Compressed-Domain Shot Boundary Detection for H.264/AVC Using Intra Partitioning Maps

Sarah De Bruyne, Jan De Cock, Chris Poppe, Charles-Frederik Hollemeersch, Peter Lambert and Rik Van de Walle

In: Lecture Notes in Computer Science, 6523(2011), 29-39, 2011.

Optional: http://www.springerlink.com/content/v0h85302q6761g07/

Sarah De Bruyne, Jan De Cock, Chris Poppe, Charles-Frederik Hollemeersch, Peter Lambert and Rik Van de Walle (2011). Compressed-Domain Shot Boundary Detection for H.264/AVC Using Intra Partitioning Maps. *Lecture Notes in Computer Science 6523(2011)* 29-39. DOI: 10.1007/978-3-642-17832-0_4

Compressed-domain shot boundary detection for H.264/AVC using intra partitioning maps

Sarah De Bruyne, Jan De Cock, Chris Poppe, Charles-Frederik Hollemeersch,
Peter Lambert, and Rik Van de Walle

Ghent University - IBBT,
Department of Electronics and Information Systems - Multimedia Lab
Gaston Crommenlaan 8 bus 201, B-9050 Ledeberg-Ghent, Belgium
sarah.debruyne@ugent.be
http://multimedialab.elis.ugent.be

Abstract. In this paper, a novel technique for shot boundary detection operating on H.264/AVC-compressed sequences is presented. Due to new and improved coding tools in H.264/AVC, the characteristics of the obtained sequences differ from former video coding standards. Although several algorithms working on this new standard are already proposed, the presence of IDR frames can still lead to a low accuracy for abrupt transitions. To solve this issue, we present the motion-compensated intra partitioning map which relies on the intra partitioning modes and the motion vectors present in the compressed video stream. Experimental results show that this motion-compensated map achieves a high accuracy and exceeds related work.

Keywords: Shot boundary detection, video analysis, compressed domain, H.264/AVC

1 Introduction

During the last decades, a significant increase in the use and availability of digital multimedia content can be witnessed. Unfortunately, these video collections often lack information related to the structure and the actual content of the video. When accessing these video streams in case no metadata is available, time-consuming, sequential scanning is the only option. As a consequence, to facilitate multimedia consumption, intensive research has been done in the domain of indexing, retrieval, browsing, and summarization. Since the identification of the temporal structure of video is an essential task for many video indexing and retrieval applications, the first step commonly taken for video analysis is shot boundary detection as shots are the basic units for a large majority of video content analysis algorithms [5]. According to whether the transition between consecutive shots is abrupt or not, boundaries are classified as cuts or gradual transitions.

In order to preserve storage space and to reduce bandwidth constraints, most video data is available in compressed form. By relying on compressed-domain features which can be extracted directly from the compressed bitstream, time-consuming decompression can be avoided and coarse but potentially useful information present in the bitstream can efficiently be reused. Consequently, compressed-domain algorithms for shot boundary detection are gaining importance. In the past, many compressed-domain algorithms were proposed which rely on the MPEG-1 Video and MPEG-2 Video standards. However, as the H.264/AVC video coding standard [12] performs significantly better than any prior standard in terms of coding efficiency, more video content will be coded in this video format in the future. Its superior compression performance can mainly be attributed to the new or improved coding tools. However, these coding tools influence the compressed domain features to a great extent and render prior algorithms working on MPEG-1 Video and MPEG-2 Video obsolete. As a consequence, recently, efforts have been undertaken to design new shot boundary detection algorithms working on H.264/AVC.

The outline of this paper is as follows. Section 2 addresses related work and remaining issues in the area of shot boundary detection algorithms operating on H.264/AVC-compressed video streams. Section 3 introduces our novel algorithm to detect shot boundaries, whereas results are provided in Section 4. Conclusions are drawn in Section 5.

2 Related work

2.1 General techniques

In literature, most of the algorithms work on MPEG-1 Video and MPEG-2 Video. On the on hand, DCT coefficients are exploited. Arman *et al.* [1] compare a subset of DCT coefficients of two successive I frames to calculate the frame differences as the coefficients in the frequency domain are mathematically related to the spatial domain. The temporal resolution of this type of techniques is low, resulting in an increased amount of false alarms when camera motion is present. Furthermore, Yeo and Liu defined the concept of DC images [13], which are spatially reduced versions of the original image and which are generated by only taking into account the first DCT coefficient in each block, i.e., the DC coefficient, and motion vectors. Based on these DC images, similarity metrics defined for color features in the pixel domain can be modified to operate in the compressed domain.

On the other hand, the distribution of the different macroblock types and motion information [4, 11] can also be used as features to detect shot boundaries. When an abrupt transition occurs between two successive P pictures, it is expected that a significant amount of macroblocks in the second frame is intra coded since these macroblocks cannot be predicted well from prior reference frames. The prediction directions of motion vectors in intermediate B frames can then be utilized to detect the exact location of the transition.

Although most algorithms mainly focus on the detection of abrupt transitions, the aforementioned features can also be used to detect gradual transitions [2]. Due to the large variety in terms of effects and duration, the accuracy for gradual transitions is typically inferior to the abrupt transitions.

2.2 Algorithms for H.264/AVC

H.264/AVC contains a number of new or improved coding tools which have a major impact on the aforementioned shot boundary detection algorithms. Firstly, intra prediction in H.264/AVC is conducted in the spatial domain by relying on neighboring samples of previously-decoded blocks in order to reduce the spatial redundancy in images. As such, DC coefficients in intra-coded pictures no longer represent average energy, but only represent an energy difference. Consequently, shot boundary detection algorithms working on DC images can no longer be applied to H.264/AVC bitstreams. The second feature, multiple reference picture motion compensation, allows an encoder to not only use the previous and following reference frame during encoding, but makes it possible to also use additional priorly-decoded frames as reference. As this results in vagueness about random access in the bitstream, the concept of Instantaneous Decoding Refresh (IDR) was introduced [12]. This special I frame indicates that no subsequent pictures in the bitstream will require references to pictures prior to the IDR picture in decoding order. The prediction chain is broken; hence it is insufficient to only rely on reference directions to detect shot boundaries.

Up to now, a few algorithms working on H.264/AVC-compressed bitstreams have been published. In [8], Kim *et al.* define a dissimilarity metric for I frames by relying on macroblock partitions (i.e., Intra_4×4 and Intra_16×16 prediction). A more complex discontinuity metric based on solely I frames is proposed by Kuo and Lo where intra mode histogram distances are calculated based on the different intra prediction mode directions [9]. However, the exact location of a shot boundary cannot be determined by these two algorithms as information from P and B frames is not taken into consideration. Extensions on this second algorithm are proposed in [10] and[14] to deal with all type of frames. Firstly, these algorithms exploit the intra prediction histograms to locate potential groups of pictures (GOPs) where the probability of a shot boundary is high. Secondly, the different inter prediction modes and motion vector directions of the intermediate P and B frames are used to locate the exact location of the transition by making use of thresholds or Hidden Markov Models. However, due to the presence of IDR frames, shot boundaries located just before the IDR frames are falsely detected in case the dissimilarity between the two I frames is relatively large. Furthermore, when the dissimilarity metric is low but both I frames actually belong to different shots, the temporal prediction chain is not considered, leading to missed detections.

In our previous work [3], these two problems were tackled. To locate the abrupt transitions, the temporal dependencies between successive frames are examined first. Secondly, when encountering an IDR frame breaking this prediction chain, spatial dissimilarities are considered. In contrast to the aforementioned related algorithms, the changing characteristics of the content are taken into account by constructing a “static intra partitioning map”. In particular, the static intra partitioning map is updated when new intra-coded macroblocks residing at intermediate, inter-coded frames are encountered. Gradual changes are detected by relying on the percentage of intra-coded macroblocks. Since fast global and local motion can result in similar patterns as gradual transitions, motion-activity intensity is considered as well to identify the exact origin of the content change.

Although the static intra partitioning map already solves several issues, its accuracy is still inferior to video streams compressed without IDR frames. When examining the origin of the false alarms, it can be seen that they mostly occur when the content slightly changes as a result of motion, which is generally compensated by using inter prediction. In the next section, we will extend the algorithm proposed in [3] by introducing the motion-compensated intra partition map which overcomes this problem by relying on motion compensation techniques.

3 Intra partitioning maps

Whereas the abrupt transitions located in between two inter-coded frames can be detected by analyzing the main reference directions of the frames, another technique is required to determine whether a shot boundary is present in between an IDR frame and its preceding frames. To solve the issue of the broken temporal prediction chain, algorithms designed for prior video coding standards would typically make use of DC images to compare the spatial characteristics of successive frames. However, due to the introduction of spatial intra prediction in H.264/AVC, these iconic versions of the content can no longer be generated without further decoding the bitstream. Therefore, we employ intra partitioning maps which reflects the spatial dissimilarities between frames based on the selected intra partitioning modes.

H.264/AVC supports multiple intra macroblock partitions, i.e., Intra_4×4, Intra_8×8 and Intra_16×16. The first mode is generally selected by an encoder in case of significant detail, whereas the Intra_16×16 mode is preferred for smooth areas. When coding high-resolution video sequences using the High profile of H.264/AVC, Intra_8×8 will also be selected and mainly replace Intra_4×4 modes to code the high-textured areas. As the subdivision in different macroblock partitions roughly reflects the detail of the content, comparing the distribution of two frames can be used to estimate the spatial dissimilarities, as illustrated in Fig. 1.

Comparing the current I frame with the previous I frame is not recommended as the content can change significantly between these two intra-coded frames. For example, a shot boundary can be located at an intermediate P or B frame, new objects can appear, or camera motion can occur (Fig. 1). Therefore, we introduce an *intra partitioning map* M_i indicating which intra partitioning modes most likely correspond to the content of the intermediate, inter-predicted frames. This partitioning map can be constructed in different ways. In [3], the static intra partitioning map was introduced, which considers all intra-coded macroblocks in intra- as well as inter-coded frames. In this paper, the motion-compensated intra partitioning map is introduced, which extends the static map by including motion information.

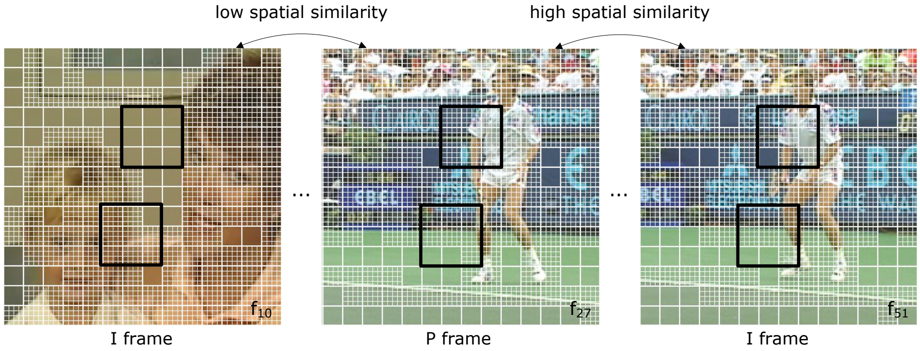


Fig. 1. Distribution of Intra_4×4 and Intra_16×16 macroblocks. Although the second frame is a P frame, it is mainly intra coded as it is the first frame of a new shot. As such, it is important to update the intra partitioning maps with information from inter-coded frames.

Static intra partitioning map A first approach to construct an intra partitioning map for the inter-coded frame f_i is by remembering for each macroblock position the intra partitioning mode which was last encountered. To put it differently, the intra partitioning map M_i of frame f_i is constructed by updating the previous map M_{i-1} with the partitioning modes of the intra-coded macroblocks in the current frame. As such, this map can be used to represent the spatial distribution of the content of the current frame, in spite of the fact that this frame mainly contains inter-coded macroblocks. By comparing the current I frame f_i with the static map M_{i-1}, the spatial dissimilarity between the current and previous frame can be calculated. However, instead of comparing partitioning modes at corresponding positions, a window of several macroblocks is selected for each macroblock. This way, small movement of objects or the camera will lead to fewer false alarms.

Motion-compensated intra partitioning map The downside of the static intra partitioning map is its limited support for motion. In particular, when the difference between two I frames belonging to the same shot is large, resulting from the large distance between these two frames or from relatively fast moving objects or the camera, this window will not be able to cover the displacement of the scene. As a result, the amount of falsely detected shot boundaries will increase. To overcome this problem, motion information can be considered in order to construct a motion-compensated intra partitioning map.

Instead of copying the partitioning modes of the previous intra partitioning map, we propose to use the partitioning modes of the reference blocks to which the motion vectors (MVs) of the current frame point to. The favorable aspect of this motion compensation step is shown in Fig. 2(a) and 2(c) and further explained below. The macroblock in f_{186} which is marked in red is part of the low-textured sky (Fig. 2(c)) and would typically result in an Intra_16×16 partitioning mode when intra prediction would be applied, as verified by coding the sequence with I frames only. When copying the partitioning mode of the macroblock located at the same location in the previous partitioning map, marked by the semi-transparent red rectangle in Fig. 2(a), incorrect spatial information belonging to the high-textured crane would be passed on. However, thanks to the motion compensation step, correct spatial information can now be stored in the partitioning map, which corresponds to the block indicated by the full red rectangle in Fig. 2(a). Obviously, for intra-coded macroblocks, the new partitioning modes are used to update the partitioning map (Fig. 2(d)).

The reference block to which the MV of the current frame points to does not necessarily coincide with macroblock or sub-macroblock boundaries. As such, this reference block can overlap with different partitioning modes, as illustrated by the green block in Fig. 2(a). Therefore, it is undesired to store only one partitioning mode for each block. Instead, we propose to store the likelihood of each possible partitioning mode. As multiple partitioning modes for inter-coded macroblocks exist and each partition contains its own MVs, it is desired to divide the intra partitioning map into a uniform field of basic units of 4×4 pixels, which corresponds to the smallest partition for which MVs can change. This subdivision also results in a finer granularity, improving the accuracy of the likelihoods of each partitioning mode.

In order to calculate the likelihoods of the different partitioning modes for an inter-coded block b in the motion-compensated intra partitioning map M_i, i.e., $M_i[b, mode]$, the likelihoods of the blocks in the reference map M_{ref} which overlap with the reference block are considered. To incorporate the percentage of overlap between the motion-compensated reference block and the overlapping block r, a new variable $OverlappingSize_r$ is introduced. This leads to the following formula where $OverlappingBlocks_b$ is defined as the set of overlapping blocks connected to block b:

$$M_i[b, mode] = \sum_{r \in OverlappingBlocks_b} OverlappingSize_r \cdot M_{ref}[r, mode]. \qquad (1)$$

When an intra-coded macroblock is encountered, the sixteen corresponding blocks in the intra partitioning map are set to one for the encountered partitioning mode, whereas the other modes are set to zero.

Let f_i denote the current I frame and M_i the intra partitioning map containing the new intra partitioning modes, M_{i-1} the map of the previous frame, and B_4 the amount of 4×4 blocks in a frame. Define $Modes$ as the set of possible macroblock partitions ($Modes = \{Intra\_4 \times 4, Intra\_8 \times 8, Intra\_16 \times 16\}$). Furthermore, let M be a placeholder for M_i and M_{i-1}, and n and m macroblocks. The dissimilarity metric Ω used to compare the current I frame f_i and the preceding motion-compensated intra partitioning map can then be defined by the average of the dissimilarity values of all blocks b in f_i.

$$\Omega(f_i) = \frac{1}{B_4} \sum_{b \in f_i} \omega(f_i, b). \qquad (2)$$

This block dissimilarity value $\omega(f_i, b)$ is obtained by comparing a window around the current block b in the intra partitioning map M_i with the collocated window in M_{i-1}. This comparison is done by making histograms of the various partitioning modes present in both windows, and thereafter, calculating the difference between these histograms.

$$\omega(f_i, b) = \frac{\sum_{t \in Modes} \left| |s_{b,t}^{M_i}| - |s_{b,t}^{M_{ref}}| \right|}{2 \cdot window\ size}. \tag{3}$$

For this purpose, the percentages of all blocks which are coded using a certain coding partitioning mode t located in the window around block b in partitioning map M_i or M_{i-1} are added up (i.e., $s_{b,t}^{M}$).

$$s_{b,t}^{M} = \sum_{n \in window(b)} M[n, t]. \tag{4}$$

Threshold selection To decide whether a shot boundary is located in between the current I frame f_i and the preceding frame f_{i-1}, the spatial dissimilarity $\Omega(f_i)$ needs to be compared to a threshold T. Although predefined, static thresholds which remain the same over the entire sequence are often applied, this type of thresholds cannot be adjusted to local properties of the sequences. As the obtained results do not represent probabilities, but rather indicate the difference compared to previous frames, we prefer to work with an adaptive threshold.

To adapt T to the local properties of the content, the M previous spatial dissimilarity values calculated for I frames are considered. Statistical information such as the mean and variation obtained from these M elements is then used to determine the local, content-dependent threshold T. Let μ_Ω denote the mean of the dissimilarity values of the M previous I frames and their preceding intra partitioning map and σ_Ω the corresponding standard deviation. T_{intra} can then be defined as:

$$T = \mu_\Omega + \alpha\sigma_\Omega. \tag{5}$$

The values for M and α are computed heuristically, resulting in typical values lying around 8 and 6 respectively. Furthermore, a lower boundary is set to this threshold to avoid extreme values. In particular, when the start of a shot is static, the corresponding dissimilarity values are close to zero. A small amount of motion further in the shot would otherwise result in a false alarm. When observing typical values for $\Omega(i)$ corresponding to abrupt transitions, an appropriate lower boundary is 0.2.

4 Performance results

Several video sequences with various characteristics in terms of resolution, length, quality, and content were selected to evaluate the performance of our shot boundary detection algorithm. The first two sequences originate from the publicly available MPEG-7 Content Set [6] and represent a part of a news sequence and a basketball sequence. Due to the low quality and resolution, three more recent, proprietary sequences were added to the test set as well.

These sequences were coded using the Joint Model reference software and Main profile enabled, which is suitable for temporal-segmentation applications. IBP GOP structures and a quantization parameter (QP) of 26 were selected. Furthermore, two configurations were made: once with I frames and once with IDR frames, which were inserted every 32 frames. As such, the effect of intra partitioning maps can be evaluated when the prediction chain is broken (i.e., when using IDR frames). As the static and motion-compensated intra partitioning maps only influence the detection of abrupt transitions, the accuracy results for gradual transitions are discarded from the accuracy results presented in Table 1.

When comparing the algorithms working on I-frame and IDR-frame sequences, a small decrease in precision is observed for some sequences, which can be attributed to the gaps in the temporal prediction chain resulting from IDR frames, which are falsely marked as abrupt transition. In general, the results of the motion-compensated intra partitioning map exceed the static intra partitioning map. This difference can mainly be attributed to shots containing medium object or camera motion. As the content between successive I frames clearly changes but the inter-coded prediction still obtains good results, the static map will not be updated whereas the motion-compensated map better reflects the content change.

The effect of the selected QP should not be neglected as this influences the selected intra partitioning modes and the amount of intra-coded macroblocks. As shown in Table 2, it catches the eye that the detection of abrupt transitions in I-frame sequences is hardly affected by the different QP values. For the IDR-frame sequences analyzed using the static and the motion-compensated intra partitioning map, the recall values in average slightly decrease. Due to the reduced amount of intra-coded macroblocks when applying higher QP values, the partitioning map is updated less frequently with intra-coded macroblock information and therefore is less accurate. As a result, the variable threshold will

Table 1. Accuracy in precision (P) and recall (R)(%) of the motion-compensated (MC) intra partitioning map. For comparison, the results for the static intra partitioning map as well as for the sequences coded using classic I frames are depicted. Furthermore, the results for the "TZI Shotdetection TrecVID 2004" algorithm are provided as reference.

Test sequence	I frames		IDR frames static map		IDR frames MC map		TZI algorithm	
	P	R	P	R	P	R	P	R
News 1	91	100	93	99	95	99	93	97
Basket	94	98	94	98	94	98	97	94
News 2	98	100	91	99	98	99	91	99
Soap	99	100	95	99	99	99	99	91
Trailer	100	100	100	100	100	100	95	99

Table 2. Influence of the quantization parameter on the accuracy of the intra partitioning map for the News 2 sequence.

QP	I frames		IDR frames static map		IDR frames MC map	
	P	R	P	R	P	R
26	98	100	91	99	98	99
30	99	99	88	99	96	99
34	100	99	91	99	97	99
38	100	99	93	98	99	98
42	99	97	97	96	98	96

typically obtain higher values, resulting in more missed detections.

The interpretation of the obtained accuracy results can be enhanced by comparing these results with related work. Therefore, the results of the publicly available “TZI Shotdetection TrecVID 2004” algorithm [7] that works in the pixel domain are added to Table 1. To detect abrupt transition candidates, this pixel-domain approach uses color histogram values which are calculated within a five frames width window and the edge change ratio calculated between consecutive frames as well as frames at a distance of ten. To confirm or reject the candidates, a block-based motion analysis is further used. On the one hand, the missed detections are primarily caused by succeeding shots covering one scene, leading to very similar color histograms. On the other hand, the origin of the false alarms are lighting changes and camera motion, even though block-based motion analysis already performed to reject some candidates.

When comparing the results of the uncompressed-domain algorithms with the results of the proposed algorithm, it can be seen that our proposed algorithm can compete in terms of accuracy. Although pixel-domain algorithms can rely on more features, they often do not perform complex calculations in order to limit the execution time.

5 Conclusions

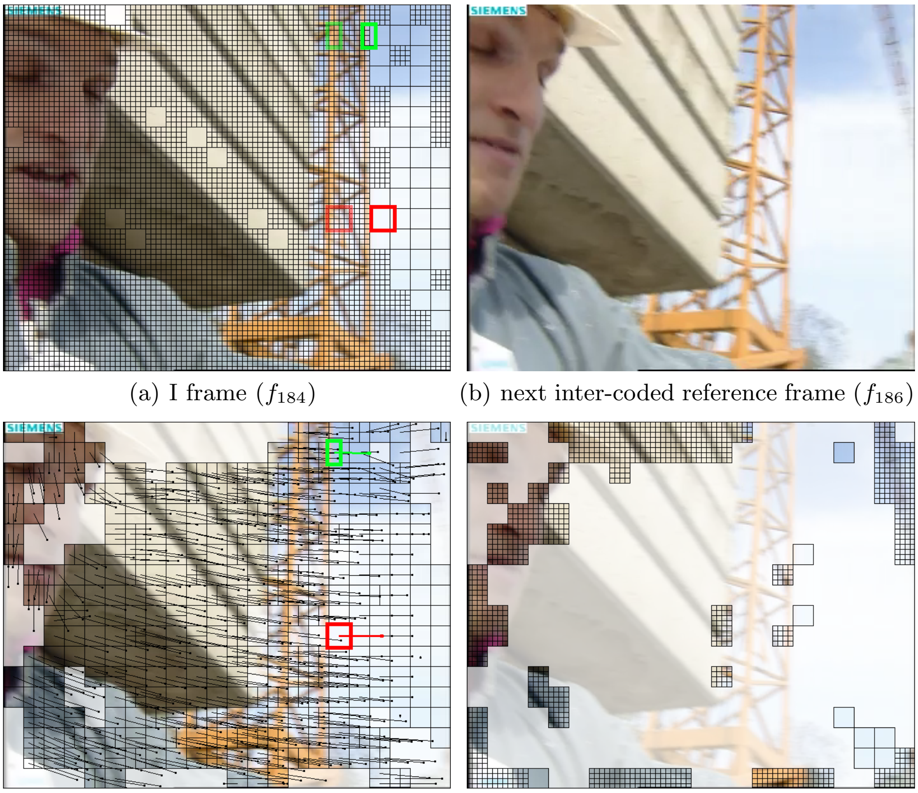
In this paper, we have introduced the concept of motion-compensated intra partitioning maps, which are used to automatically detect shot boundaries in H.264/AVC sequences. Due to the introduction of IDR frames, it is insufficient to only investigate temporal dependencies. Consequently, spatial features need to be considered to reduce the false alarms. In this paper, the motion-compensated intra partitioning map is proposed which aims at correctly detection shot boundaries near IDR frames. Experimental results show that the proposed motion-compensated intra partitioning map obtains a higher accuracy compared to the static intra partitioning map as this latter technique experiences more problems when medium object or camera motion is present.

Acknowledgments

The research activities that have been described in this paper were funded by Ghent University, the Interdisciplinary Institute for Broadband Technology (IBBT), the Institute for the Promotion of Innovation by Science and Technology in Flanders (IWT-Flanders), and the Fund for Scientific Research-Flanders (FWO-Flanders).

References

1. Arman, F., Depommier, R., Hsu, A., Chiu, M.Y.: Content-based browsing of video sequences. In: Proceedings of ACM Multimedia. pp. 97–103 (1994)
2. Bescós, J.: Real-time shot change detection over online MPEG-2 video. IEEE Transactions on Circuits and Systems for Video Technology 14(4), 475–484 (2004)
3. De Bruyne, S., Van Deursen, D., De Cock, J., De Neve, W., Lambert, P., Van de Walle, R.: A compressed-domain approach for shot boundary detection on H.264/AVC bit streams. Signal Processing: Image Communication - Semantic Analysis for Interactive Multimedia Services 23(7), 473 – 498 (2008)
4. Fernando, W.A.C.: Sudden scene change detection in compressed video using interpolated macroblocks in B-frames. Multimedia Tools and Applications 28(3), 301–320 (2006)
5. Hanjalic, A.: Towards theoretical performance limits of video parsing. IEEE Transactions on Circuits and Systems for Video Technology 17(3), 261–272 (2007)
6. ISO/IEC: Description of MPEG-7 content set. ISO/IEC JTC1/SC29/WGll/N2467 (October 1998)
7. Jacobs, A., Miene, A., Ioannidis, G.T., Herzog, O.: Automatic shot boundary detection combining color, edge, and motion features of adjacent frames. In: TRECVID 2004 Workshop Notebook Papers. pp. 197–206 (2004)
8. Kim, S.M., Byun, J.W., Won, C.S.: A scene change detection in H.264/AVC compression domain. In: Proceedings of the Pacific Rim Conference on Multimedia - Lecture Notes in Computer Science. vol. 3768, pp. 1072–1082 (2005)
9. Kuo, T.Y., Lo, Y.C.: Detection of H.264 shot change using intra predicted direction. In: Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing. pp. 204–207 (2008)
10. Liu, Y., Wang, W., Gao, W., Zeng, W.: A novel compressed domain shot segmentation algorithm on H.264/AVC. In: Proceedings of the IEEE International Conference on Image Processing. vol. 4, pp. 2235–2238 (2004)
11. Pei, S.C., Chou, Y.Z.: Efficient MPEG compressed video analysis using macroblock type information. IEEE Transactions on Multimedia 1(4), 321–333 (1999)
12. Wiegand, T., Sullivan, G.J., Bjøntegaard, G., Luthra, A.: Overview of the H.264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology 13(7), 560–576 (2003)
13. Yeo, B.L., Liu, B.: Rapid scene analysis on compressed video. IEEE Transactions on Circuits and Systems for Video Technology 5(6), 533–544 (1995)
14. Zeng, W., Gao, W.: Shot change detection on H.264/AVC compressed video. In: Proceedings of the IEEE International Symposium on Circuits and Systems. vol. 4, pp. 3459–3462 (2005)



(a) I frame (f_{184})

(b) next inter-coded reference frame (f_{186})

(c) inter-coded macroblocks and MVs in f_{186}

(d) intra-coded macroblocks in f_{186}

Fig. 2. Between the two reference frames f_{184} and f_{186} of the Foreman sequence, the camera is panning to the right side. To update the motion-compensated intra partitioning map M_{186}, the MVs of inter-coded macroblocks in f_{186} are used to locate the corresponding intra partitioning modes in f_{184} (a and c). The intra-coded macroblocks in f_{186} are used to update the intra partitioning modes (d).