DETECTION OF SLOW-MOTION REPLAY SEQUENCES FOR IDENTIFYING SPORTS VIDEOS

Vikrant Kobla Daniel DeMenthon David Doermann
Laboratory for Language and Media Processing
University of Maryland, College Park, MD 20742 - 3275, USA.
{kobla,daniel,doermann}@cfar.umd.edu

Abstract - Automated classification of digital video is emerging as an important piece of the puzzle in the design of content management systems for digital libraries. The ability to classify videos into various genres such as sports, news, movies, or documentaries increases the efficiency of indexing, browsing, and retrieval of video in large databases. In this paper, we present an automated technique for identifying slow-motion replays directly from the compressed domain of MPEG video. It uses the macroblock, motion, and bit-rate information that is readily accessible from MPEG video with very minimal decoding, leading to enormous gains in processing speeds.

INTRODUCTION

With recent advances in digital video coding and transmission, larger and larger amounts of digital video are becoming available every day from various sources. The advent of digital television and HDTV is yet another motivating factor for automated analysis of digital video. Manual annotation or analysis of video is an expensive and arduous task that will not be able to keep up with this rapidly increasing volume of video data in the near future.

Classification of digital video into various genres, or categories such as sports, news, movies, commercials, or documentaries is a important task in building any content management system for the digital libraries required to house these large amounts of data. Automated classification of video leads to more efficient indexing, retrieval, and browsing of data in digital libraries.

Our current focus is to develop a system that automatically distinguishes sports clips from other clips. This system can later be extended, with the help of other features, to classify other genres or types of video. In this paper, we present a technique that automatically detects the presence of slow-motion action replays in sports videos. Our algorithm works on features directly available from the compressed domain of MPEG video, thereby avoiding the expensive inverse DCT computation required to convert values from the frequency domain to the pixel or image domain. This allows analysis to be performed faster than real time.

RELATED WORK

To the best of our knowledge, our system is the first to tackle the problem of automated identification of action replays. Several papers exist in the literature that deal with analysis of specific sports programs such as soccer, football, or basketball. Rees et al. [5] present a system called CLICK-IT that can used for interactively viewing replays of sporting events. Their system uses computer vision techniques for tracking objects. Ariki and Sugiyama [1] describe a system for classifying sports scenes using a multiple subspace method. They exploit the fact that most sports events contain typical scenes involving batter, pitcher, field, or spectators, for example.

REPLAY DETECTION OVERVIEW

A salient feature of many sports clips is the presence of slow-motion replays. Unless high-speed cameras are used, slow-motion replays are generated by slowing the frame rate of the playback of the recorded event. This essentially causes a single frame to be repeated several times (sometimes up to 3 or 4 times for very slow motion) leading to jerky shifts between frames after periods of no motion, when the playback frame is advanced to the next frame. Let us define a *still frame* as a frame that is identical to its previous frame. Similarly, let us define a *shift frame* as a frame which results after a shift from its previous frame. Slow-motion replay sequences can be modeled as a repetitive pattern of a non-zero number of still frames being followed by a non-zero number of shift frames. Depending on the speed of the replay, there might be one still frame followed by many shift frames, or there might be one shift frame followed by many still frames.

The encoded MPEG stream provides important clues for identifying these patterns. By detecting these patterns in the MPEG stream, we are able to pinpoint sequences of the video clip where a replay is present. Occurrences of replays in a video clip provides clues for distinguishing sports clips from other types of clips.

In addition the slow-motion replay, various other features can be incorporated into our system for identifying sports videos. These include detecting the presence of *consistent* camera motion such as longs pans, large magnitudes of motion, percentage of shots in the clip containing motion. Sports clips frequently contain features that are similar to those present in raw footage clips with sudden camera jerks. Another feature of sports clips is the frequent appearance of text in score and statistic displays, and on players' jerseys.

We have implemented a replay detection algorithm that uses only the macroblock type information for identifying the patterns of repetitive still frames, and jerky shifts between the stills. Our system has options for performing further analysis by using a few more features for robust detection.

ALGORITHM DETAILS

Analyzing the macroblock pattern of the B frames yields clues about the presence of changes between the current B frame and the past and future reference frames. Macroblocks (MBs) in a B frame are primarily of three types: forward-predicted, backward-predicted, and bidirectionally-predicted [4]. There are also skipped macroblocks which essentially behave as one of the above three, and intra-coded macroblocks which are not predicted from any reference block. Forward-predicted MBs contain forward-predicted motion vectors (FPMVs), backward-predicted MBs contain backward-predicted motion vectors (BPMVs), and bidirectionally-predicted MBs contain both FPMVs and BPMVs. During a series of still frames, if a still B frame is identical to its previous reference frame and not identical to its next reference frame (i.e, a shift occurs at or before the next reference frame), then, most of its MBs are either forward-predicted or skipped as forward-predicted. When there is a shift at a B frame and this B frame after the shift is identical to the next reference frame due to the latter being a still frame, then the MBs in the B frame will primarily contain backward-predicted MBs or skipped MBs that behave as backward-predicted blocks. In either case, we make sure that most of the appropriate non-bidirectionally-predicted MBs (FPMVs in the former, and BPMVs in the latter case) have zero motion vectors. We require this since we are testing for stillness between the B frames and the appropriate reference frame. But sometimes this situation can resemble a shot change pattern [3], even though there is actually no shot change. Hence, we may need to determine whether it is an actual shot change or not. Many segmentation algorithms exist in the literature that can detect cuts easily. By pre-processing the video with a segmentation algorithm, we can quickly determine whether the pattern is due to a cut or not. If there is no cut, we have detected the presence of stillness. Our implementation utilizes the segmentation algorithm presented in our earlier work [3]. By searching for these patterns in B frames, we can quickly determine the still and shift frames. Once the frames are labeled, they are aggregated into labeled replay sequences.

As mentioned above, we have implemented a few enhancements to increase the robustness of the detection process. One such enhancement involves analyzing the number of bits that were utilized for encoding each frame in the video clip. When the same frame is being repeated several times, successive still frames that are encoded as B frames contain very little information. The numbers of bits used to encode these frames are very low. But frames at or near a jerky shift require much higher numbers of bits to encode them. This leads to large variations in the bits vs. frame number plot during a replay.

Another useful feature is the plot of the dominant magnitude of the *flow vectors* [2] of the frames in the video clip. During still frames, the dominant magnitude is almost always 0. The jerky shifts result in non-zero dominant magnitudes. Analyzing the plot of dominant magnitude vs. frame number for this pattern of non-zero values embedded in a series of zero values gives

further evidence of the presence of a slow-motion replay sequence.

Our basic algorithm using only the MB type information runs at fast speeds of up to 180 frames per second of SIF (352x240)-resolution video. Even with the enhancements, the replay detection algorithm runs faster than real time.

RESULTS

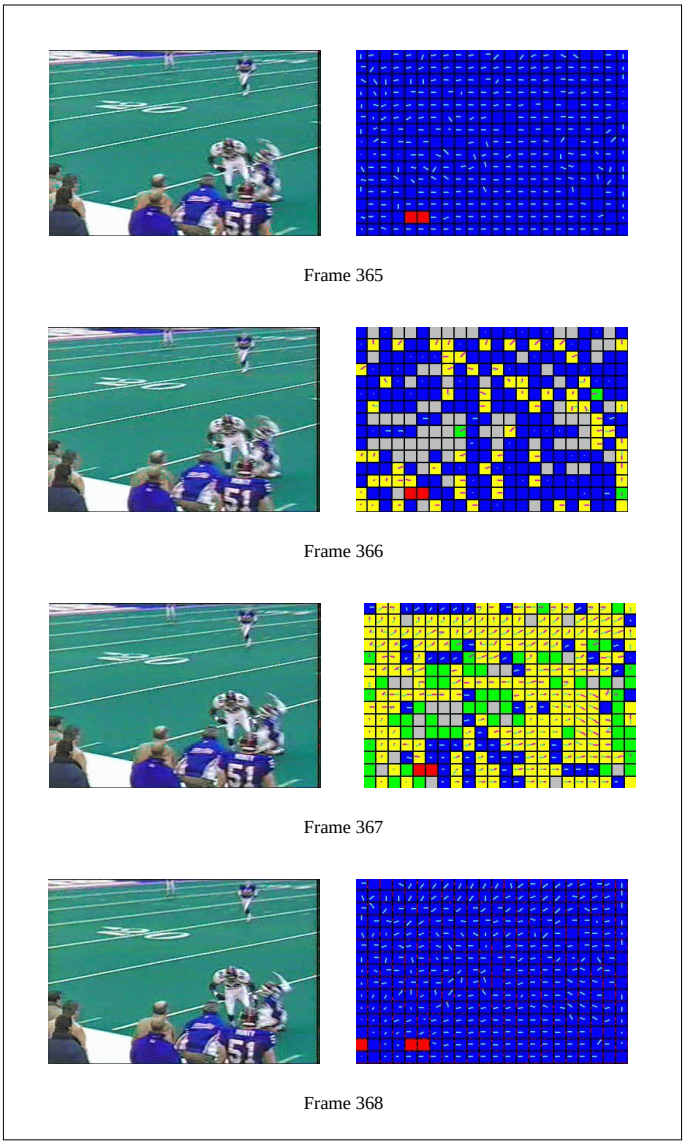
Figure 1(a) shows four consecutive frames taken from a football game replay sequence that extended from frame 165 to around frame 400. The four frames are arranged vertically from top to bottom. Frames 365 and 368 are P frames, and frames 366 and 367 are B frames. Each frame's color-coded MB map is shown to the right of it and its frame number is shown below the two in the center. The color code legend is shown in Figure 1(b). It can be seen that the first two frames (frames 365 and 366) are identical, with the next two frames being shift frames, i.e, shifts take place between frames 366 and 367, and between frames 367 and 368. The absence and presence of shifts can be noted by observing the white sleeve of the player falling down. Since frame 366 is a still frame, most of its MBs are forward-predicted with their FPMVs being zero. Also note that since frames 367 and 368 are different, many of the MBs of frame 367 are bidirectionally coded; had they been identical, the MBs would have been backward-predicted. Figure 2(a) shows a plot of the number of bits vs. frame number. The increased variation in the frame bitrate can be observed in the replay range (frames 165-400). Figure 2(b) shows a plot of the dominant flow vector magnitude vs. frame number. A zoomed version of the plot from frame 140 to frame 450 in Figure 2(c) gives better detail. The large variations in magnitudes with variations extending to zero or near-zero values further enhances the robustness of the detection. We have tested our algorithm on several minutes of replay sequences, obtaining good results.

CONCLUSION

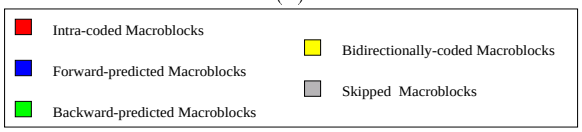
We have presented a technique for detecting slow-motion action replays in sports videos. Presence of replays can be used as a feature for distinguishing sport videos from other types of video such as movies, documentaries etc. Our technique uses macroblock type and motion vector features that can be extracted inexpensively from compressed MPEG video. Our basic algorithm runs as fast as 180 frames per second. Even with some enhancements, our algorithm runs faster than real time.

ACKNOWLEDGEMENTS

The support of this effort by the Department of Defense under contract MDA 9049-6C-1250 is gratefully acknowledged.



(a)



(b)

Figure 1: (a) Set of four frames *(start,still,shift,shift)*, their MB maps and frame numbers. (b) Color code legend for the MB maps.

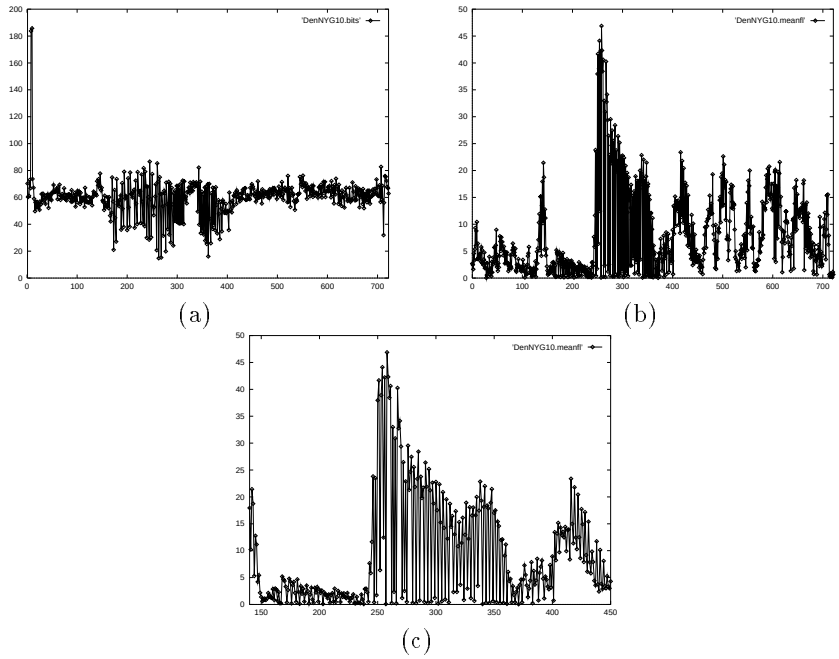


Figure 2: (a) Frame bit-rate plot, (b) Flow vector magnitude plot, (c) Close-up of plot in (b) from frame 140 to 450.

References

[1] Y. Ariki and Y. Sugiyama. Classification of TV sports news by DCT features using multiple subspace method. In *Proc. of the IEEE International Conference on Pattern Recognition*, volume 2, pages 1488–1491, 1998.

[2] V. Kobla and D. Doermann. Extraction of features for indexing MPEG-compressed video. In *Proc. of IEEE First Workshop on Multimedia Signal Processing (MMSP)*, pages 337–342, 1997.

[3] V. Kobla, D.S. Doermann, and K.I. Lin. Archiving, indexing, and retrieval of video in the compressed domain. In *Proc. of the SPIE Conference on Multimedia Storage and Archiving Systems*, volume 2916, pages 78–89, 1996.

[4] D. Le Gall. MPEG: A video compression standard for multimedia applications. *Communications of the ACM*, 34:46–58, 1991.

[5] D. Rees, J.I. Agbinya, N. Stone, F. Chen, S. Seneviratne, M. de Burgh, and A. Burch. CLICK-IT: interactive television highlighter for sports action replay. In *Proc. of the IEEE International Conference on Pattern Recognition*, volume 2, pages 1484–1487, 1998.