Integrated Parsing of Compressed Video

Suchendra M. Bhandarkar[1] Yashodhan S. Warke[1] Aparna A. Khombhadia[2]

[1] Department of Computer Science, The University of Georgia, Athens, Georgia 30602–7404, USA
[2] Netscape Communications Corporation, 501 E. Middlefield Rd, Bldg 14, MV-093, Mountain View, California 94043–4042, USA

Abstract. A technique for detecting scene changes in compressed video streams is proposed which combines multiple modes of information. The proposed technique directly exploits and combines the luminance, chrominance, motion compensation information and the prediction error signal in an MPEG1-coded video stream. By performing minimal decoding of the compressed video stream, the proposed technique results in significant savings in terms of execution time and memory usage. The technique is capable of detecting abrupt scene changes (cuts), gradual scene changes (dissolves), and dominant camera motion in the form of pans and zooms in an MPEG1-coded video stream. Experimental results show that combining multiple modes of information is more effective in detecting cuts and dissolves. The proposed technique is capable of processing video frames in real time and could be used for the rapid generation of key frames for real-time browsing of video streams and for indexing to support content-based access to video libraries.

1 Introduction

Video parsing or scene change detection is typically used to extract key frames from a video stream. These key frames can then used for rapid video browsing and automatic annotation and indexing of video streams to support content-based query access to a video database. The video parsing operation is primarily domain-independent and therefore is a crucial first step that precedes domain-dependent analysis of the video [9].

Video parsing techniques that operate upon compressed video directly have a considerable advantage in terms of execution time and memory usage. Techniques that directly parse MPEG1-coded video typically use the histograms of the DC images derived from the DCT coefficients of the I frames [6], the variance of the DC coefficients in the I and P frames [2] or the proportion of macroblocks with valid motion vectors in the P and B frames [4, 9] to detect abrupt scene changes. Yeo and Liu [8] present algorithms for detecting abrupt and gradual scene changes, intrashot variations and flashlight scenes in MPEG1-coded and MJPEG-coded video data using DC images. Bhandarkar and Khombadia [1] present a technique for the detection of abrupt and gradual scene changes that exploits motion compensation information and the prediction error in the MPEG1-coded video stream.

However, video parsing techniques that rely on only one source or mode of information are seen to suffer from certain significant shortcomings. For example, techniques that rely only on chrominance and/or luminance values (or their average values in the DC images), are prone to misses when there is little change in background color or luminance between successive video shots and to false positives when there is a change in background color or background luminance due to change in ambient lighting within a single shot. The relative motion of objects between successive frames is more effective in detecting such scene changes. Moreover, certain types of scene changes such as pans, zooms etc. cannot be detected reliably with luminance and/or chrominance information alone. Using motion information in isolation also has its limitations. In MPEG1-coded video, I frames do not contain motion compensation information. Consequently, scene changes occurring at I frames cannot be detected if the video parsing technique relies solely on motion compensation information. The same is true of MJPEG video which consists solely of I frames.

In this paper, we present a novel and efficient approach to video parsing that integrates luminance, chrominance and motion information, all of which are computed directly from an MPEG1-coded (compressed) video stream. It is shown that by appropriately combining luminance, chrominance and motion information, one can design a more accurate and robust video parsing technique. Since the proposed technique entails minimal decompression of the compressed video, it is capable of parsing video in real-time (i.e., at rates in excess of 30 frames/sec).

2 Description of MPEG1 Video

MPEG1 video compression [3] relies on two basic techniques: block-based motion compensation for reduction of temporal redundancy and Discrete Cosine Transform (DCT)-based compression for the reduction of spatial redundancy. The motion information is computed using 16×16 pixel blocks (called macroblocks) and is transmitted with the spatial information. The MPEG1 video stream consists of three types of frames: intra-coded (I) frames, predictive-coded (P) frames and bidirectionally-coded (B) frames. I frames use only DCT-based compression with no motion compensation. P frames use previous I frames for motion encoding whereas B frames may use both previous and future I or P frames for motion encoding.

An I frame in MPEG1-coded video is decomposed into 8×8 pixel blocks and the DCT computed for each block. The DC term of the block DCT, termed as $DCT(0,0)$ is given by $\frac{1}{8}\sum_{i=0}^{7}\sum_{j=0}^{7} I_B(i,j)$ where I_B is the image block. Operations of a global nature performed on the original image can be performed on the DC image [6, 8, 9] without significant deterioration in the final results.

For P and B frames in MPEG1-coded video, motion vectors (MVs) are defined for each 16×16 pixel region of the image, called a *macroblock*. P frames have macroblocks that are *motion compensated* with respect to an I frame in the immediate past. These macroblocks are deemed to have a *forward predicted* MV

(FPMV). B frames have macroblocks that are motion compensated with respect to, either a reference frame in the immediate past, immediate future or both. Such macroblocks are said to have an FPMV, *backward predicted* MV (BPMV) or both, respectively. The prediction error signal is compressed using the DCT and transmitted in the form of 16×16 pixel macroblocks.

3 Scene Change Detection in MPEG1-coded Video

Our technique is currently designed to detect two important types of scene changes in MPEG1-coded video streams: *cuts* and *dissolves*. A *cut* in a video stream is characterized by an abrupt scene change. A *dissolve* in a video stream is characterized by a gradual scene transition in which the present (outgoing) scene gradually fades out while the next (incoming) scene gradually fades in. Our technique is also capable of detecting dominant camera motion during zooms and pans and classifying the corresponding scenes as such.

3.1 Detection of Cuts Using DC Images

Let X_i, $i = 1, 2, \ldots, N$ be a sequence of DC images. The difference sequence D_i, $i = 1, 2, \ldots, N-1$, is generated where $D_i = d(X_i, X_{i+1})$. Each element of D_i is the difference of the cumulative DC values of two successive images. A scene cut is deemed to occur between frames X_l and X_{l+1} if and only if
(1) The difference D_l is the maximum within a symmetric sliding window of size $2m-1$, i.e. $D_l \geq D_j$, $j = l-m+1, \ldots, l-1, l+1, \ldots, l+m-1$, and
(2) D_l is at least n times the magnitude of the second largest maximum in the sliding window.

The parameter m is set to be smaller than the minimum expected duration between the scene changes [8]. Since the DC images have a *luminance* component and two *chrominance* components the difference image is the weighted sum of the absolute values of the individual component differences.

3.2 Detection of Cuts Using Motion Information

Since a *cut* in a video stream is characterized by an abrupt scene change, a typical motion estimator will need to intracode almost all the macroblocks at a cut. Thus, by computing the extent of motion or motion distance (MD) within each macroblock in the current frame relative to the corresponding macroblock in the previous frame and setting it to a very high value if such correspondence cannot be determined, scene cuts can be detected.

The computation of MD is complicated by the fact that the corresponding macroblocks of successive frames may have MVs based on different reference frames. As an example, consider Fig. 1. Let the macroblock under consideration be (i, j) for frames #1 and #2. Also let $MV_1 = (u_1, v_1)$ be the FPMV of the $(i, j)^{th}$ macroblock of frame #1, and $MV_2 = (u_2, v_2)$ be the BPMV of the $(i, j)^{th}$ macroblock of frame #2. Here the $(i, j)^{th}$ macroblock of frame #2 has a

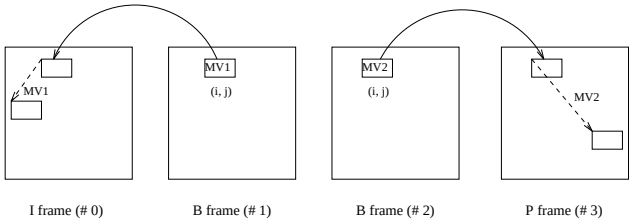

Fig. 1. Motion Distance Computation: Possible scenario

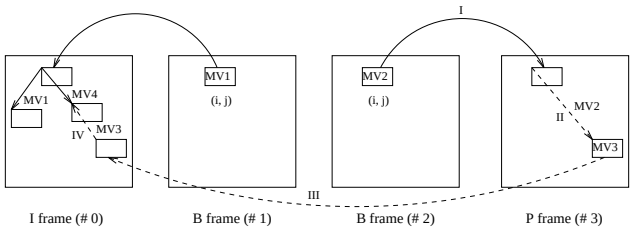

Fig. 2. Motion Distance Computation: Two-level indirection

BPMV based on the $(i,j)^{th}$ macroblock of frame #3. However, the corresponding macroblock in frame #1 has a FPMV based on the $(i,j)^{th}$ macroblock of the reference frame #0. In order to calculate the MD or the motion in frame #2 relative to frame #1, we need to find the MVs for the corresponding macroblocks of successive frames based on the *same* reference frame. Thus, we require a *two-level indirection* for the $(i,j)^{th}$ macroblock of frame #2 to obtain its MV based on frame #0. This two-level indirection is shown using dotted arrows in Fig. 2. The order of computation is indicated by the number on the arrows. This two-level indirection yields $MV_4 = (u_4, v_4)$, which is the MV of the $(i,j)^{th}$ macroblock of frame #2 based on the reference frame #0. In the above case, the MD can be computed as $\text{MD} = \sqrt{(u_4 - u_1)^2 + (v_4 - v_1)^2}$. Since the computation of MD involves floating point arithmetic, we approximated it by MD_a the computation of which entails only integer arithmetic: $\text{MD}_a = |u_4 - u_1| + |v_4 - v_1|$.

In general, for two consecutive frames *Prev* and *Curr* the corresponding macroblocks may have an FPMV, a BPMV, both or none. Thus, depending on the type of MVs the macroblocks possess, sixteen cases need to be dealt with. We refer the interested reader to [1] for a detailed explanation of how MD is computed in each of these cases. In order to detect cuts, we calculate the *motion edge* magnitude for each frame which is defined as the sum of MD's (or MD_a's) of all the macroblocks in the frame. We detect scene cuts by thresholding the peaks within a sliding window of length $2m + 1$ frames in the plot of the motion edge magnitude versus frame number using the same criteria discussed in Section 3.1.

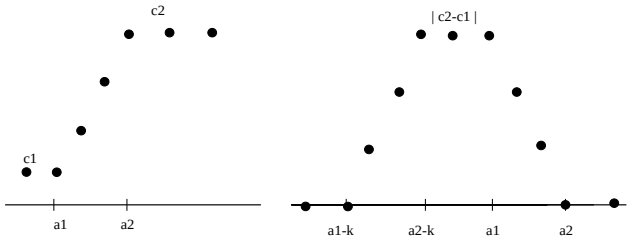

Fig. 3. Gradual Transition and Difference Sequence

3.3 Detection of Gradual Scene Changes Using DC Images

A gradual scene transition is modeled as a linear transition from c_1 to c_2, in the time interval $[a_1, a_2]$ (Fig. 3) and modeled as [8]:

$$g_n = \begin{cases} c_1, & n < a_1, \\ \frac{c_2 - c_1}{a_2 - a_1}(n - a_2) + c_2, & a_1 \leq n < a_2, \\ c_2, & n \geq a_2 \end{cases} \tag{1}$$

Assuming that $k > a_2 - a_1$, the difference sequence $D_i^k(g_n) = d(X_i, X_{i+k})$ is given by [8]:

$$D_i^k(g_n) = \begin{cases} 0, & n < a_1 - k, \\ \frac{|c_2 - c_1|}{a_2 - a_1}[n - (a_1 - k)], & a_1 - k \leq n < a_2 - k, \\ |c_2 - c_1|, & a_2 - k \leq n < a_1, \\ -\frac{|c_2 - c_1|}{a_2 - a_1}(n - a_2), & a_1 \leq n < a_2, \\ 0, & n \geq a_2 \end{cases} \tag{2}$$

The plots of g_n and $D_i^k(g_n)$ are shown in Fig. 3. The plateau between $a_2 - k$ and a_1 has a maximum constant height of $|c_2 - c_1|$ if $k > a_2 - a_1$. In order to detect the plateau width, it is required that for fixed k:
(1) $|D_i^k - D_j^k| < \epsilon$, where $j = i - s, \ldots i - 1, i + 1, \ldots, i + s$, and
(2) $D_i^k \geq l \times D_{i - \lfloor k/2 \rfloor - 1}^k$ or $D_i^k \geq l \times D_{i + \lfloor k/2 \rfloor + 1}^k$, for some large value of l.

Since the width of the plateau is $k - (a_2 - a_1) + 1$, the value of k should be chosen to be $\approx 2(a_2 - a_1)$ where $a_2 - a_1$ is the (expected) length of the transition.

3.4 Detection of Gradual Scene Changes Using Motion Information

During a dissolve, the motion estimation algorithm typically finds the best matching blocks in the reference frame(s) for blocks in the current frame but at the cost of higher prediction error. Also, in a typical dissolve, the error is uniformly distributed in space and value over all of the macroblocks. These observations are encapsulated in the following metrics (i) The average error $E_{avg} = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} E(i, j)$ over the $M \cdot N$ macroblocks should be high, (ii) The error variance $\sigma_E^2 = \frac{1}{MN} \sum_{i=1}^{M} \sum_{j=1}^{N} [E_{avg} - E(i, j)]^2$ should be high

and (iii) The error cross covariance $\sigma_{ij}^2 = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} ijE(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} E(i,j)} - i_{avg}j_{avg}$, where $i_{avg} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} iE(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} E(i,j)}$ and $j_{avg} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} jE(i,j)}{\sum_{i=1}^{M}\sum_{j=1}^{N} E(i,j)}$, should be low.

4 Detection of Scenes with Camera Motion

Scenes containing dominant camera motion such as pans and zooms can be detected and classified based on the underlying pattern of the MVs. In a pan, most of the MVs are aligned in a particular direction. For a zoom-in most of the MVs point radially inwards towards the focus-of-expansion (FOE) whereas for a zoom-out they point radially outwards from the FOE.

Let θ_{ij} be the direction of the MV associated with the ijth macroblock. Let $\theta_{avg} = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\theta_{ij}$ be the average of the MV directions in the frame. For a frame to qualify as a member of a pan shot, the variance $\sigma_\theta^2 = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}[\theta_{ij} - \theta_{avg}]^2$ should be less than a predefined threshold.

For zoom-ins and zoom-outs we analyze the MVs of frame in each of the four quadrants. Let MV_x and MV_y denote the MV components along the horizontal and vertical directions respectively. We compute the total number of macroblocks in each quadrant satisfying the following criteria:
Zoom-In: (i) First Quadrant: $MV_x < 0$ and $MV_y < 0$; (ii) Second Quadrant: $MV_x > 0$ and $MV_y < 0$; (iii) Third Quadrant: $MV_x > 0$ and $MV_y > 0$; and (iv) Fourth Quadrant: $MV_x < 0$ and $MV_y > 0$.
Zoom-Out: (i) First Quadrant: $MV_x > 0$ and $MV_y > 0$; (ii) Second Quadrant: $MV_x < 0$ and $MV_y > 0$; (iii) Third Quadrant: $MV_x < 0$ and $MV_y < 0$; and (iv) Fourth Quadrant: $MV_x > 0$ and $MV_y < 0$.

If the total number of macroblocks in the frame satisfying the aforementioned quadrant-specific conditions exceeds a specified threshold then the frame is classified as belonging to a zoom-in or zoom-out as the case may be.

5 Integrated Parsing

The current integration of luminance-based, chrominance-based and motion-based approaches is based on computing a joint decision function which is the weighted sum of the detection criteria of the individual approaches. At present the weights are selected by the user. Automatic weight selection is a topic we wish to pursue in the future. At this time the integrated approach applies only to scene change detection i.e., detection of abrupt and gradual scene changes. Detection of pans and zooms is done based on motion information alone. The GUI for the system has been developed in *Java* using JDK 1.1.5. The GUI allows the user to select the MPEG video, decide which parsing approach to use, decide various threshold values, decide the relative weights of the individual approaches in the integrated approach, and view the plots of the various detection criteria.

6 Experimental Results

The MPEG1 video clips, used for testing the proposed technique were either generated in our laboratory or downloaded over the Web [7]. The *Tennis* sequence contains cuts at frames 90 and 150. It was observed that the scene changes were successfully detected only by the MV approach (Fig. 4), with the DC difference approach displaying very small peaks in the luminance and chrominance domain (Fig. 5). The integration of the two approaches indicates clear peaks at both scene cut points (Fig. 6) thus showing it to be more reliable.

The *Spacewalk1* sequence contains 3 different dissolve sequences: between frames 74 and 85, between frames 155 and 166, and between frames 229 and 242. Both the approaches detected the 3 dissolves accurately. However, it was found that the DC k-difference approach overestimated the span of the dissolve (Fig. 7). The results of the integrated approach (Fig. 9) were found to be more accurate than those of the MV approach (Fig. 8) and the DC k-difference approach (Fig. 7) individually.

The videos used for testing the zoom and pan detection were created in our laboratory using a video camera. The results for one of the zoom sequences are depicted in Fig. 10. This clip has 267 frames with 2 zoom-outs, between frames 77 and 90, and frames 145 and 167. The algorithm successfully detected these zoom-out sequences (Fig. 10). The results for one of the pan sequences are shown in Fig. 11. The clip has a pan sequence starting from frame 90 and ending at frame 153. The algorithm successfully detected this pan sequence, by displaying a low value for the variance of the MV angle (Fig. 11).

7 Conclusions and Future Work

The integrated approach presented in this paper, combines information from multiple sources such as luminance, chrominance and motion and is capable of more reliable detection of cuts and dissolves, and also pans and zooms in MPEG1 video. Since the MPEG1 video is minimally decompressed during parsing, the approach results in great savings in memory usage and processing bandwidth. Future work will investigate detection of other scene parameters such as wipes, morphs and object motion, the automatic selection of the various threshold values and also learning these threshold values from examples.

References

1. S.M. Bhandarkar and A.A. Khombadia, Motion–based Parsing of Compressed Video, *Proc. IEEE IWMMDBMS*, Aug. 1998, pp. 80–87.
2. H. Ching, H. Liu, and G. Zick, Scene decomposition of MPEG compressed video, *Proc. SPIE Conf. Dig. Video Comp.*, Vol. 2419 Feb. 1995, pp. 26-37.
3. D.L. Gall, MPEG: A video compression standard for multimedia applications, *Comm. ACM*, Vol. 34(4), 1991, pp. 46-58.

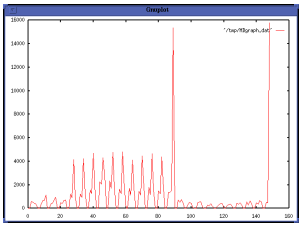

Fig. 4. *Tennis*: Motion edge plot

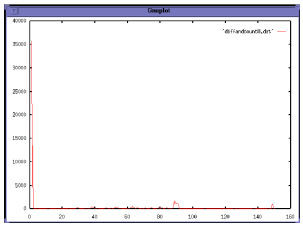

Fig. 5. *Tennis*: DC difference plot

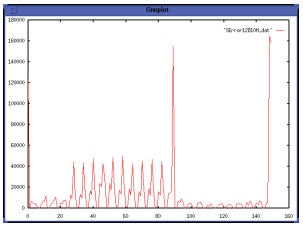

Fig. 6. *Tennis*: Integrated approach plot

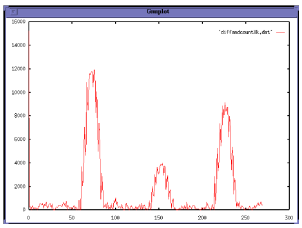

Fig. 7. *Spacewalk1*: DC k-difference plot

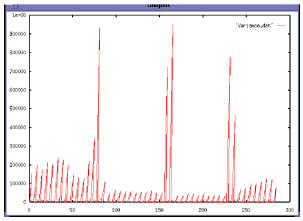

Fig. 8. *Spacewalk1*: Error variance plot

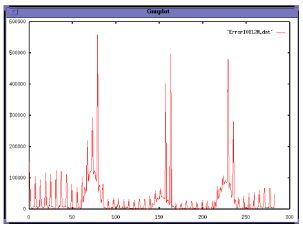

Fig. 9. *Spacewalk1*: Integrated approach plot

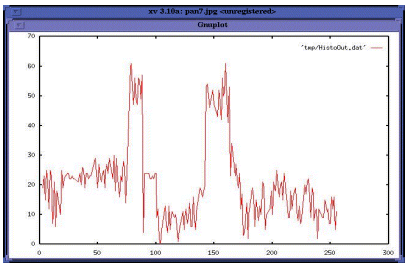

Fig. 10. *Pan7*: Plot of % of pixels satisfying zoom criteria

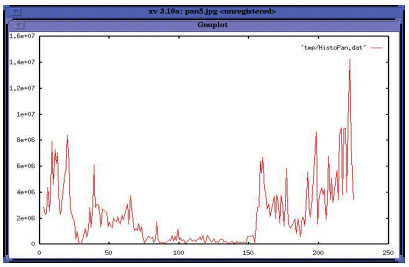

Fig. 11. *Pan5*: Plot of MV angle variance

4. V. Kobla, D. Doermann, K.I. Lin and C. Faloutsos, Compressed domain video indexing techniques using DCT and motion vector information in MPEG video, *Proc. SPIE Conf. Stor. Retr. Img. Vid. Dbase.*, Vol. 3022, Feb. 1997, pp. 200-211.
5. J. Meng and S.F. Chang, CVEPS - A Compressed Video Editing and Parsing System. *Proc. ACM Conf. Multimedia*, Nov. 1996, pp. 43-53.
6. K. Shen and E. Delp, A fast algorithm for video parsing using MPEG compressed sequence, *Proc. IEEE Intl. Conf. Image Process.*, Oct. 1995, pp. 252-255.
7. University Of Illinois, Urbana-Champaign. ACM Multimedia Lab for sources of MPEG video data, URL : http://www.acm.uiuc.edu/rml/Mpeg/
8. B.L. Yeo and B. Liu, Rapid scene analysis on compressed video, *IEEE Trans. Cir. and Sys. for Video Tech.*, Vol. 5(6), 1995, pp. 533-544.
9. H.J. Zhang, C.Y. Low, and S.W. Smoliar, Video parsing and browsing using compressed data, *Jour. Multimedia Tools Appl.*, Vol. 1(1), 1995, pp. 89-111.