and shot annotation.

Many algorithms rely on the use of histogram comparison (of intensity, color, etc) because the global nature of histograms makes them less sensitive to the typical changes within a shot, thereby reducing the number of false positive with respect to pixel by pixel comparisons.

However, the automatic video partition is still a very challenging research problem especially for detecting gradual transitions, false detection of shot transition due to flashing light, threshold selection, and to convey a meaningful storyline or narrative from the regions of the video frames. Further work is required in this area.

References

Aslandogan, Y.A., and C.T. Yu (1999). Techniques and systems for image and video retrieval, In *IEEE Trans. on Knowledge and Data Engineering*, **11**(1). pp. 56–63.

Boreczky, J.S., and L.A. Rowe (1996). Comparison of video shot boundary detection techniques, In *Proc. of Storage and Retrieval for Image and Video Databases IV*, SPIE 2670. pp. 170–179.

Browne, P., A. Smeaton, N. Murphy, N. O'Connor, S. Marlow and C. Berrut (2000). Evaluating and combining digital video shot boundary detection algorithms, In *Proc. of Fourth Irish Machine Vision and Image Processing Conference* (*IMCIP 2000*), Belfast, Northern Ireland. pp. 211–234.

Brunelli, R., O. Mich and C.M. Modena (1999). A survey on video indexing, *Journal of Visual Communication and Image Representation*, **10**(2), 78–112.

Chen, S.Shyu, M., C. Zhange and R. Kashyap (2001). Video scene change detection method using unsupervised segmentation and object tracking, In *IEEE International Conference on Multimedia and Expo* (*ICME2001*), Tokyo, Japan. pp. 57–60.

Corridoni, J.M., and A.D. Bimbo (1995). Film editing reconstruction and semantic analysis, In *Proc. of the International Conference on Analysis of Image Patterns*, Prague, Czech Republic. pp. 938–943.

Corridoni, J., A. Bimbo and D. Lucarella (1995). Navigation and visualization of movies content, In *Proc. of the 11th International IEEE Symposium on Visual Languages*, Germany. pp. 217–225.

Fernando, W.A.C., C.N. Canagarajah and D.R. Bull (2001). Scene change detection algorithms for context-based video indexing and retrieval, *Electronics and Communication Engineering Journal*, **13**(3), 117–126.

Hampapur, A., R. Jain and T. Weymouth (1995). Production model based digital video segmentation, *Journal of Multimedia Tools and Applications*, **1**(1), 9–46.

Irani, M., and P. Anandan (1998). Video indexing based on mosaic representations, In *Proc. of IEEE on Pattern Analysis and Machine Learning*, **86**(5). pp. 905–921.

Lee, J., and B.W. Dickinson (2000). Hierarchical video indexing and retrieval for subband-coded video, In *IEEE Trans. on Circuits and Systems for Video Technology*, **10**(5). pp. 824–829.

Lienhart, R. (1999). Comparison of automatic shot boundary detection algorithms, In *Proc. of Storage and Retrieval for Image and Video Databases VII*, Vol. 3656, CA. pp. 290–301.

Lienhart, R. (2001). Reliable transition detection in videos: a survey and practitioner's guide. *International Journal of Image and Graphics*, **1**(3), 469–486.

Patel, N.V., and I.K. Sethi (1997). Video shot detection and characterization for video databases, *Pattern Recognition*, **30**(4), 583–592.

Shahraray, B. (1995). Scene change detection and content-based sampling of video sequences, In *Proc. of Digital Video Compression: Algorithm and Technologies*, SPIE 2419, CA. pp. 2–13.

Sethi, I.K., and N. Patel (1995). A statistical approach to scene change detection, In *Proc. of Storage and Retrieval for Image and Video Databases III*, SPIE 2420, VA, USA. pp. 329–339.

Swanberg, D., C.H. Shu and R. Jain (1993). Knowledge guided parsing and retrieval in video databases, In *Proc. of Storage and Retrieval for Image and Video Databases*, SPIE 1908. pp. 173–187.

Ueda, H., T. Miyatake and S. Yoshizawa (1991). IMPACT: an interactive natural-motion-pictures dedicated multimedia authoring system, In *Proc. of CHI*, (New Orleans, LA) ACM, New York. pp. 343–350.

Zabih, R., J. Miller and K. Mai (1995). Feature-based algorithms for detecting and classifying scene breaks, In *Proc. of the 4th ACM Conference on Multimedia*, CA. pp. 189–200.

Zhang, D., W. Qi and H. Zhang (2001). A new shot boundary detection algorithm, In *Proceedings of Second IEEE Pacific Rim Conference on Multimedia*, China. pp. 63–70.

Zhang, H.J., A. Kankanhalli and S.W. Smoliar (1993). Automatic partitioning of full-motion video, *Multimedia Systems*, **1**(1), 10–28.

E. El-Qawasmeh received his B.Sc. degree in Computer Science in 1985 from Yarmouk University, Jordan. He then joined the Yarmouk University as teaching assistant in the Computer Science Department. In 1992, he joined the George Washington University, Washington, D.C., USA where he obtained his MS and Ph.D. degrees in Software and Systems in 1994 and 1997, respectively. In 2001, he joined George Washington University as visiting researcher. His areas of interest include multimedia databases, information retrieval, and object-oriented. Dr. El-Qawasmeh is currently an assistant professor in the Department of the Computer Science and Information Systems at Jordan University of Science and Technology, Jordan. He is a member of the ACM and IEEE.

Vaizdų pokyčių aptikimo schemos indeksuojant negludintus vaizdų duomenis

Eyas EL-QAWASMEH

Taikomosiose programose vis dažniau naudojami vaizdų įrašai. Kad būtų galima greitai iškviesti saugomą vaizdinę informaciją, ją indeksuoti, analizuoti vaizdus, reikia turėti efektyvią prieigą prie jų. To siekiant yra sukurta daug pokyčių (staigių ir lėtų) aptikimo algoritmų, kai analizuojamieji vaizdai yra neapdoroti – neglaudinti. Šiame straipsnyje pateikiamas rėžio tarp sričių aptikimo ir klasifikavimo būdų palyginimas, naudojant histogramas, pikselių skirtumus, tikėtinumo santykį ir judesio vektorių. Yra pateikiami argumentai už ir prieš minėtus metodus bei paaiškinantys komentarai.