Scene change detection algorithms for content-based video indexing and retrieval

by W. A. C. Fernando, C. N. Canagarajah and D. R. Bull

There is an urgent need to extract key information from video automatically for the purposes of indexing, fast retrieval, and scene analysis. To support this vision, reliable scene change detection algorithms must be developed. Several algorithms have been proposed for both sudden and gradual scene change detection in uncompressed and compressed video. In this paper some common algorithms that have been proposed for scene change detection are reviewed. A novel algorithm for sudden scene change detection for MPEG-2 compressed video is then presented. This uses the number of interpolated macroblocks in B-frames to identify the sudden scene changes. A gradual scene change detection algorithm based on statistical features is also presented.

1 Introduction

With the development of various multimedia compression standards coupled with significant increases in desktop computer performance and storage, the widespread exchange of multimedia information is becoming a reality. Audio-visual information is becoming available in digital form in various places around the world. As more and more of this information appears, finding the desired information becomes increasingly difficult. Currently, solutions exist that allow searching for textual information. Many text-based search engines are available on the World Wide Web. However, searching information based on content is difficult for audio-visual content, as no generally recognised or standardised descriptions of this material exist. To this end, MPEG (the Moving Pictures Experts Group) will set a standard, called 'Multimedia Content Description Interface (MPEG-7)', or 'MPEG-7' in short, that will extend the limited search capabilities of today to allow efficient retrieval of multimedia information. It is envisaged that this information will be associated with the content itself, to allow fast and efficient searching for material of interest.

Today, the major bottleneck preventing the widespread use of digital video is the present slow retrieval of desired information based on content from a huge database. A reliable way of solving this problem is to index the video sequence using a suitable descriptor, thus enabling fast access to the video clips stored in a multimedia database. Video images contain a wider range of primitive data types (the most obvious being motion vectors) and occupy far more storage than still images; they can take hours to review compared to a few seconds at the most for still images. The process of organising video for retrieval is, in some ways, akin to that of abstracting and indexing long text documents. All but the shortest videos are made up of a number of distinct scenes, each of which can be further broken down into individual shots depicting a single view, conversation or action. A common way of organising a video for retrieval is to prepare a storyboard of annotated still images (often known as key frames) representing each scene.

The most common approach to content-based video segmentation is shot transition detection: the video sequence is partitioned into shots, each video shot representing a meaningful event or a continuous sequence of action. Shot transitions can be divided into two categories: abrupt transitions and gradual transitions. Gradual transitions include camera movements—panning, tilting, zooming—and video editing special effects—fade-in, fade-out, dissolving, wiping. Segmentation into shots is the first of the indexing phases in Fig. 1, which shows a block diagram of a video database management system for content-based video indexing and retrieval.

Once shots have been identified, key frames for each shot must be selected. Several techniques have been proposed for key frame 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 colour, texture and shape are available—a technology now generally referred to as content-based image retrieval (CBIR). CBIR technology is now beginning to move out of the laboratory and into the marketplace, in the form of commercial products like QBIC and Virage.

The large channel bandwidth and memory requirements for the transmission and storage of image and video necessitate the use of video compression techniques. Hence, the visual data in multimedia databases is expected to be stored mostly in compressed form. To avoid unnecessary decompression operations in indexing and searching processes, it is therefore preferable to index images and video in their compressed format. A common and natural idea is first to index the compressed video sequences into video shots by identifying changes that take place in a scene in the compressed domain itself. Therefore, shot change detection in the compressed domain is also required to allow for a complete characterisation of a video sequence.

The above discussion has shown that searching for information from a huge video database based on its audio-visual content is a difficult task, the more so if it is done in the compressed domain itself. As explained earlier, the ability to identify shot transitions automatically is the first step towards automatic video indexing or video storyboard browsing. Therefore the main objective of this paper is to review some of the currently used algorithms and to present some new algorithms for shot transition detection in uncompressed and compressed video sequences.

2 Sudden (abrupt) transitions

Abrupt shot transitions are very easy to detect as the two frames being compared are completely uncorrelated. Most previous work on detecting a sudden scene change is based on entire images and uses difference metrics to evaluate the changes between successive frames.

Uncompressed domain abrupt transitions

Zhang proposed that a change between two frames could be detected by comparing the difference in the intensity values of corresponding pixels in the two frames. His algorithm counts the number of pixels that have changed, and an abrupt transition is declared if the number of pixels that have changed, expressed as a percentage of the total number of pixels, exceeds a certain threshold. However, this technique may produce false alarms since camera movements can have the same effect on a large number of pixels and hence a scene change will be detected. Fast moving objects also have the same effect. Therefore, detecting sudden scene changes at the pixel level is not a very robust approach.

In the likelihood ratio approach, the frames are subdivided into blocks, which are then compared on the basis of the statistical characteristics of their intensity levels. Eqn. 1 represents the formula that calculates the likelihood function, \( \lambda \):

\[
\lambda = \frac{\sigma_i + \sigma_{i+1}}{2} + \frac{(\mu_i - \mu_{i+1})^2}{2\sigma_i\sigma_{i+1}}
\]

\( \mu_i \) and \( \mu_{i+1} \) are the mean intensity values for a given region in two consecutive frames and \( \sigma_i \) and \( \sigma_{i+1} \) are the corresponding variances. The number of blocks for which \( \lambda \) exceeds a certain threshold are counted and if this number exceeds a certain value a scene change is declared. A subset of the blocks can be used to detect the difference between the images so as to expedite the process of block matching. This approach is better than the pixel-based approach as it increases the tolerance to

---

**Fig. 1 Block diagram of a video database management system for content-based video indexing and retrieval**

indexing phase

*temporal segmentation*

index each still image (key image). Image indexing and retrieval methods based on automatically-derived features such as colour, texture and shape are available—a technology now generally referred to as content-based image retrieval (CBIR). CBIR technology is now beginning to move out of the laboratory and into the marketplace, in the form of commercial products like QBIC and Virage.

The large channel bandwidth and memory requirements for the transmission and storage of image and video necessitate the use of video compression techniques. Hence, the visual data in multimedia databases is expected to be stored mostly in compressed form. To avoid unnecessary decompression operations in indexing and searching processes, it is therefore preferable to index images and video in their compressed format. A common and natural idea is first to index the compressed video sequences into video shots by identifying changes that take place in a scene in the compressed domain itself. Therefore, shot change detection in the compressed domain is also required to allow for a complete characterisation of a video sequence.

The above discussion has shown that searching for information from a huge video database based on its audio-visual content is a difficult task, the more so if it is done in the compressed domain itself. As explained earlier, the ability to identify shot transitions automatically is the first step towards automatic video indexing or video storyboard browsing. Therefore the main objective of this paper is to review some of the currently used algorithms and to present some new algorithms for shot transition detection in uncompressed and compressed video sequences.

2 Sudden (abrupt) transitions

Abrupt shot transitions are very easy to detect as the two frames being compared are completely uncorrelated. Most previous work on detecting a sudden scene change is based on entire images and uses difference metrics to evaluate the changes between successive frames.

Uncompressed domain abrupt transitions

Zhang proposed that a change between two frames could be detected by comparing the difference in the intensity values of corresponding pixels in the two frames. His algorithm counts the number of pixels that have changed, and an abrupt transition is declared if the number of pixels that have changed, expressed as a percentage of the total number of pixels, exceeds a certain threshold. However, this technique may produce false alarms since camera movements can have the same effect on a large number of pixels and hence a scene change will be detected. Fast moving objects also have the same effect. Therefore, detecting sudden scene changes at the pixel level is not a very robust approach.

In the likelihood ratio approach, the frames are subdivided into blocks, which are then compared on the basis of the statistical characteristics of their intensity levels. Eqn. 1 represents the formula that calculates the likelihood function, \( \lambda \):

\[
\lambda = \frac{\sigma_i + \sigma_{i+1}}{2} + \frac{(\mu_i - \mu_{i+1})^2}{2\sigma_i\sigma_{i+1}}
\]

\( \mu_i \) and \( \mu_{i+1} \) are the mean intensity values for a given region in two consecutive frames and \( \sigma_i \) and \( \sigma_{i+1} \) are the corresponding variances. The number of blocks for which \( \lambda \) exceeds a certain threshold are counted and if this number exceeds a certain value a scene change is declared. A subset of the blocks can be used to detect the difference between the images so as to expedite the process of block matching. This approach is better than the pixel-based approach as it increases the tolerance to
noise associated with camera and object movement. It is possible that even though the two corresponding blocks are different they can have the same density function and in such cases no change is detected.

Sensitivity to camera and object motion can be further reduced by comparing the grey level histograms of the two frames. This is because two frames whose backgrounds differ little and which have the same amount of object motion have almost the same histograms. The histogram is given by the number of pixels belonging to each grey level in the frame. The histogram metric is defined on the left-hand side of eqn. 2:

\[ \sum_{j=1}^{G} |H(j) - H_{n}(j)| > T_{k} \]

where \( G \) is the number of grey levels, \( j \) is the grey value, \( i \) is the frame number, and \( H(j) \) is the value of the histogram for the grey level \( j \). If the sum of the absolute differences of corresponding values of consecutive histograms is greater than a given threshold \( T_{k} \), then a transition is declared. The histogram-based algorithm for detecting shot boundaries is one of the most reliable detection algorithms.

Zabith et al. have proposed a feature-based approach for detecting sudden scene changes. During a cut, new intensity edges appear far from the locations of old edges. Similarly, old edges disappear far from the location of new edges. By counting the number of entering and exiting edge pixels, an abrupt scene change can be identified. However, this algorithm requires edge detection in every frame, which is computationally very costly. Another limitation of this scheme is that the edge detection method does not handle rapid changes in overall scene brightness or scenes with high contrast levels.

Histogram and statistic-based metrics are sensitive to lighting changes: for example, if the light flickers between frames of the same shot. These variations alter the shape of the histogram and also the mean and variance of the grey level. This produces large metric values and false positives. 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 and camera zooming and panning.

**Compressed domain abrupt transitions**

**MPEG-2 overview**

MPEG-2 video compression is used in many current and emerging products for digital television and broadcasting. MPEG-2 video is broken up into a hierarchy of layers to help with error handling, random search editing, and synchronisation. The first (top) layer is known as the video sequence layer. The second layer down is the group of pictures (GOP) layer, a GOP comprising one or more groups of intra (I) frames and/or non-intra (P and/or B) frames and begins with an I-frame (see Fig. 2). An I-frame is encoded with no reference to other frames; no motion compensation is performed. A P-frame is predictively encoded with motion compensation from past I- or P-frames. A B-, or bidirectional, frame is encoded using motion compensation with reference to past, future, or both past and future I- or P-frames. Each GOP is divided into subunits, called sub-GOPs, which contain B-frames and an I- or a P-frame. The third layer down is the picture layer itself, and the layer below that is called the slice layer. Each slice consists of macroblocks (MBs), which are 16 x 16 arrays of luminance pixels, or picture data elements, with 8 x 8 arrays of associated chrominance pixels. Macroblocks are the units for motion-compensated compression. The macroblocks are further divided into 8 x 8 blocks of pixels. Fig. 2 shows a typical MPEG-2 video sequence including a GOP of 12 frames; the sub-GOP size is 3.

Compression of the video is carried out by dividing each picture into a set of 8 x 8 pixel blocks. The pixels in a block are transformed into 64 coefficients using the discrete cosine transform (DCT), and these coefficients are then quantised and Huffman entropy encoded.

**Sudden scene change detection in MPEG-2**

Because the coefficients in the frequency (DCT) domain are mathematically related to the pixels in the spatial domain, they can be used in detecting changes in the video. One approach to using the DCT coefficients to find frames where camera breaks occur is as follows:

\[ \sum_{i=1}^{G} |H_{i}(j) - H_{n}(j)| > T_{k} \]

where \( G \) is the number of grey levels, \( j \) is the grey value, \( i \) is the frame number, and \( H_{i}(j) \) is the value of the histogram for the grey level \( j \). If the sum of the absolute differences of corresponding values of consecutive histograms is greater than a given threshold \( T_{k} \), then a transition is declared. The histogram-based algorithm for detecting shot boundaries is one of the most reliable detection algorithms.

Zabith et al. have proposed a feature-based approach for detecting sudden scene changes. During a cut, new intensity edges appear far from the locations of old edges. Similarly, old edges disappear far from the location of new edges. By counting the number of entering and exiting edge pixels, an abrupt scene change can be identified. However, this algorithm requires edge detection in every frame, which is computationally very costly. Another limitation of this scheme is that the edge detection method does not handle rapid changes in overall scene brightness or scenes with high contrast levels.

Histogram and statistic-based metrics are sensitive to lighting changes: for example, if the light flickers between frames of the same shot. These variations alter the shape of the histogram and also the mean and variance of the grey level. This produces large metric values and false positives. 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 and camera zooming and panning.

**Compressed domain abrupt transitions**

**MPEG-2 overview**

MPEG-2 video compression is used in many current and emerging products for digital television and broadcasting. MPEG-2 video is broken up into a hierarchy of layers to help with error handling, random search editing, and synchronisation. The first (top) layer is known as the video sequence layer. The second layer down is the group of pictures (GOP) layer, a GOP comprising one or more groups of intra (I) frames and/or non-intra (P and/or B) frames and begins with an I-frame (see Fig. 2). An I-frame is encoded with no reference to other frames; no motion compensation is performed. A P-frame is predictively encoded with motion compensation from past I- or P-frames. A B-, or bidirectional, frame is encoded using motion motion compensation with reference to past, future, or both past and future I- or P-frames. Each GOP is divided into subunits, called sub-GOPs, which contain B-frames and an I- or a P-frame. The third layer down is the picture layer itself, and the layer below that is called the slice layer. Each slice consists of macroblocks (MBs), which are 16 x 16 arrays of luminance pixels, or picture data elements, with 8 x 8 arrays of associated chrominance pixels. Macroblocks are the units for motion-compensated compression. The macroblocks are further divided into 8 x 8 blocks of pixels. Fig. 2 shows a typical MPEG-2 video sequence including a GOP of 12 frames; the sub-GOP size is 3.

Compression of the video is carried out by dividing each picture into a set of 8 x 8 pixel blocks. The pixels in a block are transformed into 64 coefficients using the discrete cosine transform (DCT), and these coefficients are then quantised and Huffman entropy encoded.

**Sudden scene change detection in MPEG-2**

Because the coefficients in the frequency (DCT) domain are mathematically related to the pixels in the spatial domain, they can be used in detecting changes in the video. One approach to using the DCT coefficients to find frames where camera breaks occur is as follows:

\[ \sum_{i=1}^{G} |H_{i}(j) - H_{n}(j)| > T_{k} \]
Fig. 3 Flowchart for the proposed sudden-scene-change-detection algorithm. \( t \) is the frame number, and \( T \) a threshold. \( T_a = 5\% \), \( T_s = 60\% \)

From the 8 \( \times \) 8 pixel blocks of a single video frame, \( m \), that have been encoded using the DCT, a subset of blocks is chosen \textit{a priori}. The blocks are chosen from \( n \) connected regions in each frame. Again a subset of the 64 coefficients for each block is chosen. The coefficients chosen are randomly distributed among the AC coefficients of the blocks. Taking coefficients \( c \) from each frame a vector is formed as follows:

\[ V_n = \{ c_1, c_2, c_3, \ldots \} \]

(3)

This vector represents the frame of the video sequence in DCT space. The inner product is used to find the difference between the two frames:

\[ \psi = \frac{V_n V_{n+1}}{|V_n| V_{n+1}|} \]

(4)

where \( V_n \) is the vector of the frame being compared and \( V_{n+1} \) is the vector of the successor frame. A transition is detected when \( 1 - |\psi| > t \), where \( t \) is some threshold.

Zhang et al.\(^1\) have also experimented with motion-based segmentation using the motion vectors in the MPEG compressed data as well as the DCT coefficients. Meng et al.\(^{10}\) have extended this concept further by performing more detailed operations on the MPEG compressed data. If a break occurs at a B-frame, most of the motion vectors will come from the following anchor frame (I or P-frame) and few will come from the previous anchor frame (see Fig. 2). A scene cut is detected based on the ratio of backward and forward motion vectors. When a scene change occurs at a P-frame the encoder cannot use macroblocks from the previous anchor frame for motion compensation as P-frames have only forward motion compensation. A scene break is detected based on the ratio of macroblocks without motion compensation to macroblocks with motion compensation. Since I-frames are completely intra-coded, without motion vectors, the method using DCT coefficients described above\(^{16}\) can be used for scene change detection.

Most of the algorithms proposed for sudden scene change detection in the compressed domain fail when the sequence contains special effects like fading and dissolving. This is because the encoder uses more intra-coded macroblocks for P-frames to code these special effects. We have therefore proposed the algorithm described in the following subsection to eliminate this problem.

**Proposed algorithm for sudden scene change detection in MPEG-2**

In MPEG-2 compressed video, B-pictures have three types of macroblocks (only rarely are a few macroblocks intra-coded): forward predicted macroblocks (predicted
from previous I-/P-frame), backward predicted macroblocks (predicted from next I-/P-frame) and interpolated macroblocks (predicted from both previous and next I-/P-frames). Thus, the number of interpolated macroblocks ($N_{IP}$) for a given B-frame measures the strength of correlation between the previous and next I-/P-frames. If the number of interpolated macroblocks for a given first B-frame is high, then there is a strong correlation between the previous and next I-/P-frames with respect to the current B-frame. Therefore, it is not possible for an abrupt scene change to occur between these two I-/P-frames in the vicinity of the current B-frame. If the number of interpolated macroblocks is below a certain threshold, this indicates that the previous and next I-/P-frames are not correlated. This situation implies an abrupt scene change at one of the B-frames or at the next I-/P-frame under consideration. If the number of backward predicted macroblocks for the first B-frame ($N_{BP}$) is high, then most of the macroblocks for the first B-frame have been predicted from the next I-/P-frame and therefore the scene change must have occurred at the first B-frame. If the number of backward predicted macroblocks is below the threshold, then the algorithm checks the number of backward predicted macroblocks for the second B-frame ($N_{B2}$) against the same threshold used for first B-frame. If the threshold is exceeded, then the scene change is declared to be at the second B-frame; if it is not satisfied then the scene change should have occurred at the next I-/P-frame. The flow chart for the complete algorithm is shown in Fig. 3. It is interesting to note that this algorithm is independent of the GOP structure and can also be run in real time.

Fig. 4 shows the variation in the number of interpolated macroblocks as a percentage of the total number, $N$, of macroblocks for the first B-frame ($B_1$). A value of ($N_{IP}/N$) less than the threshold $T_{IP}$ (5%) indicates the sub-GOP where abrupt scene changes occur. Figs. 5 and 6 show the numbers of forward- and backward-predicted macroblocks for first and second

![Fig. 4 Number of interpolated macroblocks (%) against frame number for the first B-frame ($B_1$)](image)

![Fig. 5 Number of forward- and backward-predicted macroblocks (%) against frame number for the first B-frame ($B_1$)](image)

![Fig. 6 Number of forward- and backward-predicted macroblocks (%) against frame number for the second B-frame ($B_2$)](image)
B-frames, respectively. Abrupt scene changes can now be determined by comparing $N_{B1}$ and $N_{B2}$ to the threshold $T_h$ (60%). For instance, the algorithm identifies frame number 20 (where the relative number of interpolated macroblocks is below $T_{rel}$, Fig. 4) as a candidate for sudden scene change detection. From Fig. 5, it is clear that $N_{B1}$ is less than $T_h$ and so there is no sudden scene change at $B_1$. However, from Fig. 6, it is can be seen that $N_{B2}$ exceeds $T_h$ and so a scene change was declared at $B_2$ (frame 21). Likewise, following the same argument, other scene changes were detected at frame numbers 36 ($B_3$), 50 ($B_4$), 81 ($B_5$) and 97 ($P$).

This algorithm is very efficient and eliminates cumbersome variance calculations or partial decompression of the compressed data. Furthermore, this algorithm can detect abrupt scene changes anywhere in the scene very accurately, even when the sequence contains special effects. This was tested using sequences containing more than 100 sudden changes and was observed to be reliable and accurate.

Once the locations of sudden scene changes have been identified, we can distinguish the shots for video indexing. Furthermore, in the uncompressed domain, an I-frame can be used to encode the start of the new shot and the GOP structure changed within each shot until the next scene change is found.

### 3 Gradual transitions

With 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 as the difference between frames corresponding to two successive shots is substantially reduced. Therefore, comparison based on successive frames alone is not adequate for detecting gradual transitions because changes are small in this case. 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. In the scheme described in Reference 20, every frame is used and compared to the $k$th following frame. Let $X_k$ ($i = 1, 2, 3, \ldots N$) be a sequence of spatially reduced images and $D^k$ be defined as:

$$D^k = d(X_i, X_{i+k}), \quad i = 1, 2, 3, \ldots (N-k)$$

where

$$d(X,Y) = \sum_{ij} |x_{ij} - y_{ij}|$$

A gradual transition in the form of a linear transition of some variable from $c_1$ to $c_2$ in the time interval $[a_1, a_2]$ can be modelled by a function $u$ as follows:

$$u_n = \begin{cases} 
  c_1 & n < a_1 \\
  c_2 - c_1, & a_1 \leq n \leq a_2 \\
  c_2 & n > a_2
\end{cases}$$

If $k > a_2 - a_1$, a plateau in $D^k(u_n)$ can be observed during the transition. The main problem with this scheme is that it is very difficult to select the value for $k$ since gradual transitions occur over a large number of frames. Another limitation of this scheme is that it cannot classify the gradual transition.

Several statistical-feature based techniques have also been proposed for gradual transition detection. Alattar used quadratic behaviour of the variance curve to detect fading. This algorithm can only detect fade-in and fade-out where the end frames are fixed. When the sequence has considerable motion, this algorithm fails to identify fade-in and fade-out regions. Alattar has also proposed a statistically based approach for wipe detection. This scheme is very sensitive to the type of the video sequence as the proposed algorithm uses a crude approximation for the mean and variance curves.

The authors have considered an algorithm using the ratio of the incremental change in the mean of the luminance signal to the incremental change in the mean of the chrominance signal (the average sum of $C_1$ and $C_2$) as the criterion for identifying fading transitions. This algorithm may fail to identify fade regions when the solid colour is very close to the mean of the original sequence (before fading is applied).

Most of the above techniques have only been proposed for gradual scene change detection in uncompressed video. However, as already explained, identification of shots in the compressed domain is also important but this remains an unsolved problem.
In this section we focus on the identification of fading and dissolving using statistical features in the uncompressed domain. In video editing and production, proportions or two or more picture signals are simply added together so that the two pictures appear to merge on the output screen. Very often this process is used to move on from picture A to picture B. In this case, the proportions of the two signals are such that as the contribution of picture A changes from 100% to zero, the contribution of picture B changes from zero to 100%. This is called dissolving. When picture A is a solid colour, the process is called fade-in, and when picture B is a solid colour it is known as fade-out. Mathematically, dissolving can be expressed as follows:

\[ S_c(x, y) = \left\{ \begin{array}{ll}
  f_s(x, y), & 0 \leq n < L_s \\
  \left[ 1 - \left( \frac{n - L_s}{F} \right) \right] f_s(x, y) + \left( \frac{n - L_s}{F} \right) g_s(x, y), & L_s \leq n \leq (L_s + F) \\
  g_s(x, y), & (L_s + F) < n \leq L
\end{array} \right. \]

where \( S_c(x, y) \) is the resultant video signal, \( f_s(x, y) \) is picture A, \( g_s(x, y) \) is picture B, \( L_s \) is the length of the sequence of picture A alone, \( F \) is the length of the dissolving sequence, and \( L \) is the length of the total sequence.

It can be proved from eqn. 7 that during fading/dissolving, the mean \( (m) \) and variance \( (\sigma) \) have a linear and a quadratic behaviour, respectively, as shown in eqns. 8 and 9:

\[ m_n = E[S_c(x, y)] = \left\{ \begin{array}{ll}
  m_s, & 0 \leq n < L_s \\
  m_s - \frac{L_s}{F} (m_r - m_s) - \frac{n}{F} (m_r - m_s), & L_s \leq n \leq (L_s + F) \\
  m_r, & (L_s + F) < n \leq L
\end{array} \right. \]  

\[ \sigma^2_n = E[S_c^2(x, y)] - E[S_c(x, y)]^2 = \left\{ \begin{array}{ll}
  \sigma^2_r, & 0 \leq n < L_s \\
  \phi, & L_s \leq n \leq (L_s + F) \\
  \sigma^2_r, & (L_s + F) < n \leq L
\end{array} \right. \]

where

\[ \phi = \xi^2 \left[ \frac{2\sigma^2_r + 2L_r \sigma^2_r}{L} \right] n + \left( \sigma^2_r + L_r \sigma^2_r \right) \xi + \left( \sigma^2_r + \frac{\sigma^2_r}{L} \right) \]

These mathematical derivations are valid under the assumption that the video sequences are an ergodic process. In practice, an ergodic process cannot always be guaranteed due to motion in the video. Therefore, these statistical behaviours may be slightly distorted for a practical video sequence. Alternative strategies are needed besides the mean and variance of the video sequences in order to identify these special effects.

Since it is not possible to identify these special effects accurately by considering only the mean or the variance individually, we have proposed a scheme in which both these features are combined. Since the mean has a linear behaviour its first derivative should be constant during the dissolving period. The second derivative of the variance should also be a constant as the variance curve has a quadratic behaviour during a dissolve period. Therefore, the ratio of the second derivative of the variance curve to the first derivative of the mean curve (the test ratio) should be a constant. This is used as the criterion for identifying a dissolve. This condition may also be satisfied for a small number of consecutive frames in a non-dissolve sequence. Short dissolve sequences are not, however, common in practice and this argument is used in our proposed algorithm to eliminate false regions. Furthermore, if there are two consecutive dissolve regions separated by a very small gap, they are bridged to form a longer dissolve region.

The above argument can be extended to detect fade-ins and fade-outs since fading is a special case of dissolving in which one scene is a solid colour. Note that the variance is zero at the start of a fade-in sequence and at the end of a fade-out sequence as all the pixels have the same value, namely that of the solid colour. Thus, fade-in, fade-out and dissolving are identified as follows:

- **Fade-in:** Detect a frame with zero variance followed by a continuous sequence during which the test ratio is lower than the dissolve threshold \( T_{\text{dissolve}} \).
- **Fade-out:** Detect a continuous sequence during which the test ratio is lower than \( T_{\text{dissolve}} \) followed by a frame with zero variance.
- **Dissolving:** Detect a continuous sequence during which the test ratio is lower than \( T_{\text{dissolve}} \).

<table>
<thead>
<tr>
<th>Actual fade region</th>
<th>Detected fade region</th>
<th>Nature of the region</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-60</td>
<td>31-60</td>
<td>fade-in</td>
</tr>
<tr>
<td>111-150</td>
<td>111-150</td>
<td>fade-out</td>
</tr>
<tr>
<td>248-303</td>
<td>249-303</td>
<td>fade-out</td>
</tr>
<tr>
<td>576-624</td>
<td>576-625</td>
<td>fade-in</td>
</tr>
<tr>
<td>754-778</td>
<td>754-778</td>
<td>fade-out</td>
</tr>
<tr>
<td>944-986</td>
<td>944-987</td>
<td>fade-in</td>
</tr>
<tr>
<td>1102-1167</td>
<td>1102-1168</td>
<td>fade-in</td>
</tr>
<tr>
<td>1365-1420</td>
<td>1366-1420</td>
<td>fade-out</td>
</tr>
<tr>
<td>1500-1550</td>
<td>1500-1550</td>
<td>fade-in</td>
</tr>
<tr>
<td>1620-1680</td>
<td>1620-1681</td>
<td>fade-in</td>
</tr>
<tr>
<td>1760-1840</td>
<td>1761-1840</td>
<td>fade-out</td>
</tr>
<tr>
<td>1920-1985</td>
<td>1920-1985</td>
<td>fade-out</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Actual dissolve region</th>
<th>Detected dissolve region</th>
</tr>
</thead>
<tbody>
<tr>
<td>31-60</td>
<td>31-60</td>
</tr>
<tr>
<td>121-180</td>
<td>121-180</td>
</tr>
<tr>
<td>221-280</td>
<td>221-280</td>
</tr>
<tr>
<td>325-385</td>
<td>324-385</td>
</tr>
<tr>
<td>446-496</td>
<td>446-497</td>
</tr>
<tr>
<td>548-604</td>
<td>548-604</td>
</tr>
<tr>
<td>804-868</td>
<td>804-869</td>
</tr>
<tr>
<td>1010-1089</td>
<td>1010-1089</td>
</tr>
<tr>
<td>1168-1232</td>
<td>1169-1232</td>
</tr>
<tr>
<td>1356-1424</td>
<td>1356-1424</td>
</tr>
</tbody>
</table>
We shall now illustrate how this algorithm is used to identify fading and dissolving. Two test video sequences were used to assess the performance of the proposed scheme. The first sequence contained 2200 frames and was used for assessing the fade detection algorithm. The second test sequence contained 1500 frames and was used for identifying dissolves. Both sequences contained several other transitions, such as sudden scene changes, wiping, panning and tilting.

Figs. 7 and 8 show the variation of the mean and variance, respectively, for the first 240 frames of the first sequence. This test sequence contained one fade-in and one fade-out as indicated in the figures. The linear and quadratic behaviour of the mean and variance curves, respectively, during the fade region can be clearly seen. Fig. 9 shows the absolute values of the difference between successive values of the test ratio. There are two fade regions—immediately after the 31st frame and after the 111th frame.

From a consideration of the variance of the sequence, we can distinguish between fade-in and fade-out regions as discussed previously: the fade-in region is from frame 31 to frame 60, and the fade-out region from 111 to frame 150. Table 1 summarises the results for the first test sequence with the proposed algorithm. These results show that the algorithm is capable of detecting all fade regions accurately, even when the video sequence contains other special effects.

Figs. 10 and 11 show the linear and quadratic behaviour of the mean and variance curves, respectively, during a dissolve region. Fig. 12 shows the absolute values of the difference between successive values of the test ratio. It can be seen that dissolve regions can easily be identified using the same threshold as for fading and occur over frames 31–60 and 121–180. The full results for the second test sequence are summarised in Table 2.

Once the shot boundaries have been identified, the shots can be distinguished for video indexing. The properties of fading and dissolving can also be used to encode the video more effectively at the encoder.
4 Conclusions

A powerful scene change detection algorithm is required in order to characterise video sequences completely for content-based video indexing and retrieval. In this paper several algorithms for scene change detection in both uncompressed and compressed video have been discussed. We have also presented a real-time algorithm for detecting abrupt scene changes in MPEG-2 compressed video using the number of interpolated macroblocks for a given B-frame. Experimental results show that this algorithm can detect abrupt scene changes irrespective of the nature of the sequences. Furthermore, we have presented an algorithm for fade and dissolve scene change detection in video sequences using the statistical features of each image. Test results show that these special effects can be identified accurately with the proposed scheme.

All of these algorithms may, however, be adversely influenced by large local and global motion in the video sequences. More work is required on the identification of such local and global motions and how they can be compensated for in scene change detection algorithms.

References

Anil Fernando is an assistant professor at Asian Institute of Technology (AIT), Thailand. He received a BSc Engineering degree (first class Honours) in Electronic and Telecommunications Engineering from the University of Moratuwa in 1995 and an MEng degree (Distinction) in telecommunications from AIT in 1997. He recently completed a PhD degree at the Department of Electrical and Electronic Engineering, University of Bristol. His current research interests include scene change detection in uncompressed and compressed video, video editing in compressed video, intelligent video encoding, COFDM for wireless channels, and channel coding and modulation schemes for satellite channels.

Address: Telecommunications Program, School of Advanced Technologies, Asian Institute of Technology, PO Box 4, Klong Luang 12120, Thailand
E-mail: fernando@ait.ac.th

Nishan Canagarajah is a Senior Lecturer in Signal Processing at the University of Bristol. He has BA (Hons) degree and a PhD degree in DSP techniques for speech enhancement, both from the University of Cambridge. He is a member of the Virtual Centre of Excellence in Digital Broadcasting and Multimedia Technology and an Associate Editor of the IEE’s Electronic & Communication Engineering Journal. He is also an editor of a book on Mobile Multimedia Technologies. His research interests include image and video coding, nonlinear filtering techniques and the application of signal processing to audio and medical electronics.

Address: Image Communications Group, University of Bristol, Merchant Venturers Building, Woodland Road, Bristol BS8 1UB, UK
E-mail: nishan.canagarajah@bristol.ac.uk

David Bull is a Professor of Digital Signal Processing and Deputy Director of the Centre for Communications Research at the University of Bristol, where he leads the Image Communications Group. He has worked widely in the fields of 1-D and 2-D signal processing and his current research is focused on the problems of image and video communications for both low-bit-rate and broadcast applications. In particular he is working on error-resilient source coding, linear and nonlinear filter banks, scalable coding methods, motion estimation and architectural optimisation (for filters, transforms and wavelet filter banks).


©IEEE:2001
First received 13th October 1999