EUSIPCO 2013 1569743887

AN ADAPTIVE REFERENCE FRAME RE-ORDERING ALGORITHM FOR H.264/AVC BASED MULTI-VIEW VIDEO CODEC

[1]Hany Said, [2]Akbar Sheikh Akbari and [3]Mansour Moniri

Faculty of Computing, Engineering and Sciences
Staffordshire University, Beaconside, Stafford, UK
Email: {[1]h.h.said, [2]a.s.akbari, [3]m.moniri}@staffs.ac.uk

ABSTRACT

This paper proposes an adaptive reference frame re-ordering for H.264/AVC based multi-view video codecs. The algorithm relies on statistical analysis of block matching among reference frames at low bitrate. The coded macroblocks are statistically analysed and the corresponding order for reference frames is then determined. The adaptive reference frame re-ordering algorithm is evaluated for two scenarios. In the first scenario, the multi-view videos are coded using a prediction structure with a number of reference frames. In the second scenario, a video sequence that contains several scene changes is coded. The proposed algorithm has been tested using two different prediction structures for both scenarios. The measurements were carried out on four standard multi-view datasets in addition to a sequence that contains several scenes changes. Results show that the application of the proposed reference frame re-ordering algorithm significantly saves up to 6.2% of the bitrate when coding a sequence with multiple scene changes and up to 0.2 dB when coding a sequence using multiple reference frames at low bitrate.

Index Terms— H.264/AVC, Multi-view video codec, statistical analysis, reference frames re-ordering, scene change

1. INTRODUCTION

Multi-view videos (MVVs) enable the viewer to watch these type of videos from different view-points as in free viewpoint TV (FTV) or enjoys perceiving scene depth through watching 3D videos as in three-dimensional TVs (3D-TVs) [1]. These MVVs are generated by capturing the same scene using multiple synchronized cameras at different positions and view-points [2]. Multi-view videos (MVV) contain several videos; their sizes are proportional with the number of views and resulting in huge amount of visual data which need to be compressed efficiently to enable the applications of FTV and 3D-TV.

Since the cameras filming the same scene, multi-view videos share significant amount of correlations among their views [3]. These correlations enable H.264/AVC to code MVVs efficiently through extending its coding property of multiple reference frames to exploit efficiently these correlations [4-10].

It can be seen from the literature that the H.264/AVC based MVV Codecs (MVCs) use different prediction architectures with different number of reference frames and reference frame orderings to improve their coding efficiency. Reference frame selection entails coding the current frame using previous decoded frames. These decoded frames are frames that belong to the current view (temporal reference frame) or neighbouring views (spatial and spatiotemporal reference frame) [4-10]. Reference frames (RFs) ordering reflect the way that the reference frames is placed inside H.264/AVC Decoded Picture Buffer (DPB) where few numbers of bits are used to address the closer reference frames inside this buffer (Buffer list0 is used when coding P-frames and, buffers list0 and list1 for compressing B-frames). A number of H.264/AVC based MVCs with different static RFs ordering for coding P-frames have been reported in the literature [6-10]. Temporal RFs are placed either at the beginning of list0 (e.g. [8] and [10] are depicted in Figure 1-a and, 1-b respectively), or at the end of the buffer (as shown in Figure 1-c [10]). *Fecker and Kaup* ordered RFs in opposite direction of the coding order [5] while temporal, spatial and spatiotemporal RFs are placed in an interleaved manner inside buffer as in [6, 7]. Dynamic RFs ordering for stereoscopic video coding was proposed by *Hong and Yu* [9]. Their algorithm re-orders the RFs when the number of skipped macroblocks increases. Although this algorithm efficiently encodes the stereoscopic videos, it may not meet the requirements of the real-time applications as each frame is encoded twice.

In this paper, an adaptive RFs re-ordering algorithm for multi-view video coding is proposed that encode each frame once. The proposed algorithm determines the significance of each reference frame in terms of how much it has been used as a reference in predicting blocks. Hence an analysis of block matching among the reference frames is performed to reveal the statistics of block matching. Based on the statistics of block matching for each frame, reference frames are adaptively re-ordered such that the significant references

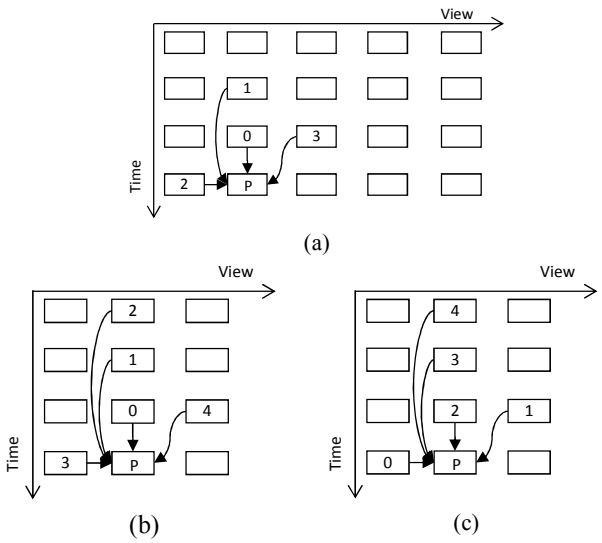

Figure 1-a Reference frame orders proposed by Bilen *et al.* (mode 3) [8], Figures 1-b and 1-c are modes 3 and mode 1 respectively that represent reference frame orders proposed by Sheikh Akbari *et al.* [10].

frames are placed first in DPB. Performance of the MVC using the propose RFs re-ordering is evaluated against the use of a statistic RFs order in two different scenarios. The first scenario is concerned with coding standard MVVs [11] and the second is concerned with coding a sequence with multiple scene changes. The performance of the H.264/AVC based multi-view video codec using the proposed algorithm is applied on prediction structure proposed in [10] for the first scenario and on prediction structure proposed in [8] for the second scenario. Results indicate the merit of the proposed RFs re-ordering in dealing with scene changes. The rest of the paper is organized as follows: in Section 2 MVV datasets are introduced. Section 3 briefly justifies the necessity of using fix RFs ordering or adaptive RFs re-ordering. Adaptive reference frame re-ordering algorithm is presented in Section 4. Experimental results are given in Section 5 and finally paper is concluded in Section 6.

2. DATASET DESCRIPTION

Four multi-view video datasets have been used in this investigation. The description of each dataset is provided in Table I. These MVVs are captured using eight cameras and they have different characteristics of motion, disparity and scene complexity [12]. Since this investigation targets coding multi-view video at low bitrate transmission, MVV datasets of CIF and QVGA size were generated from Microsoft, KDDI and MERL multi-view sequences. To achieve this, all the frames of the Microsoft datasets have been filtered using a 5×5 FIR Kaiser low-pass filter (the coefficients of this filter are tabulated in Table II); filtered frames are then down-sampled by a factor of 2 both

Table I Datasets Description

Dataset Name	Provider	Frame size / Frame format	Camera setup	Inter-cameras' distance
Break-dancers	Microsoft	1024×768 4:4:4	1D/arc	20 cm
Ballet	Microsoft	1024×768 4:4:4	1D/arc	20 cm
Race1	KDDI	640×480 4:2:0	1D/ parallel	20 cm
Exit	MERL	640×480 4:2:0	1D/ parallel	19.5 cm

Table II KAISER FIR FILTER COEFFICIENTS

0	0	0.0393	0	0
0	0.0653	0.1077	0.0653	0
0.0393	0.1077	0.1511	0.1077	0.0393
0	0.0653	0.1077	0.0653	0
0	0	0.0393	0	0

horizontally and vertically then the resulting frames are cropped from point (P_x, P_y)=(120,47) for Break-dancers and (P_x, P_y)=(80,47) for Ballet sequences and corresponding CIF size sequences are generated [7]. The resulting RGB frames are finally converted to YUV in full colour sampling format 4:4:4. The luminance components of the KDDI and MERL datasets are also filtered and down-sampled generating full colour sampling QVGA sizes. Frames of different views are interleaved using time first ordering to generate a single sequence [12]. A sequence of QVGA size with different multi-view scenes is generated by interleaving the previous MVVs together. Microsoft datasets are further down-sampled in order to match QVGA resolution size. The first six frames from each view within MVVs are used to generate a MVV sequence where sixteen consecutive frames from each video are concatenated to a MVV sequence, thus the resulting sequence contains 192 frames.

3. DO FRAMES USE SAME OR DIFFERENT REFERENCE FRAME ORDERING IN MULTI-VIEW VIDEO CODING?

In a H.264/AVC based multi-view video codecs, the order of RFs is fixed through coding the entire MVVs. This section investigates whether frames should use the same order of RFs or should they follow different RFs orders. A statistical analysis of block matching among reference frames has been conducted using H.264/AVC based MVV codec using a prediction structure depicted in Figure 2. This analysis determines the contribution of each RF for

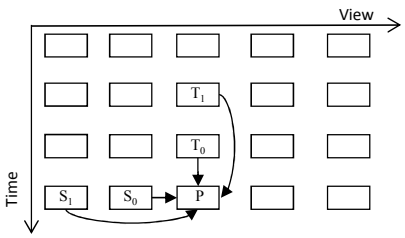

Figure 2 the prediction structure used in investigating reference frame order

Table III Six cases for reference frames orders

Case Label	Ref_0	Ref_1	Ref_2	Ref_3
A	T_0	S_0	S_1	T_1
B	T_0	S_0	T_1	S_1
C	S_0	T_0	S_1	T_1
D	S_0	T_0	T_1	S_1
E	T_0	T_1	S_0	S_1
F	S_0	S_1	T_0	T_1

Table IV shows labels which reflect the appropriate order of reference frames for the coded Break-dancers.

t_i / V_k	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
V_2	C	C	C	C	D	C	D	E	D	C	C
V_3	B	B	B	B	B	B	A	B	B	B	B
V_4	C	D	C	C	C	C	C	C	C	C	C
V_5	A	A	A	C	C	C	C	C	C	C	C
V_6	C	C	C	C	C	C	C	C	C	C	C

Table V shows labels which reflect the appropriate order of reference frames for the encoded Ballet.

t_i / V_k	t_2	t_3	t_4	t_5	t_6	t_7	t_8	t_9	t_{10}	t_{11}	t_{12}
V_2	A	C	D	B	B	B	B	B	B	B	A
V_3	B	B	B	A	B	A	A	A	A	A	B
V_4	A	B	B	B	A	A	B	A	A	B	A
V_5	B	A	B	A	D	B	D	D	C	B	C
V_6	B	A	B	C	C	C	C	C	C	C	C

predicting P-frame using all block sizes. All inter-picture coding modes and intra-prediction have been enabled. Bitrate control is enabled to encode the given MVV at low bitrate (64 Kbps).

The basic idea beyond this section is to reveal the order of RFs after encoding the P-frame using this order; T_0, T_1, S_0 and S_1. The statistic of the block matching amongst RFs is calculated and used to sort the RFs in descending order.

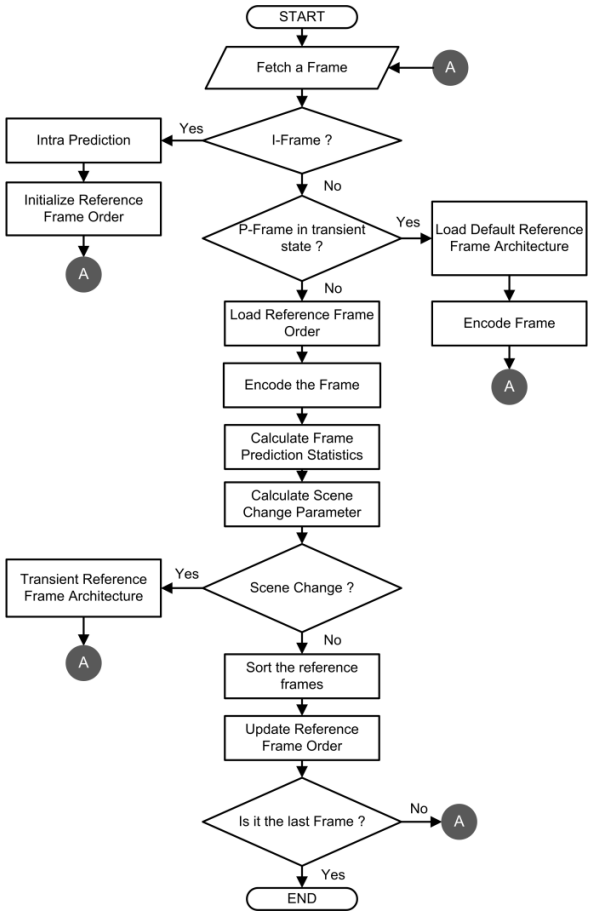

Figure 3 Adaptive Reference Frames Re-ordering Algorithm

The sorted RFs are then given a label. These labels are tabulated in Table III, where there are six different RFs orders starting from Label A to Label F and Ref_i represent a temporal T- or Spatial S-reference frame.

This investigation has been applied on Break-dancers and Ballet using the first seven views. The first two views; V_0 and V_1; are not involved in this analysis due to unavailability of some reference frames (e.g. S_0 and S_1). Tables IV and V, show the suitable reference frames order in terms of "labels", based on the statistics of block matching among four reference frames for the first 55 frames from time step t_2 to t_{12}. It is worth to mention that RFs order labelled by 'A' and 'B' are similar because their first two RFs are the same (T_0 then S_0) and they always have the most contribution of block matching prediction (the same concept applies to labels 'C' and 'D'). The shaded cells in Table IV and V show consecutive frames within the same view (temporal frames) that should be coded using different RFs orders. Also, it can be inferred that the suitable RFs order would be predicted in most cases, using previous frames within the same view.

4. ADAPTIVE REFERENCE FRAME RE-ORDERING ALGORITHM

Section 3 shows that the order of RFs is predicted using the corresponding information from the recent temporal frames. The flow of the proposed algorithm is presented in Figure 3. For a P-frame, it checks first if the frame is located in a position where partial of RFs are available (transient state e.g. all P-frames in first time slice; t_0). In this stage, the algorithm uses predefined prediction architecture to encode the frame where the prediction structure involves available RFs with their initial order. In a non-transient scenario, the algorithm loads the corresponding order of RFs then encodes the P-frame using that order. After that, the algorithm loops on all its macroblocks to compute the block matching statistics among all RFs. When there is no scene change, the algorithm orders the reference frames based on their block matching statistics and its new order will be stored and applied to the next temporal frame.

When scene changing, the majority of frame's macroblocks in the new scene are intra-predicted. Hence the algorithm is relying on the number of intra-coded macroblocks to detect scene changes. If the percentage of intra-predicted macroblocks exceeds certain threshold (60%) [13], then the following P-frames will use similar RFs order to the corresponding P-frames in transient state (e.g. following frame in coding order will use RFs order where spatial RFs are placed first in DPB). In other word, following frames that are located within the same time slice when scene changes, will use RFs order where spatial reference frames are placed first in list0.

5. EXPERIMENTAL RESULTS

The proposed algorithm has been evaluated in encoding MVVs, (Break-dancers, Ballet, Race1 and Exit) and also encoding a sequence that contains a number of scene changes at low bitrate. The proposed algorithm has been implemented using prediction structures reported by Sheikh Akbari *et al*. [10] and Bilen *et al*. [8] as they clearly highlighted the order of the selected reference frames in their reported prediction structures.

In the first scenario, the algorithm uses the prediction structures proposed in [10] for coding four different MVVs at low bitrates. This prediction structure contains five reference frames with two different reference frame orders. Figure 3-c presents the first reference frame order where spatial and spatiotemporal RFs have higher priority than the temporal frames (Mode 1 in [10]). Figure 3-b places temporal reference frames in the beginning of the other reference frames (Mode 3 in [10]). The proposed algorithm starts with the same order of reference frames that was suggested in each Mode. P-frame located in time slice below t_3 will be coded using the available reference frames (transient state). After t_3, the algorithm starts to adapt the reference frames re-ordering dynamically. Figure 4 shows

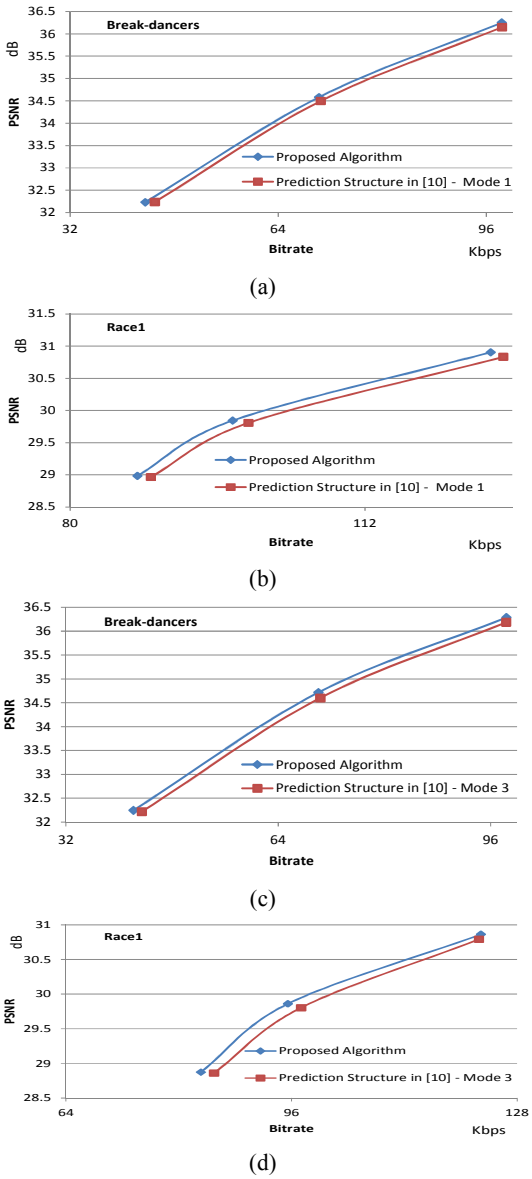

Figure 4. Coding performance of the MVC using the proposed algorithm on the prediction architectures proposed by Sheikh Akbari *et al*. [10] using: a-b) Mode 1 and c-d) Mode 3.

the coding performance of the MVC using the proposed adaptive re-ordering algorithm in coding Break-dancers and Race1 MVV datasets in comparison to RFs order proposed in [10]. From Figure 4, it can be seen that the proposed algorithm gives higher coding performance compared to the use of static RFs orders (up to 0.2 dB).

In the second scenario, the proposed RFs re-ordering algorithm and the prediction structure reported in [8] (Mode 3 is shown in Figure 1-a) are used to code a sequence with scenes changes. Results are shown in Figure 5. From figures 5 and 6, it can be seen that the proposed algorithm gives

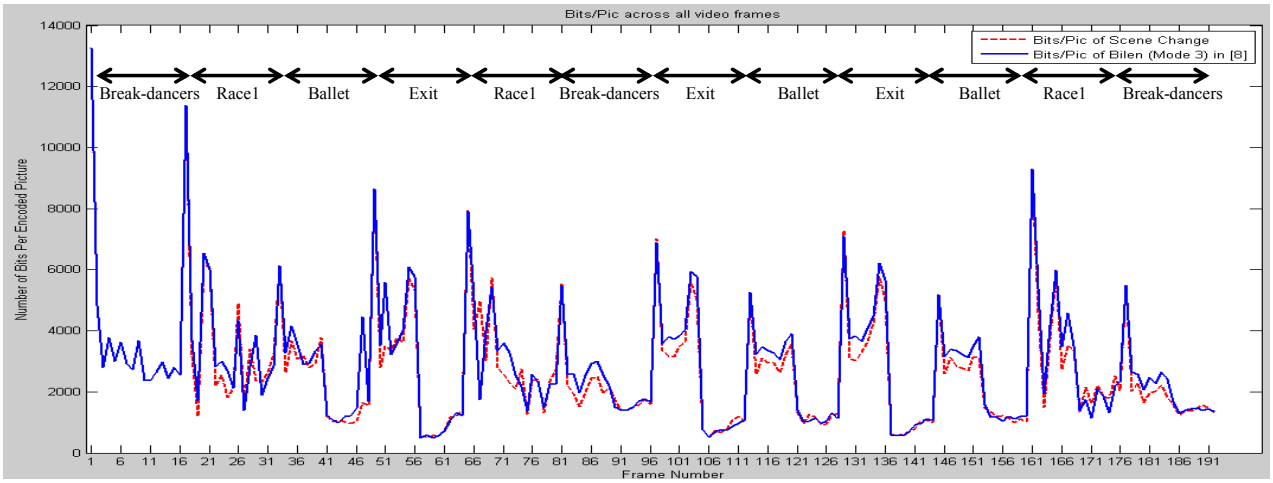

Figure 5. Number of bits per coded picture when using prediction structure proposed in [8] and the proposed algorithm.

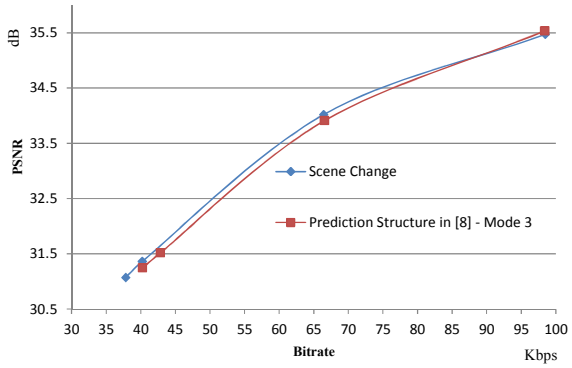

Figure 6. Coding performance for the proposed algorithm using the prediction structure proposed by Bilen *et al.* [8].

slightly higher coding performance compared to the use of static RFs order (as shown in Figure 6), at the same time it saves significant bitrates, up to 6.2%.

6. CONCLUSIONS

An adaptive reference frames re-ordering was proposed. The proposed algorithm updates the reference frame orders adaptively using the statistics of block matching. The proposed re-ordering algorithm gives superior coding performance compared to the state of arts (up to 0.2 dB). In addition, it efficiently re-orders reference frames when dealing with scene changes and saves bitrates of up to 6.2%.

7. REFERENCES

[1] M. Tanimoto, "FTV: Free-viewpoint Television," *Signal Processing: Image Communication*, vol. 27, no. 6, pp. 555–570, Jul. 2012.

[2] A. Smolic, "3D video and free viewpoint video - From capture to display," *Pattern Recognition*, Elsevier, vol. 44, no. 9, pp. 1958–1968, Sep. 2011.

[3] J. Seo and K. Sohn, "Early disparity estimation skipping for multi-view video coding," *EURASIP Journal on Wireless Communications and Networking*, vol. 2012, no. 1, Feb. 2012.

[4] A. Vetro, T. Wiegand, and G. J. Sullivan, "Overview of the Stereo and Multi-view Video Coding Extensions of the H.264/MPEG-4 AVC Standard," *Proceedings of the IEEE*, vol. 99, no. 4, pp. 626–642, Apr. 2011.

[5] U. Fecker and A. Kaup, "H.264/AVC-Compatible Coding of Dynamic Light Fields Using Transposed Picture Ordering," *EUSIPCO 2005*, Antalya, Turkey, Sept. 2005.

[6] H. Said, A. Sheikh Akbari and, M. M. Abbas Malik, "H.264/AVC based Stereoscopic Video Coding Scheme using the Statistics of Block Matching," accepted for publication in 36th International Conference on Telecommunications and Signal Processing TSP 2013, July 2013.

[7] H. Said and, A. Sheikh Akbari, "H.264/AVC Based Multi-view Video Codec using the Statistics of Block Matching," accepted for publication in 55th International Symposium ELMAR-2013, Sept. 2013.

[8] C. Bilen, A. Aksay, and G.B. Akar, "A Multi-View Video Codec Based on H.264," *in 2006 International Conference on Image Processing*, pp. 541–544, Oct. 2006.

[9] S. Hong and Y. Yu, "Dynamic reference frame reordering for frame sequential stereoscopic video encoding," Patent, US 20110109721, Sony Corporation, Jul 2012.

[10] A. Sheikh Akbari, N. Canagarajah, D. Redmill, D. Agrafiotis, "A Novel H.264/AVC Based Multi-View Video Coding Scheme," *3DTV Conference 2007*, pp.1-4, 7-9 May 2007.

[11] Y. Zhang, S. Kwong, G. Jiang, and H. Wang, "Efficient Multi-Reference Frame Selection Algorithm for Hierarchical B Pictures in Multi-view Video Coding," *IEEE Transactions on Broadcasting*, vol. 57, no. 1, pp. 15–23, Mar. 2011.

[12] Y. Chen, Y.-K. Wang, K. Ugur, M. M. Hannuksela, J. Lainema, and M. Gabbouj, "The Emerging MVC Standard for 3D Video Services," *EURASIP Journal on Advances in Signal Processing,* vol. 2009, no. 1, 2009.

[13] J. Brandt, J. Trotzky, L. Wolf, "Fast Frame-Based Scene Change Detection in the Compressed Domain for MPEG-4 Video," *in Proc. of Next Generation Mobile Applications, Services and Technologies, 2008. NGMAST '08*, pp.514-520, Sept. 2008.