IS&T/SPIE Symposium Proceedings Vol. 2419, February 1995, San Jose, California

Scene Change Detection in a MPEG Compressed Video Sequence

Jianhao Meng, Yujen Juan*, Shih-Fu Chang
Department of Electrical Engineering & Center for Telecommunication Research
Columbia University
*Hitachi America, Ltd.

Abstract

An algorithm is proposed for the detection of abrupt scene change and special editing effects such as dissolve in a compressed MPEG/MPEG-2 bitstream with minimal decoding of the bitstream. Scene changes are easily detected with DCT DC coefficients and motion vectors. By performing minimal decoding on the compressed bitstream, the processing speed for searching a video database of compressed image sequences can be dramatically improved. In addition, the algorithm may also be applied in video scene browsing and video indexing as well.

Keywords: video indexing, scene change detection, compressed-domain editing, MPEG compressed streams.

1. Introduction

Information databases have evolved from simple text to multimedia with video, audio, and text. The query mechanism for a video database is similar in concept compared to a textual database. Object searching is analogous to word searching; scene browsing is similar to paragraph searching; video indexing is comparable to text indexing or bookmarking. However, the implementation for video content based searching is very different and much more difficult than the query mechanism for a textual database. Scene change detection technique can often be applied for scene browsing and automatic and intelligent video indexing of video sequences for video databases. Once individual video scenes are identified, we can use content-based indexing mechanisms (such as indexing by object texture, shape, color, motion) to index and query image contents in each video scene[4,5].

Due to the large amount of data, video sequences are often compressed for efficient transmission or storage on-line. However, most of current scene change detection techniques operates on uncompressed video sequence. Therefore, these compressed video sequences have to go through computationally intensive processing steps to be de-compressed first before any scene change detection algorithms can be applied.

Manipulation of compressed video sequences directly in the compressed domain has been proposed in [1,2]. This paper presents an approach to process a compressed digital video bitstream to detect scene changes. In the MPEG standard, frames of video are compressed using motion compensation and discrete cosine transform (DCT). By examining the DCT DC coefficients and the motion vectors in the MPEG bitstream, the algorithm can detect scene changes and special editing effects such as dissolve in the bitstream without fully decompressing the bitstream.

This work is part of the effort of implementing a multimedia server with advanced content-based image query capability in the Image and Advanced TV Lab in Columbia University.

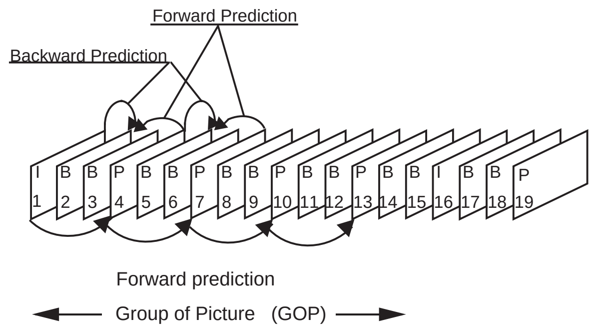


FIGURE 1. A Typical MPEG Frame Sequence in Display

2. Previous Approaches

Most of algorithms used today to detect scene changes are performed on the pixel domain. Nagasaka and Tanaka [3] suggested a few measures to detect scene changes. The simplest measures take the sum of absolute pixel-wise luminance differences between 2 frames of image. This approach can be extended to calculating the sum of absolute pixel-wise color (all three R, G, B components considered) differences for better performance and easier detection. However, all other better approaches suggested in [2] operate on the histograms of the frames. The best approach found was to take the χ^2 test of color histograms between 2 frames. The test is essentially the sum of the squares of the bin-wise color histogram differences normalized to one of the frames used. When a scene change occurs, the color composition between 2 frames is expected to be very different. Hence, the histogram differences between the 2 frames will be large. Scene changes can be detected by examining the histogram differences for peaks. To detect dissolve, Zhang[10] used the *twin comparison* technique to detect the accumulated frame difference during a dissolve region and Alattar[8] used the second order of the frame variance, both performed in pixel domain.

Since the color histogram and the variance are gathered in the pixel domain, it requires a compressed picture to be decompressed first. For the international MPEG standard [6,7], this requires computationally intensive processing steps such as Huffman code decoding, inverse DPCM, inverse quantization, inverse discrete cosine transform (IDCT) and motion compensation to reconstruct an image frame in the pixel domain.

3. MPEG Compression Standard

The MPEG group defines the syntax for a compressed bitstream. MPEG is a defined international standard (ISO 11172) for a compressed video bit-rate of about 1.5 Mbit/sec. MPEG-2 extends the MPEG standard in many ways including higher bit-rate and more powerful and efficient compression algorithms, such as different motion compensation modes. In order to achieve high compression ratio, the MPEG algorithm uses many different techniques to reduce temporal and spatial redundancy in a video sequence.

To reduce temporal redundancy, predictive coding via motion compensation is used. Each 16x16 macroblock consisting of 4 8x8 blocks in a frame is matched as closely as possible to another 16x16 macroblock in a frame in the past (forward motion prediction) or a frame in the past or the future (bi-directional motion prediction). Once a matching block is found, a motion vector indicating the type and the distance is sent in the bitstream. If no good matching block can be found, the macroblock can simply be intra-coded without any motion compensation. The blocks of residual error after motion compensation and the blocks without motion compensation are further processed with DCT using 8x8 blocks. The DCT coefficients are quantized and run-length encoded to further compress the information.

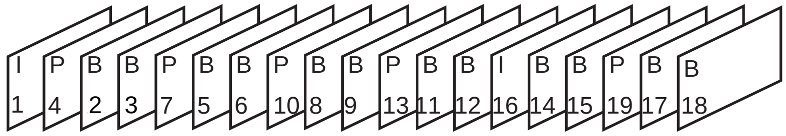


FIGURE 2. A MPEG Frame Sequence in Transmit or Storage

When the video sequence is compressed, the frames are compressed in groups (GOP, group of pictures). Within each group, the first frame is an I-frame. An I-frame is completely intra-coded, i.e. no motion compensation is performed when the frame is compressed. A P-frame is predictively coded with motion compensation only from past I- or P- frames. Together, I- and P- frames are called anchor frames. They are used as basis for motion compensation in bi-directionally coded B-frames. A typical sequence of frames would be: IBBPBBPBBPBBPBBIBBP... and so on. Figure 1 illustrates a typical MPEG sequence of frames with a GOP size of 15 frames and 2 B-frames between anchor frames

Since a B-frame requires the 2 closest anchor frames around it to be decoded, the sequence of frames would be stored (or transmitted) as: IPBBPBBPBBPBBIBBP... and so on, as shown in Figure 2. The transmit order is different from the display order in that the anchor frames will be received before the B-frames that are to be displayed before the anchor frames to be displayed.

4. Direct Scene Change Detection in a MPEG bitstream

The following scene change detection algorithm is formed based on the statistical characteristics of the DCT DC values and motion vectors in MPEG bitstream. The algorithm detects suspected scene change on I, P, B-frames separately first, then a final decision will be made to select the true scene changes.

4.1 Detection Algorithm for Anchor Frames

Since the anchor frames are received (or accessed) out of order (before the B-frames that are displayed first), one may have to examine frames in a later transmitted order but earlier in display order to determine the location of the scene change. For example, if a scene change occurs on a B-frame (frame 5 in Figure 1), then it is very likely that the first anchor frame following the scene change (frame 7 in Figure 1) will have the characteristics of an anchor frame with a scene change. However, this anchor frame will be accessed first in the compressed bitstream. The decision on whether the scene change occurs on the anchor frame should be delayed until all of the frames in earlier display order (frames 5 and 6) have been received and examined.

In addition, if a scene change occurs on a B-frame (say frame 5), then it is very likely that all B-frames following the B-frame after the scene change (frame 6) will have very similar characteristics of a B-frame with a scene change. These B frames should not be identified as locations of the scene change. This eliminate the possibility of detecting more than 1 scene changes in a series of B-frames. However, in practical encoding, the number of B-frames between anchor frames are usually kept small (2 or 3) for picture quality considerations. In addition, practical video sequences generally do not have multiple scene changes within such a small frame spacing. Therefore, this kind of situations are very rare. Even if this occurs, examinations of subsequent frames will reveal a scene change as well. Even though the scene change may be detected, in this instance, on the wrong frame, it may still be acceptable from a practical application point of view.

Similarly, when a scene change occurs between 2 consecutive I-frames, the later I-frame will likely meet the criteria for a scene change. This second I-frame should also be rejected as a scene change location. This eliminates the possibility of detecting more than 1 scene change between 2 consecutive I-frames with the last scene change occurring on the second I-frame. Again, the likelihood of this situation is small with a typical GOP size of 12 or 15 (approximately 0.4 to 0.5 seconds in time when the frame rate is 30 frames/sec) and the scene change may still be detected subsequently on the wrong frame.

4.1.1 Detection Algorithm for I-frames

When a scene change occurs on an I-frame, there is very little encoded data (such as residual errors or motion vectors) that could indicate this since an I-frame is completely intra-coded. Therefore, the previous I-frame is needed to determine whether a scene change occurs. When a scene change occurs on an I-frame, the picture color, brightness would be quite different from the previous scene sequence. In the pixel domain, the color histogram will be ideal to indicate such difference. In the frequency (DCT) domain, the DCT DC coefficients will contain such information since the DC coefficient is essentially the average of all pixel values within the 8x8 block. To detect a scene change that occurs on an I-frame, the color histograms (the luminance and the 2 chrominance components, i.e.YUV since the MPEG standard processes pixels in this color space) of 2 consecutive I-frames are compared. Each component of the color histogram is gathered from the DC coefficients of 8x8 DCT blocks of the corresponding component.

The absolute differences between the histogram bins, *H(f,i)*, are summed and normalized by the histogram bin size, *Hist*. This normalized sum is squared and summed together among the three *y*, *u*, *v* components. The normalized sum is divided by the normalized sum of the previous I-frame to get *factor.* A suspected scene change is declared by three thresholds based on the high, medium and low *factor* values.

$$Sum_y = \sum_{i=0}^{Hist-1} \left| H_y(f_1, i) - H_y(f_2, i) \right|$$

$$Sum_{f1} = \frac{(Sum_y + Sum_u + Sum_v)^2}{Hist} \tag{1}$$

The difficulty of this approach was to set up the right thresholds. Different image size and encoding bit rate would affect the threshold. A new approach with more relaxed constraint on thresholds is found when experimenting with the dissolve sequences. It is based on the fact that the intensity variance of frames within the same scene tends to remain stable at a level, although high-motion with drastically changing intensity would drive it up and down. A scene change is often indicated by a peak of $|\Delta\sigma^2|$, the absolute value of the frame variance difference. But this method alone is not accurate with the high-motion scenes. Therefore, we add a new criterion to solve this problem. When a scene change happens on an I-frame, the B-frames right before it (in display order) would have most of the motion vectors pointing forward, and this is true even for high-motion sequences. Now the ratio R_f, the number of forward motion vectors to the number of backward motion vectors in each B-frame is used in detecting scene changes on I-frames later. The peaks of R_f are detected by the local window method (see 4.3). An I-frames is marked as a suspected scene change only when a peak $|\Delta\sigma^2|$ happens and the immediate B frame before it has a R_f peak.

4.1.2 Detection Algorithm for P-frames

P-frames are predictively coded (from previous I or P frames) with only forward motion compensation. In addition, P-frames are always accessed in display order. When a scene change occurs on a P-frame, the encoder could not use many macroblocks from the previous anchor frame to perform motion compensation and therefore a large amount of macroblocks will be intra-coded without any motion compensation.

To detect a scene change that occurs on a P-frame, R_p the ratio of the number of macroblocks without motion compensations to the number of macroblocks with motion compensation is computed. A suspected scene change is declared if there is a peak (see Figure 3a).

It is also possible to use the residual error of the P-frame after motion compensation to determine scene changes. This information is carried by the DCT coefficients, especially the DC coefficients. However, this mechanism is usually not very reliable to determine scene changes especially when there is very fast motion in the video sequence.

4.2 Detection Algorithm for B-frames

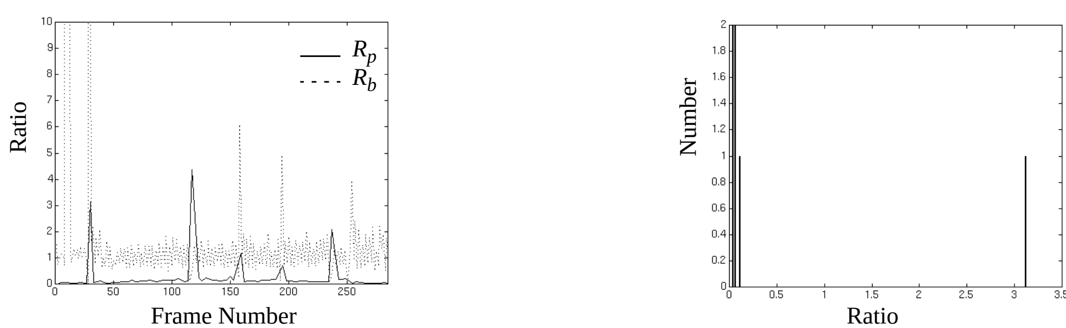
When a scene change occurs on a B-frame (for example, frame 5 in Figure 1), most of the motion vectors will come from the anchor frame in a later display order (frame 7) and very few motion vectors will come from the anchor frame in an earlier display order (frame 4).

To detect a scene change on B frames, R_b the ratio of the number of backward motion vectors to forward motion vectors is computed. Two situations are found to be typical when a scene change occurs on a B-frame. First, the ratio is very high (>> 10) with a lot of macroblocks motion compensated. Second, the ratio is lower (< 10) with few macroblocks motion compensated, but an examination of anchor frames received earlier showed a possible scene change. In stead of setting two global thresholds for above situations, the adaptive local window method is again used to detect scene change peak points.

4.3 The Adaptive Local Window Threshold Setting Technique.

Different scenes have very different motion vector ratios. But within the same scene, they tend to be similar. Setting several levels of global thresholds will not only complicate the process but also cause false alarms and false dismissals. A local adaptive threshold technique is used to overcome this problem.

The peak ratios R_b, R_p and R_f are detected separately. All ratios are first clipped to a constant upper limit, usually 10 to 20. Then ratio samples are segmented into windows. Each window is 2 to 4 times of the Group of Picture(GOP) size. For typical movie sequences, scene change distances are mostly greater than 24, so a window size of 2 GOP is sufficient. If GOP is 12, and window size is 2, there will be 6 ratio samples for P-frames and 12 samples for B-frame. These samples are enough to detect the peaks.



(a) R_b, R_p in bmix4.mpg
(b) Histogram of a local window

FIGURE 3. Motion Vector Ratio In B and P Frames

Within the window, histogram of the samples with a bin size of 256 is calculated. If the peak-to-average ratio is greater than the threshold, T_d, then the peak frame is declared as a suspected scene change. The peak values are not included in calculating the average. A typical T_d is 3 for R_b, R_p and R_f. Figure 3b. shows the histogram of a local window corresponding to P-frames (from frame 24 to 47) in Figure 3a. where there is a peak at frame 29

For B-frames, if a scene change happens at a B-frame (frame 10, in Figure 3a.), then the ratio of the immediately subsequent B frame (frame 11) will be high also. Both of them will be considered as peaks and will not be calculated into the average, only the first B-frame will be marked as a suspected scene change.

5. Detection of Special Editing Effects: Dissolve

Dissolve is the most frequently used editing technique to connect two scenes together. A dissolve region is created by linearly mixing two scenes sequences: one gradually decreasing in intensity, one gradually increasing. A dissolve usually lasts from 30 to 60 frames in a typical movie. The algorithm described above cannot detect the dissolve region due to the similarity of the adjacent frames in that region. We follow Alattar's approach[8] to use variance as the indication for detecting dissolve. However, we measure frame variance directly in the compressed domain by using the DCT DC coefficients of I and P frames.

5.1 Pixel Domain Dissolve Region Characteristics

Assume $f_1(t)$ and $f_2(t)$ are two ergotic sequences with intensity variance σ_1^2 and σ_2^2. Let the intensity of $f_1(t)$ linearly decrease to 0: fade-out, and $f_2(t)$ linearly increase from 0 to normal: fade-in. The dissolve region is the sum of $f_1(t)$ and $f_2(t)$ during the dissolve ($t_1 \le t \le t_2$, t_1: start, t_2: end).

$$f(t) = f_1(t)[1-\alpha(t)]+f_2(t)\alpha(t) \tag{2}$$

where $\alpha(t)$ is $(t-t_1)/(t_2 - t_1)$ increases from 0 to 100%. The variance of $f(t)$ in the dissolve region is:

$$\sigma^2(t) = (\sigma_1^2 + \sigma_2^2)\alpha^2(t) - 2\sigma_1^2\alpha(t) + \sigma_1^2 \tag{3}$$

This is the ideal case when $f_1(t)$ and $f_2(t)$ are ergotic with σ_1^2 and σ_2^2, see Figure 4. The variance curve in the dissolve region shows a clear parabolic shape. In real sequences with motions, the variance of $f_1(t)$ and $f_2(t)$ will not be constant, but the dissolve region still demonstrates the parabolic shape as in the ideal case. The variance curve for sequence with fade-out followed by a immediate fade-in is similar to the dissolve case except the valley goes down deeper to almost 0.

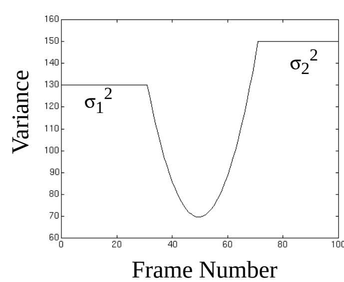


FIGURE 4. An Ideal Dissolve Curve

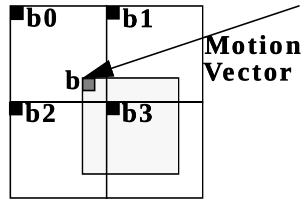


FIGURE 5. Inverse Motion Compensation of DCT DC

5.2 Dissolve Detection in a MPEG2 Bitstream

Based on the characteristics of the dissolve region, the detection can be achieved by taking the pixel domain intensity variance of each frame and then detecting the parabolic curve[8]. Given an MPEG bitstream, the variance of the DCT DC coefficients in I and P frames is calculated in stead of the spatial domain variance. Since the DCT DC is the average pixel values of an 8x8 block, the approximation error term is:

$$E[error] = E[\sigma_p^2 - \sigma_{dc}^2] = E[E[X_p^2] - E[X_{dc}^2]] \tag{4}$$

where σ_p^2 is the pixel domain variance, σ_{dc}^2 is the DCT DC variance, $E[X_p^2]$ and $E[X_{dc}^2]$ are the second moment of the pixel intensity and DCT DC respectively. If the block size is not too large, the above error term can be assumed to be small as well. Experiments show that the variance of DCT DC values is accurate enough to detect dissolve.

5.2.1 Inverse Motion Compensation of DCT DC in P Pictures

The I-frames in MPEG are all intra coded, so the DCT DC value can be obtained directly. The P-frames consist of motion compensated(MC) macroblocks and the intra coded macroblocks. An MC macroblock has a motion vector and a DCT coded MC error. Each macroblock consists of 4 luminance blocks(8x8 each) and some chrominance blocks(2, 6 or 8 based on chrominance format). To ensure maximum performance, only the DCT DC of luminance block are used for dissolve detection. The DCT DC values of B pictures are not reconstructed due to implementation complexity, although the following technique can still be applied. Since a typical dissolve would last 30 to 60 frames, using a typical IBBPBB coding format, there will be 10 to 20 I or P frames, which are sufficient for detecting a dissolve.

To get the DCT DC values of a P frame, inverse motion compensation is applied on the luminance blocks and the DCT DC value of the error term is added to the reconstructed DCT DC. Assume the variance within each block is small enough, then *b*, the DCT DC of a MC block in P frame can be approximated by taking the area-weighted average of the four blocks in the previous frame pointed by the motion vector.

$$b = [b_0*(8-x)*(8-y) + b_1*x*(8-y) + b_2*(8-x)*y + b_3*x*y]/64 + b_{error\_DCT\_DC} \tag{5}$$

where x, y are horizontal and vertical motion vector modulo block size 8; b_0, b_1, b_2 and b_3 are the DCT DC coefficients of the four neighboring blocks pointed by the motion vector; $b_{error\_DCT\_DC}$ is the DCT DC of the motion compensation residue error of block to be reconstructed (see Figure 5).

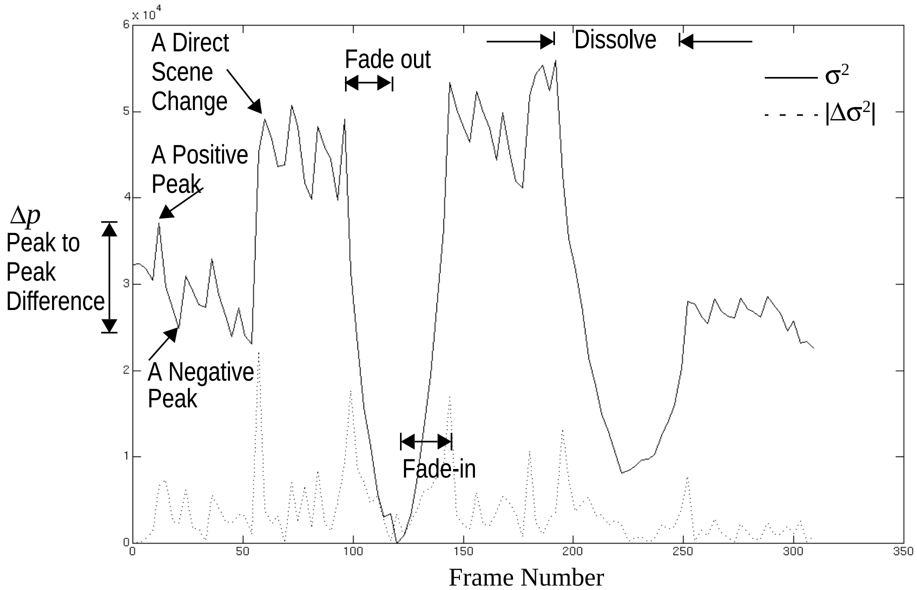


FIGURE 6. Variance σ^2 and $|\Delta\sigma^2|$ in bmix2.mpg

5.2.2 Dissolve Detection

After a minimal partial inverse motion compensation, we use two criteria--first, the depth of the variance valley must be large enough and second, the duration of the suspected dissolve region must be long enough (otherwise it's more likely an abrupt scene change). The specific procedure is as following.

All the positive-peaks, p^+ are detected by using the local window method on $\Delta\sigma^2$, the frame variance difference; all the negative-peaks p^- are found by the minimum value between the two positive-peaks; the peak-to-peak difference Δp = *current peak* - *previous peak,* is calculated and thresholded using the proposed local window method; finally potential matches with duration length (positive peak to current negative peak) long enough (> T_f frames, e.g. 1/3 of the minimum allowed dissolve duration) are declared as suspected dissolves.

The starting point of the suspected dissolve is the previous positive peak. If the next positive peak is at least T_f frames from the current negative peak, then a dissolve is declared and the ending point is set to the next positive peak. Frames whose peak-to-peak distance meets the magnitude threshold, but fail to meet the duration threshold are usually the direct scene changes. Similarly, if the frame duration from the current negative peak to the next positive peak fails to meet T_f, the suspected dissolve will be unmarked also.

6. Scene Change Detection Algorithm Summary

The overall scene change detection algorithm has five stages: minimal decoding, parsing, statistical, detection and decision stages. Figure 7. shows the function blocks of each stage.

6.1 Scene Change Detection Function Description

Minimal Decoding Stage: MPEG bitstream is decoded just enough to obtain motion vectors and the DCT DCs.

Parsing Stage: Motion Vectors in B, P frames are counted; DCT DCs in P frames are reconstructed.

Statistical Stage:
1. Compute R_p, the ratio of intra coded blocks and forward motion vectors in P-frames.
2. Compute R_b, the ratio of backward and forward motion vectors in B-frames.
3. Compute R_f, the ratio of forward and backward motion vectors in B-frames.
4. Compute the variance of DCT DC of luminance in I and P frames.

Detection Stage:
1. Detect R_p peaks in P frames and mark them as suspected scene change frames.
2. Detect R_b peaks in B frames and mark them as suspected.
3. Detect R_f peaks in B frames. Detect all $|\Delta\sigma^2|$ in I, P frames, the absolute value of variance frame difference. Mark I-frames as suspected if they have $|\Delta\sigma^2|$ peaks and if the B-frames in front them have R_f peaks.
4. Detect the parabolic variance curve for dissolve.

Decision Stage:
1. All suspected frames which fall in the dissolve region are unmarked.
2. Exam through all marked frames from the lowest frames number
   if current marked frame number - last scene change number > $T_{rejection}$
   then current marked frame is a true scene change
   else unmark current frame
   (where $T_{rejection}$ is the rejection threshold, default one GOP.)

The criterion in step 2 of Decision Stage is used to eliminate the situations discussed in 4.1, i.e., when a scene change happens on a B frame, then the immediately subsequent P-frame and/or I-frame (of display order) will be likely marked as suspected one as well. But since they don't satisfy the criterion of "the minimum distance between two scene changes has to be greater than $T_{rejection}$," these suspected frames will be unmarked.

For P frames, the marked frame decision can be obtained both from step 1 (R_p peaks) and step 3 ($|\Delta\sigma^2|$ peaks) in the Detection Stage. The outcome from step 1 is usually more reliable. The outcome from step 3 can be used as reference if they are conflicting with the outcome of step 1.

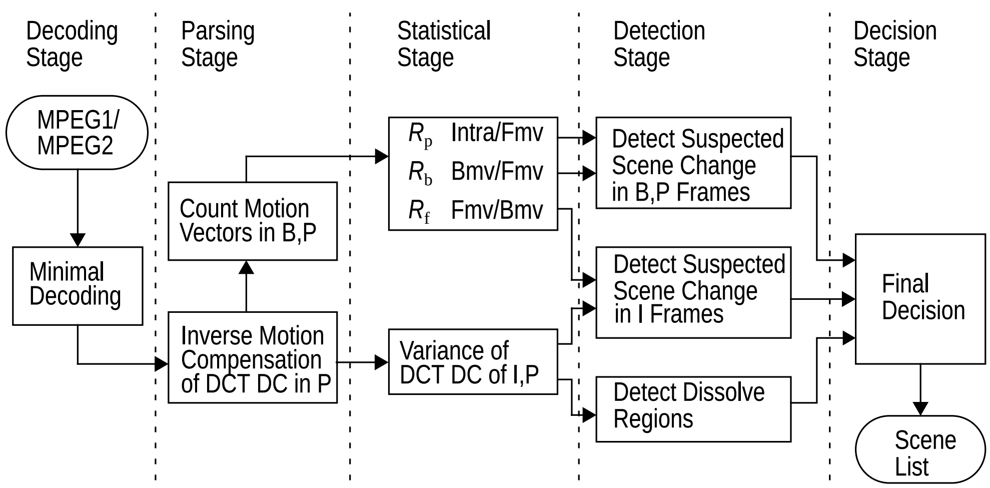


FIGURE 7. The Function Blocks of the Scene Change Detection Algorithm

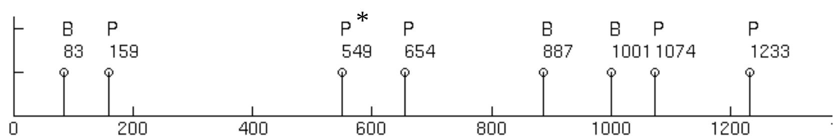
7. Applications of the Algorithm

Once a scene change is detected, the bitstream can then be fully decoded from the frame of the scene change and displayed. It may be necessary to back up a few frames in the bitstream to the beginning of the GOP and decode all prior anchor frames in the GOP in order to fully decode the frame of interest for display.

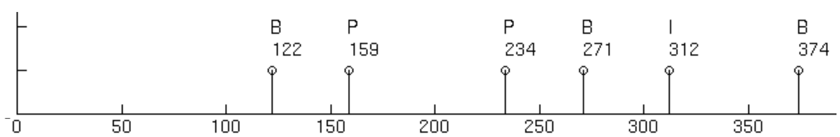
It is also conceivable that the video sequence can be automatically indexed during the compression process using this algorithm to ease the burden of the decoder. The algorithm described above may be simpler and more computationally efficient than the pixel domain algorithm since only DC DCT coefficients are needed for the I-frames and motion vector types are needed for the P- and B- frames.

8. Results

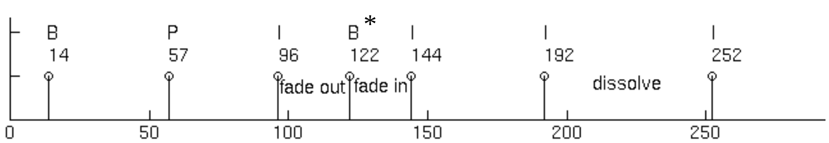
The algorithm is implemented based on Columbia Unversity's MPEG2 codec in C language. It has been tested on several video sequences from movies. Three test sequences 'bmix2.mpg', 'bmix4.mpg' and 'b15c.mpg' have image size of 608x224(letterbox movie) and coded at a bit rate of 1.5 Mbits/s, which is equivalent to VHS quality. One sequence 'b52c.mpg' has image size of 320x128 (sub-sample of the original 608x224) and coded at a bit rate of 0.6 Mbits/s. They are all coded in 4:2:0 format with GOP set to 12. Among the four test videos, 'bmix2.mpg' has one fade-out/fade-in and one dissolve; all other three videos have only abrupt scene changes.



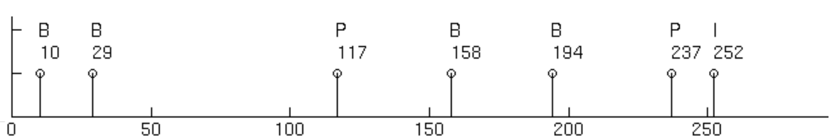
(a) b15c.mpg (abrupt scene change)



(b) b52c.mpg (abrupt scene change)



(c) bmix2.mpg (fade out/in, dissolve)



(d) bmix4.mpg (abrupt scene change)

FIGURE 8. Detection Results of Four MPEG2 Test Sequences

The window size for all tests is set to 2 (GOPs). With threshold T_d equals to 3 for detecting peaks, the algorithm detected all the abrupt scene changes. None of the I, B, P frames were missed, but there were one false alarm in ‘b15c.mpg’ on frame 549. When T_d is set to 4, this false alarm is gone. Also there were one false alarm in ‘bmix2.mpg’ on frame 122, which was actually during the fading period and can be resolved at the decision stage.

For the dissolve detection, the algorithm can not differentiate dissolve and fade-out/fade-in since the two cases are extremely similar. The actual fading in ‘bmix2.mpg’ is from frame 96-143, the algorithm detected 96-144, because frame 143 is a B-frame and this algorithm only runs on I and P frames for dissolve detection. The actual dissolve in ‘bmix2.mpg’ is from 192-247, the algorithm detected 192-252 as dissolve region. This is because frame 247(B-frame) is the 8th frame in the GOP and the next frame 249(P-frame) was reconstructed from the previous IPP frames and the accuracy of this third time reconstruction is low due to the accumulated inverse motion compensation errors. Therefore, frame 252, the next I-frame is detected.

9. The Video Browser User Interface

An interactive user-friendly interface for this algorithm has been developed in MOTIF, see Figure 9. User can open an MPEG2 video stream and run the scene change detection algorithm. A minimal set of variables can be set by the user, such as the local window size and the threshold for detecting the peak and the minimum allowed frame distance between scene changes. After running the algorithm, a scene list is extracted and each scene is represented by a pixmap icon. By clicking on the scene icon, user can view the MPEG2 video stream for this particular scene. A video control panel is supplied for user to perform VCR functions such as play, step forward, fast forward and fast reverse on the video stream. User can also use the video panel to access any random frame by click in the frame number and “Search”.

The scene list returned by the algorithm will be become the default index for the video. User can add or delete scenes from the default scene list. A few keyword descriptions such as: color, object and motion can also be added to each scene for simple keyword base video query. A content base query is being developed, which will allow users to search desired image frames based on image contents (color, texture and shape)[9].

10. Conclusion

This paper presented an algorithm of detecting scene changes and special effects such as dissolve with minimal decoding of a MPEG/MPEG2 bitstream. Compared to the traditional approach, the detection is performed with only a partial decoding of the compressed bitstream: minimal decoding for the DCT DC coefficients for the I and P frames and motion vectors for the P and B frames. A full decode of the compressed bitstream is not necessary and therefore computation time can be saved.

FIGURE 9. The Video Browser User Interface with Scene Change Detection

11. References.

1. S.-F. Chang and D.G. Messerschmitt, “Manipulation and Compositing of MC-DCT Compressed Video,” to appear in IEEE Journal of Selected Areas in Communications, Special Issue on Intelligent Signal Processing, 1994.
2. F. Arman, A. Hsu, and M-Y. Chiu, “Image Processing on Compressed Data for Large Video Databases,” Proceedings of ACM Multimedia ’93, June 1993, pp. 267-272.
3. A. Nagasaka and Y. Tanaka, “Automatic Video Indexing and Full-Video Search for Object Appearances” In E. Knuth and L. M. Wegner, editors, *Video Database Systems, II*, Elsevier Science Publishers B.V., North-Holland, 1992, pp. 113 - 127.
4. R. Barber, W. Equitz, M. Flickner, W. Niblack, D. Petrovic, P. Yanker, “Efficient Query by Image Content for Very Large Image Database,” COMPCON ’93, San Francisco, CA., 1993, pp. 17-19.
5. J.R. Smith and S.-F. Chang, “Quad-Tree Segmentation for Texture-Based Image Query,” ACM 2nd Multimedia Conference Proceedings, 1994.
6. ISO/IEC 11172 - 2 Committee Draft (MPEG)
7. ISO/IEC 13818 - 2 Committee Draft (MPEG-2)
8. A. M. Alattar, “Detecting and Compressing Dissolve Regions in Video Sequences with DVI Multimedia Image Compression Algorithm,” *ISCAS* 1993, pp. 13-16.
9. S.-F. Chang, J.R. Smith, and H. Wang, “Feature Extration & Indexing for content-Based Visual Query,” submitted to IEEE *Computer Magazine*, 1995.
10. H.J. Zhang, A. Kankanhalli, S.W. Smoliar, “Automatic Partitioning of Full-motion Video,” *Multimedia Systems* 1993 1(1), pp. 10-28, ACM-Springer.