Signal Processing: *Image Communication* 26 (2011) 130–142

Contents lists available at ScienceDirect

Signal Processing: *Image Communication*

journal homepage: www.elsevier.com/locate/image

Encoder adaptable difference detection for low power video compression in surveillance system

Xin Jin [a,*], Satoshi Goto [b]

[a] *Information Technology Research Organization, Waseda University, Fukuoka, Japan*
[b] *Graduate School of Information, Production and Systems, Waseda University, Fukuoka, Japan*

ARTICLE INFO

Article history:
Received 16 November 2009
Accepted 25 January 2011
Available online 4 February 2011

Keywords:
Difference detection
Low power video compression
Surveillance video compression
Video coding

ABSTRACT

As a state-of-the-art video compression technique, H.264/AVC has been deployed in many surveillance cameras to improve the compression efficiency. However, it induces very high coding complexity, and thus high power consumption. In this paper, a difference detection algorithm is proposed to reduce the computational complexity and power consumption in surveillance video compression by automatically distributing the video data to different modules of the video encoder according to their content similarity features. Without any requirement in changing the encoder hardware, the proposed algorithm provides high adaptability to be integrated into the existing H.264 video encoders. An average of over 82% of overall encoding complexity can be reduced regardless of whether or not the H.264 encoder itself has employed fast algorithms. No loss is observed in both subjective and objective video quality.

1. Introduction

Video surveillance has become a very important tool in our daily life to improve the public safety. During the last decade, law enforcement agencies in Great Britain, France, Monaco, Spain and other countries have increasingly relied on the close circuit television (CCTV) surveillance system to enhance public security [1]. According to the report, CCTV surveillance system is successful in reducing and preventing crimes [2,3]. It has become increasingly popular in many countries all over the world.

Fig. 1 shows a typical diagram of a video surveillance system. As depicted in the figure, a video surveillance system generally consists of two parts: a client side and a server side. At the client side, a number of surveillance cameras are installed scattering over an area to capture the video scenes. Using the CCTV system in United Kingdom as an instance, there are around 4.2 million surveillance cameras covering every corner of the country [4]. At the server side, the video data captured by the clients are received, processed and stored.

Because of the huge size of raw video data, especially at a high definition (HD) resolution, video compression techniques have been integrated into surveillance cameras to reduce data storage. As a state-of-the-art compression technique, H.264/AVC [5] has been deployed in many surveillance cameras recently to further improve the compression efficiency. Compared with video compression standards of the previous generation that were deployed in the surveillance cameras before (e.g. Motion JPEG [6], MPEG-1 [7], MPEG-2 [8], MPEG-4 [9], H.261 [10] and H.263 [11]), H.264/AVC provides much higher coding efficiency by fully exploiting both spatial and temporal redundancies. Enhancement tools, like variable block-size motion compensation (MC), quarter-sample-accurate MC, multiple reference picture, directional Intra prediction, in-loop deblocking filtering, context-adaptive binary arithmetic coding (CABAC), context-adaptive variable-length coding (CAVLC), etc., endue H.264/AVC with approximately a 50%

* Corresponding author.
E-mail addresses: xjin@aoni.waseda.jp (X. Jin), goto@waseda.jp (S. Goto).

0923-5965/$ - see front matter © 2011 Elsevier B.V. All rights reserved.
doi:10.1016/j.image.2011.01.002

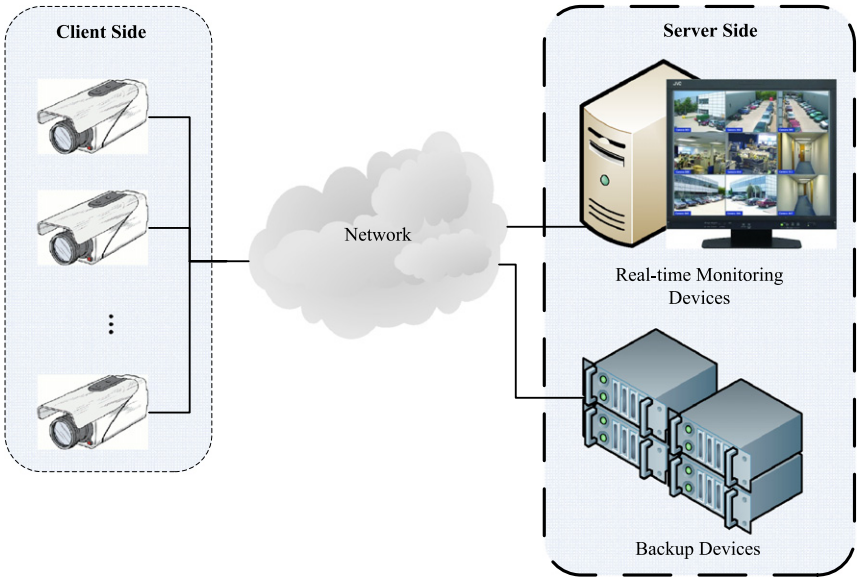

Fig. 1. Diagram of video surveillance system.

bit rate savings for equivalent perceptual quality relative to the performance of prior standards [12].

However, as a trade-off of achieving such a significant improvement in coding efficiency, the complexity of H.264/AVC encoding is roughly three times higher than the prior video coding standards [13]. Such high coding complexity induces high power consumption to the H.264/AVC encoders. Generally, H.264/AVC encoder consumes around 50% of the total power in a surveillance camera. As the video resolution is increasing from the standard definition (SD) to HD, and the number of video cameras is increasing from thousands to millions, reducing the video coding complexity for the surveillance system has drawn great attention in both industry and research areas.

Techniques have been developed to reduce the video encoding complexity in the video surveillance system. Those techniques can be classified into two categories: one is the object-based coding schemes [14–16] and the other is the decoder complexity trade-off based schemes [17,18]. The object-based coding schemes segment the objects from the input video to assign different coding schemes to different object types. The moving foreground objects in the stationary camera surveillance video are detected by a background subtraction technique and encoded by MPEG-4 object-based coding in Ref. [14]. The state of each pixel is analyzed to distinguish the foreground and background, and a Lempel-Ziv-Huffman codec is used to encode the foreground objects in Ref. [15], which shows a 15% bit saving compared to MPEG-4. Venkatraman and Makur [16] applied compressive sensing code to encode the object prediction error. Although the object-based coding schemes can efficiently handle the video contents with a long period of static scenes, such as videos captured over a parking lot, they are not efficient for the cases with a lot of moving objects, such as traffic scenes captured on a busy road. The background information is usually needed as prior information for background subtraction or object segmentation. Model training or human decision is introduced, which is not a fully automatic processing for the surveillance video cameras with real-time encoding requirements [19–23]. Additionally, the complexity consumed by the object segmentation is not always negligible if accurate results are desired for compression fidelity. The latest implementation results showed that the generally used background modeling method, GMM, for offline object segmentation, consumes around 36 ms in processing one 320×256 image on a PC with 2.5 GHz Duo Core Processor and 4 GB RAM [24]. The complexity is too high to be integrated into an encoder with real-time encoding speed (30 frames per second) requirement.

The decoder complexity trade-off based schemes shift the coding complexity from the encoder to the decoder by doing the motion estimation (ME) on the decoder side, e.g. Wyner-Ziv coding, to reduce the complexity. They try to provide a simple encoder and a complex decoder solution for surveillance system. Yaman and AlRegib [17] uses a shape-adaptive Set Partitioning in Hierarchical Trees (SPIHT) encoder to code the decoder unpredictable region, and performs ME on the decoder by exploiting temporal coherency over the successive decoded frames to obtain an estimation for the other regions. In Ref. [18], a backward-channel aware Wyner-Ziv video coding approach is proposed to provide similar coding performance with H.264/AVC Intra coding while maintaining the low complexity at the encoder. Although the complexity trade-off schemes can make the encoder a bit simpler, performance degradation in the compression efficiency is always a problem. If a better performance is desired, a considerable complexity increase on the decoder side is required. Such a complexity increase will be a heavy burden on the server side if the video data from a tremendous amount of surveillance cameras need to be processed. Furthermore,

because of the complete change in the coding architecture, it is difficult to integrate the schemes of the second category into the existing surveillance cameras. The requirement of changes in both software and hardware highly limits their encoder adaptability.

In this paper, a technique with low complexity and high encoder adaptability is proposed to reduce computational complexity of H.264/AVC video compression in the surveillance systems. The proposed algorithm, named as difference detection, directly detects the content differences in the input frame by analyzing the color and moving correlation feature of video content. The detection process is performed on a macroblock-by-macroblock basis according to the video encoding order. According to its content feature, a macroblock (MB) is automatically assigned to the corresponding module of the video encoder. Both computational complexity and storage complexity of the difference detection itself are negligible compared to those of the video encoder. The proposed scheme can be easily integrated into the existing H.264/AVC encoding systems by only updating the high-level control software via simply adding a new task. No original encoder hardware needs to be changed for implementation. The proposed algorithm can effectively reduce the overall video encoding complexity by over 82% on average, regardless of whether or not the encoder itself has employed fast algorithms. Furthermore, since the proposed algorithm can produce more skipped macroblocks, the compression efficiency can be further improved for some of the video surveillance sequences.

The rest of this paper is organized as follows. The proposed difference detection algorithm is described in detail in Section 2. The implementation complexity of the proposed algorithm together with the power saving introduced by it are analyzed in Section 3. Experimental results are shown in Section 4 followed by conclusions in Section 5.

2. Difference detection

2.1. System architecture

The system architecture of the proposed algorithm integrated with a video encoder is shown in Fig. 2. As depicted in the figure, each input MB is first processed by a Difference Detector (DD) to decide whether any content difference exists in comparison to the collocated MB in the previous frame. If a difference is detected, the MB will be passed to the Mode Decision module of the encoder; otherwise, the MB will be directly passed to the Bit Stream Writer of the encoder without doing the mode decision. To be compliant with the H.264/AVC standard, the MB without content differences is written as a skip-mode [5] coded MB into the compressed bit-stream.

Mode decision module of an H.264/AVC encoder generally consists of Inter/Intra mode costs generation and Inter/Intra mode decision. Its actual decision flow is determined by the implementation of the encoder. Some encoders may use rate-distortion-optimization (RDO) based brute-force mode search to select the best coding mode out of all of the prediction modes for each MB; some may use a sub-optimal mode decision scheme, such as the one implemented in the reference software JM15.1[25]. Since the Mode Decision Module consumes the highest complexity in H.264/AVC encoding, many implementations of H.264/AVC encoder introduce fast algorithms for Mode Decision. These fast algorithms are applied from the best mode candidates selection level, e.g. the selective Intra mode decision in Refs. [26,27] and the fast Inter mode selection in Ref. [26,28–30], down to the prediction generation level, e.g. unsymmetric-cross multi-Hexagon-grid search (UMHexagonS) [31] and enhanced predictive zonal search (EPZS) [32]. They introduce early detection of prediction mode or early termination in motion estimation to accelerate the decision flow. However, no matter which mode decision scheme is used in the encoder, it is transparent to the proposed Difference Detection, or in other words, Difference Detection is adaptable to any implementation of Inter/Intra mode decision.

2.2. Difference detection flow

Difference Detection processing flow is depicted in Fig. 3 in detail. As shown in the figure, Difference Detection mainly consists of two parts: chrominance feature retrieval and comparison (CFRC); and motion feature retrieval and comparison (MFRC).

CFRC exploits chrominance feature to evaluate the content similarity. Using chrominance feature instead of luminance feature is based on two observations: chrominance signal presents higher value consistency than the luminance signal; using chrominance signal presents lower computational complexity than using luminance signal. The first observation is achieved by comparing the pixel values of successive frames of surveillance videos,

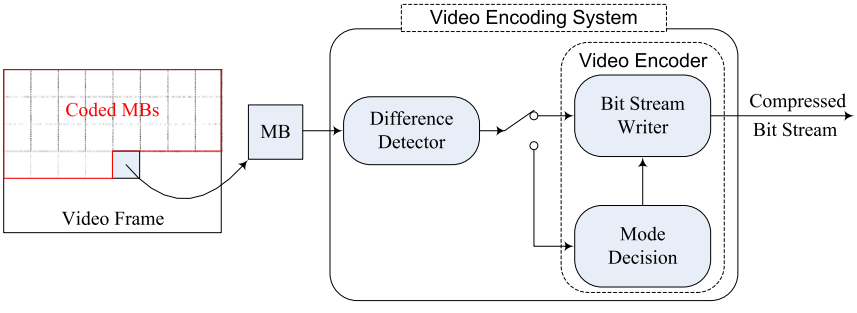

Fig. 2. System architecture of Difference Detection integrated with a video encoder.

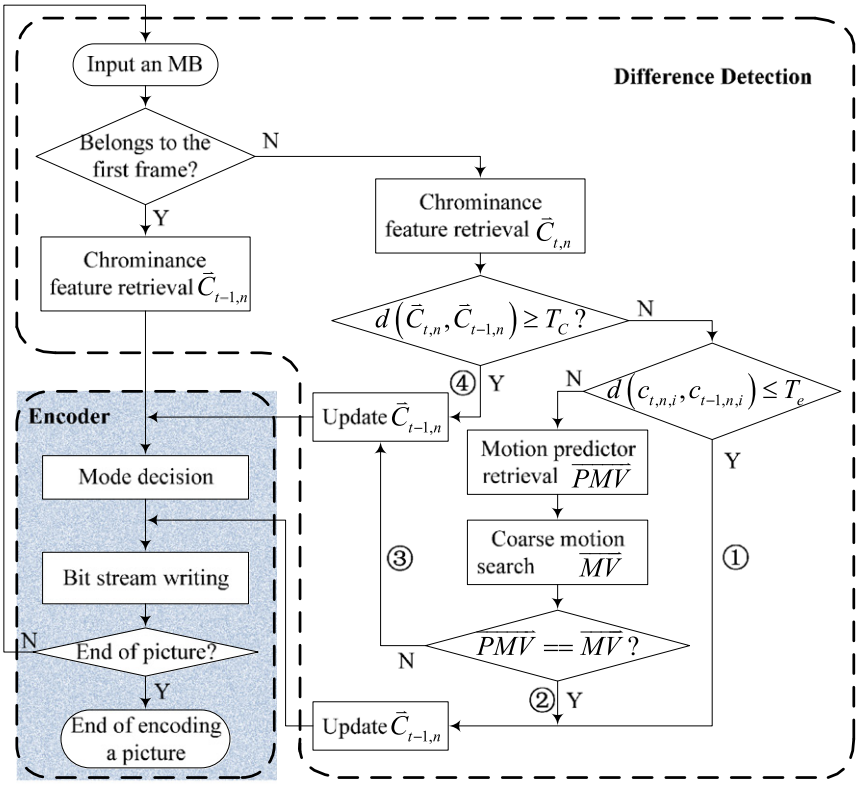

Fig. 3. Difference Detection flow.

which are visually the same, for luminance image and chrominance images. Two successive frames, which are visually the same, are expected to have the same pixel values both in the luminance image and in chrominance images. However, as one frame is subtracted from the other, it is found that the pixel-to-pixel value difference is not always equal to 0. Such value variation is caused by environment noise and lens noise during video capturing, which is very general for video surveillance system. Statistical results showed that an average of 10.37% pixels in luminance signal has non-zero pixel-to-pixel value difference for the contents which are visually the same. The pixel value difference varies from -7 to 9. While, for chrominance signal, only an average of 2.01% pixels have non-zero pixel-to-pixel value difference and the difference variation range is -3 to 4. So, using chrominance feature is easier to mitigate the impacts of noise to find out the area with the same content. And, for the general used color format, 4:2:0, in video surveillance, the size of the chrominance MB is quarter of luminance MB. So, using the feature from two chrominance features consume only half of the complexity that needed by using luminance feature. Because of the higher consistency in pixel value and the lower requirement on computational complexity, chrominance feature is selected to be used in the proposed algorithm to evaluate the content similarity.

The chrominance feature of the current MB n in the current frame t (denoted as $\vec{C}_{t,n}$ in Fig. 3) is retrieved and compared with that of the collocated MB n in the previous frame $t-1$ (denoted as $\vec{C}_{t-1,n}$ in Fig. 3). $\vec{C}_{t,n}$ is an m-dimensional vector, which is given by $\vec{C}_{t,n} = (c_{t,n,0}, c_{t,n,1}, \ldots, c_{t,n,i}, \ldots, c_{t,n,m-1})$. $c_{t,n,i}$ represents average energy of ith chrominance component of the MB. Since the direct current (DC) coefficient shows the average energy of the block, $c_{t,n,i}$ is approximated by the summation of the pixels of the component. For the video data using YUV format, m is equal to 2 to represent U and V components. $d(\vec{C}_{t,n}, \vec{C}_{t-1,n})$ is the norm distance between $\vec{C}_{t,n}$ and $\vec{C}_{t-1,n}$. If the distance of any of the two chrominance components exceeds a pre-determined threshold T_C, the MB is decided to have content differences, and will be passed to the Mode Decision module of the encoder for searching the best coding mode.

For the MB with similar color as decided by T_C, the distance between each corresponding component of $\vec{C}_{t,n}$ and $\vec{C}_{t-1,n}$, denoted as $d(c_{t,n,i}, c_{t\text{-}1,n,i})$ in Fig. 3, is further compared with T_e. As mentioned above, even though the content looks identical, the pixel value may be different because of the noise introduced by capturing devices and environment. An instance is shown in Fig. 4. The two consecutive frames, frame 20 and 21, look identical both in their color chart (YUV images in Fig. 4(a) and (b)) and U component (Fig. 4(c) and (d)). However, the residual U image between frame 20 and frame 21, shown in Fig. 4(e), does not

Fig. 4. A surveillance video captured over a park lot: (a) frame 20; (b) frame 21; (c) U component of frame 20; (d) U component of frame 21; and (e) U residual image generated by (d)-(c), the gray pixels represents value 0, the whiter the larger residual value is.

only consist of zero value pixels. The non-zero pixels are caused by the environment noise during capturing. So, T_e is larger than zero to overcome such noise influence.

For the MB whose chrominance distance is larger than T_e, it is passed to the MFRC block to further determine whether the small color difference is caused by noise or content difference. The MFRC retrieves both a motion vector predictor, $\vec{PMV}$, and a motion vector, $\vec{MV}$, for the MB. $\vec{PMV}$ is formed by the motion vectors of nearby, coded partitions immediately above, diagonally above and to the right, and immediately left of the current partition or sub-partition, which is defined in H.264/AVC [5]. $\vec{MV}$ is obtained by integer-pixel accuracy coarse motion search for a 16×16 partition. The comparison of those two motion vectors determines the motion correlation of the MB to its neighboring MBs. The MB is passed to Mode Decision module of the encoder if $\vec{PMV}$ is different from $\vec{MV}$.

Thresholds T_C and T_e are two chrominance energy difference constraints to classify MBs according to color feature. Detection flow shown in Fig. 3 reveals that: T_C is used to find out the MB whose content is quite different from that of the collocated MB; T_e tends to find out the MB which is visually the same with the collocated one. Value of T_C and T_e presents a kind of trade-off between complexity reduction and compression efficiency. Smaller T_C results in passing more MBs through the original encoding path, Path ④ in Fig. 3. The reduction in computational complexity becomes lower, but the compression efficiency will be better. While, larger T_e provides higher computational complexity reduction with lower compression efficiency.

Investigation results showed that although the chrominance energy varies with video contents, the chrominance energy difference in distinguishing content similarity can be content independent to avoid content based training. Fig. 5 provides an instance in comparison of two different video contents (one is captured over a parking lot (Fig. 5(a) and (b)), the other is captured for a basketball game (Fig. 5(d) and (e))). The average chrominance energy, $c_{t,n,0}$ and $c_{t,n,1}$, are calculated for four pairs of collocated MBs (two pairs are identical-looking MBs, circled by green line; the other two are different-looking MBs, circled by red line). As shown in the figure, although there is an obvious difference in average energy between the two video contents (comparing the vertical axis of Fig. 5(c) with that of (f)), the energy difference between the identical-looking MBs (e.g. MB 26 and MB 36) is still close to zero for both of the two contents. For the different-looking MBs (MB 171 and MB 198), the energy difference between the corresponding chrominance components is obvious, which is easy to be distinguished.

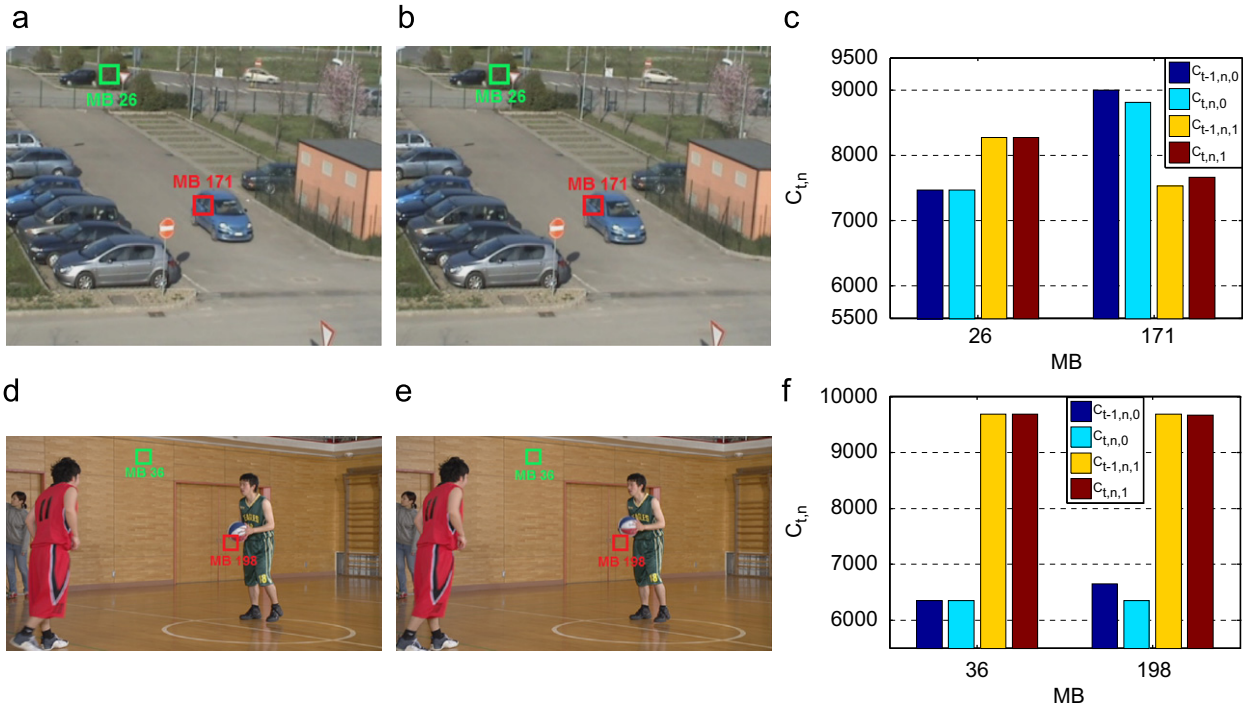

Fig. 5. Chrominance energy, $c_{t,n,0}$ and $c_{t,n,1}$, comparison for MBs with and without content difference for two video contents. (c) shows U and V average energy of MB 26 and MB 171 for frame (a), $c_{t\text{-}1,n,0}$ and $c_{t\text{-}1,n,1}$, and frame (b), $c_{t,n,0}$ and $c_{t,n,1}$; (f) shows those of MB 36 and MB 198 for frame (d) and frame (e). (For interpretation of the references to color in this figure, the reader is referred to the web version of this article.)

Consequently, statistical results are retrieved for the MBs looked same and differently for different video contents. Based on the results, a constant value, 20, is set to T_C and 2 is set to T_e, respectively. Using these two thresholds, an average of 94% of different-looking MBs and an average of 87% of identical-looking MBs can be distinguished effectively. For other MBs with energy differences between T_C and T_e, motion feature will be further compared to determine the content similarity. Experimental results shown in Section 4 also proved that using the constant values set to T_C and T_e provides an attractive complexity reduction for all the test sequences with negligible compression efficiency degradation. For some sequences, they can further improve the compression efficiency because of a big saving in coding bits.

For the video sequences with the color spaces other than 4:2:0, e.g. 4:2:2, the proposed algorithm can also be applied. While, threshold T_C and T_e need to be increased because of an increase in the chrominance energy.

3. Complexity analysis

In this section, the implementation complexity of the proposed algorithm together with the power saving introduced by it are analyzed in detail both from theoretical and experimental point of view.

3.1. Implementation complexity analysis

For the computational complexity of DD: CFRC introduces three comparisons with 130 additions/subtractions, and the additions can be processed in parallel; MFRC consumes 8 additions/subtractions and at most 14 comparisons in motion vector predictor retrieval. 16×16 integer-pel motion search can be easily implemented in parallel.

For the storage complexity, DD needs to have a buffer to store the chrominance feature (2 bytes/element) of the previous frame. The corresponding entry in the buffer is updated immediately after the current MB of the current frame is processed. For commonly used VGA size (640×480 pixels) surveillance video, the storage requirement of DD is 2.4 K bytes.

To evaluate the largest complexity overhead introduced by the proposed algorithm, "Path 3" of DD (denoted as ③ in Fig. 3), the path with the highest complexity, is integrated into H.264/AVC reference software JM15.1 [25], and force all of the MBs go though it to the Mode Decision module of the encoder to test its computational complexity. Following the recommended simulation common conditions defined in [33], the "IPPP…" prediction structure is tested for 7 surveillance sequences with different resolutions and different content features. Some important configuration parameters of JM15.1 together with the test materials are listed in Table 1.

All of the MBs of each sequence are forced to go through "Path 3" to the Mode Decision module of the encoder. As listed in Table 1, the seven sequences and four QP values result in 28 experiments for each testing case (e.g. JM15.1 and JM15.1 with Path 3 of DD). All the experiments are conducted on a PC with 3.16 GHz Intel Core™2 Duo CPU and 3.25 GB RAM. The similar method introduced in

Table 1
Testing materials and major coding parameter settings according to common test conditions [33].

	Resolution	Sequence name	Frames to be encoded
Testing materials	CIF (352 × 288)	Hall Monitor (H-M) Intersection01 (IS01) [34] 1.14abandoned object (AB114) [34] from frame 50	150
	QVGA (320 × 256)	StoppedVehicle0 (SV0) [35] StoppedVehicle3 (SV3) [35]	150
	VGA (640 × 480)	Street	150
	SXGA (1392 × 1040)	Hermes_Outdoor_cam1 (HOC1) [35] from frame 600	60
Coding parameter settings	Profile	Main Profile	
	GOP	IPPP…	
	Intra period	0 (first image only)	
	Number of reference frames	4	
	Rate distortion optimization	1	
	EarlySkipEnable	0	
	SelectiveIntraEnable	0	
	Search range	64	
	Search mode	EPZS	
	CABAC	On	
	Deblocking filter	On	
	Quantization parameter (I/P)	22/23, 27/28, 32/33, 37/38	

Table 2
The largest computational complexity overhead introduced by DD (The statistical results are retrieved by forcing all of the MBs to go through "Path 3" of DD to the Mode Decision module of the encoder).

Seq.	QP (I)	Average total running time (s)		Average total running time increase ratio (%)
		JM	Path 3+JM	Path3+JM vs. JM
H-M	22	495.40	497.48	0.42
	27	454.99	454.07	-0.21
	32	429.61	431.85	0.50
	37	418.02	421.42	0.82
IS01	22	479.22	483.40	0.86
	27	456.00	455.32	-0.14
	32	437.90	441.67	0.87
	37	431.13	432.36	0.28
AB114	22	461.49	460.89	-0.13
	27	444.72	446.84	0.47
	32	432.95	432.78	-0.04
	37	426.10	426.89	0.18
SV0	22	355.83	356.89	0.31
	27	342.92	343.73	0.23
	32	341.39	344.09	0.79
	37	334.23	337.57	1.00
SV3	22	372.03	374.35	0.63
	27	352.66	356.51	1.09
	32	347.38	350.59	0.92
	37	342.09	345.81	1.09
Street	22	1444.83	1449.27	0.31
	27	1382.97	1404.11	1.53
	32	1349.88	1355.07	0.39
	37	1310.99	1333.05	1.67
HOC1	22	2464.61	2453.24	-0.46
	27	2380.37	2372.51	-0.33
	32	2301.03	2347.17	2.00
	37	2283.08	2329.82	2.06
Average				**0.611**

Ref. [36] is used for each experiment that: each experiment is conducted five times in a row, the total running time consumed by both Path 3 of DD and video encoder is recorded, and the results of the last four replications are averaged and saved. We also disabled the writing of the reconstructed and compressed file to minimize the effect of I/O operations on the execution time, thus focusing on measuring the encoding complexity. The average total running time is compared between JM15.1 (JM) and JM15.1 with path3 of DD (Path3+JM) in Table 2.

As shown in the table, the longest path of DD only introduces an average of 0.611% complexity overhead compared to JM itself, which indicates that even if the worst case happens, that all of the MBs are decided to go through Path 3 to the H.264/AVC Mode Decision Module, the complexity increase is still negligible for the whole encoding system.

The mainstream video surveillance chips normally consist of a hardware accelerator (HWA) based video coding sub-system which provides high performance video capture, and programmable processors which enable customers to create differentiations in their products, such as implementation different rate control algorithms, region of interests, etc. The proposed technique, DD, is very friendly to mapping to such a chip architecture. It only requires to implement the DD

Table 3
Average ratio of MBs going through each path of DD.

Seq.	Ratio of MBs (%)			
	Path 1	Path 2	Path 3	Path 4
H-M	66.87	0.66	0.87	31.59
IS01	68.03	3.23	5.44	23.31
AB114	68.72	4.71	7.14	19.43
SV0	98.28	0.21	0.22	1.29
SV3	98.35	0.24	0.16	1.24
Street	83.44	0.47	1.03	15.06
HOC1	81.38	3.91	6.71	8.00
Average	**80.72**	**1.92**	**3.08**	**14.27**

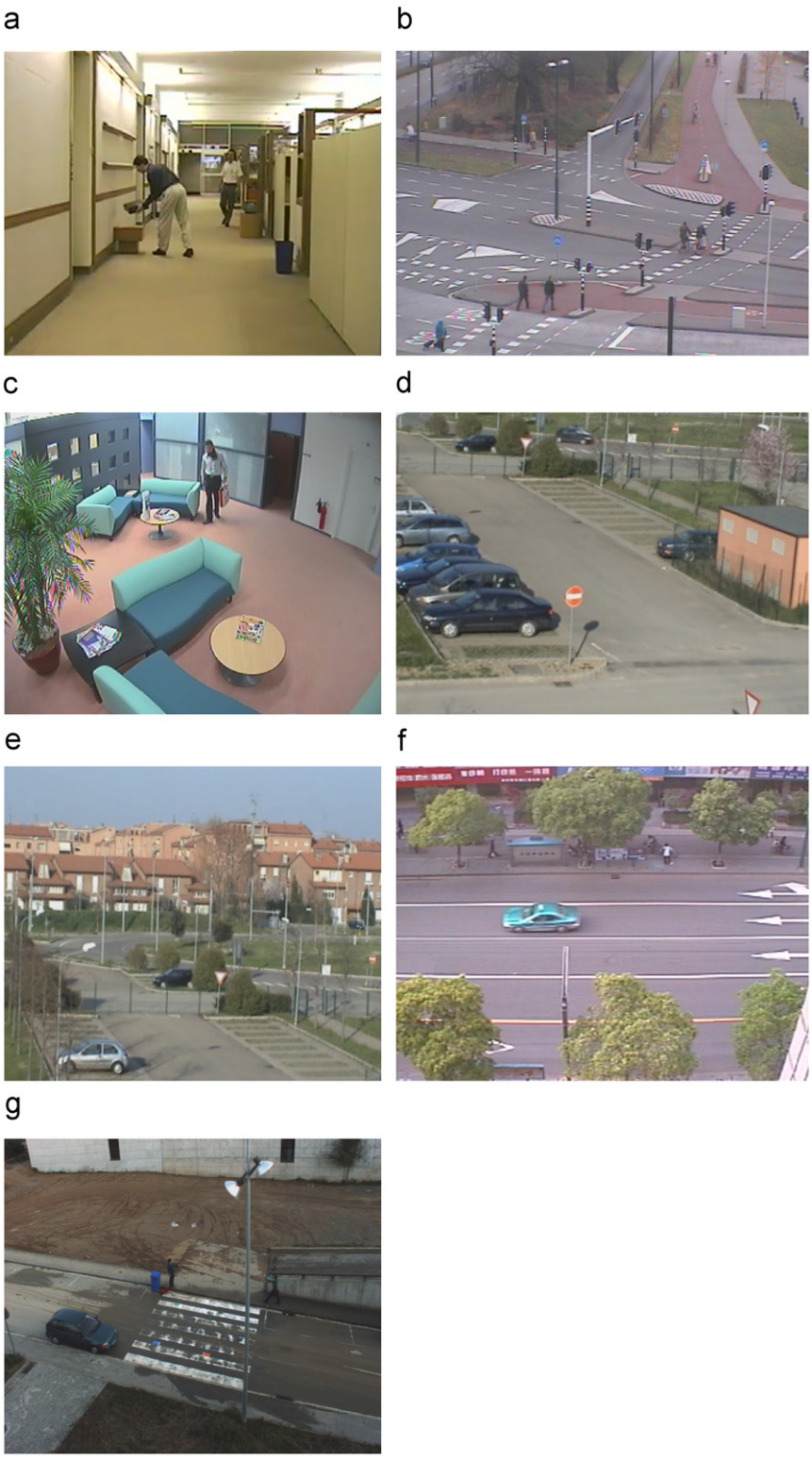

Fig. 6. Testing materials: (a) H-M; (b) IS01; (c) AB114; (d) SV0; (e) SV3; (f) Street and (g) HOC1.

on the programmable processors and to change the high-level control software to have the DD operated with the video encoder, without changing the hardware parts of the video encoder. The host will still invoke the video encoder at a picture by picture basis to avoid excessive switch overhead.

Consequently, both the computational complexity of DD and its cost in changing the existing system is low to integrate it into video surveillance cameras.

3.2. Power saving analysis

Using the flow described above, the complexity of the encoding system can be significantly reduced for surveillance video coding. If an MB goes though "Path 1" (denoted as ① in Fig. 3) to the Bit Stream Writer, the power consumed in both DDR data transfer and logic part is saved. The ratio of MBs going through "Path 1"depends only on the video content, and is not influenced by the coding bit rates or QP values. For the MBs going through "Path 2" (denoted as ② in Fig. 3), although the integer-pel-ME is performed for the 16×16 partition, the logic part of power consumption is still saved. The averaged ratios over four QP values, listed in Table 3, reveal that an average of 80.72% MBs of the surveillance videos directly goes through "Path 1" to the Bit Stream Writer, which significantly reduces power consumption by skipping the entire encoding process. Furthermore, an average of 1.92% MBs will go through "Path 2". For the surveillance video with a long period of a static scene, like SV0 and SV3, over 98% of MBs will go through "Path 1", which leads to significant reduction in power consumption.

4. Experimental results

4.1. Testing conditions

The proposed Difference Detection algorithm is integrated into JM15.1 to test its performance (T_C=20, T_e=2). The same common test conditions [33] and the testing materials are used as that listed in Table 1.

As listed in Table 1, seven surveillance sequences with different resolutions and different features are tested. Features include a long period of a static scene (e.g. SV0 and SV3 that captured over parking lots), various motions (e.g. IS01 and Street that captured over busy streets), burst motion (e.g. H-M and AB114 that captured at lobbies), etc. The sequences selected in our experiments are shown in Fig. 6.

Compression performances in terms of objective quality, computational complexity, and subjective quality are compared. The objective quality is evaluated by BD-PSNR and BD-Bitrate [37]. The computational complexity is evaluated by average running time reduction ratio (ARTRR). The subjective quality is evaluated by average difference in visual information fidelity (VIF) [38].

The same testing system together with the same testing method described in Section 3.1 is performed for each experiment: each experiment is conducted five times in a row on a PC with 3.16 GHz Intel Core^{TM}2 Duo CPU and 3.25 GB RAM, the total running time consumed by both DD and video encoder is recorded, and the results of the last four replications are averaged and saved. The running time reduction ratio of testing case i against testing case j, such as the JM15.1 with our proposed algorithm against the original JM15.1, for an experiment is given by $r=(AT_j-AT_i)/AT_j$, where AT_i and AT_j are averaged total running time of testing case i and testing case j, respectively, over four replications. Then, ARTRR is the averaged r over four QP values.

VIF value is calculated for each decoded video using the original video (uncompressed video) as a reference. It models the distorted image (a decoded video frame) as the consequence of passing its reference image (the corresponding uncompressed video frame) through distortion channels. Then, the mutual information between the input and output of human visual system (HVS) channel are quantified for both the distorted image and the reference image by passing each of them through the HVS channel. Finally, the two mutual information are combined to form a VIF measure for visual quality. For all practical distortion types, VIF value is within the range from 0 to 1. The larger the value is the better the subjective quality is. VIF value is calculated using the pixel domain implementation released in Ref. [39] by the algorithm designer. The VIF differences between two testing cases are calculated and averaged over four QP values for each sequence.

4.2. Testing results

Since the proposed algorithm determines some MBs as skip-mode coded MBs prior to the mode decision and motion estimation, its performance is first compared with another two representative early skip mode decision (ESMD) algorithms: one is a ME-relied ESMD approach proposed in Ref. [26] (ES) and also implemented in JM15.1 as an option of fast algorithm; the other is a cost-mode-relied algorithm proposed in Ref. [28] (MAMD), which performs ESMD prior to Mode Decision using a QP-based cost model. The objective

Table 4
Objective quality and computational complexity comparison among ES [26], MAMD [28] and the proposed algorithm.

Seq.	ES [26] vs. JM			MAMD [28] vs. JM			Proposed vs. JM		
	ARTRR (%)	BD-PSNR (dB)	BD-Bitrate (%)	*ARTRR* (%)	BD-PSNR (dB)	BD-Bitrate (%)	*ARTRR* (%)	BD-PSNR (dB)	BD-Bitrate (%)
H-M	28.25	−0.058	0.709	59.24	−0.053	1.932	69.02	−0.078	3.189
IS01	41.97	0.022	−0.303	65.85	−0.045	1.029	75.31	0.007	−0.179
AB114	41.09	−0.002	0.0414	66.94	−0.074	1.805	77.76	0.021	−0.489
SV0	52.66	0.006	−0.088	81.05	−0.006	0.096	97.3	0.267	−4.0249
SV3	56.29	0.003	−0.025	69.46	0.022	−0.262	97.36	0.084	−1.082
Street	38.89	0.020	−0.476	82.42	−0.110	3.042	84.24	−0.070	1.879
HOC1	52.37	0.001	−0.020	94.96	−0.007	0.143	88.41	−0.006	0.144
Average	**44.50**	**−0.001**	**−0.023**	**74.27**	**−0.039**	**1.112**	**84.20**	**0.032**	**−0.080**

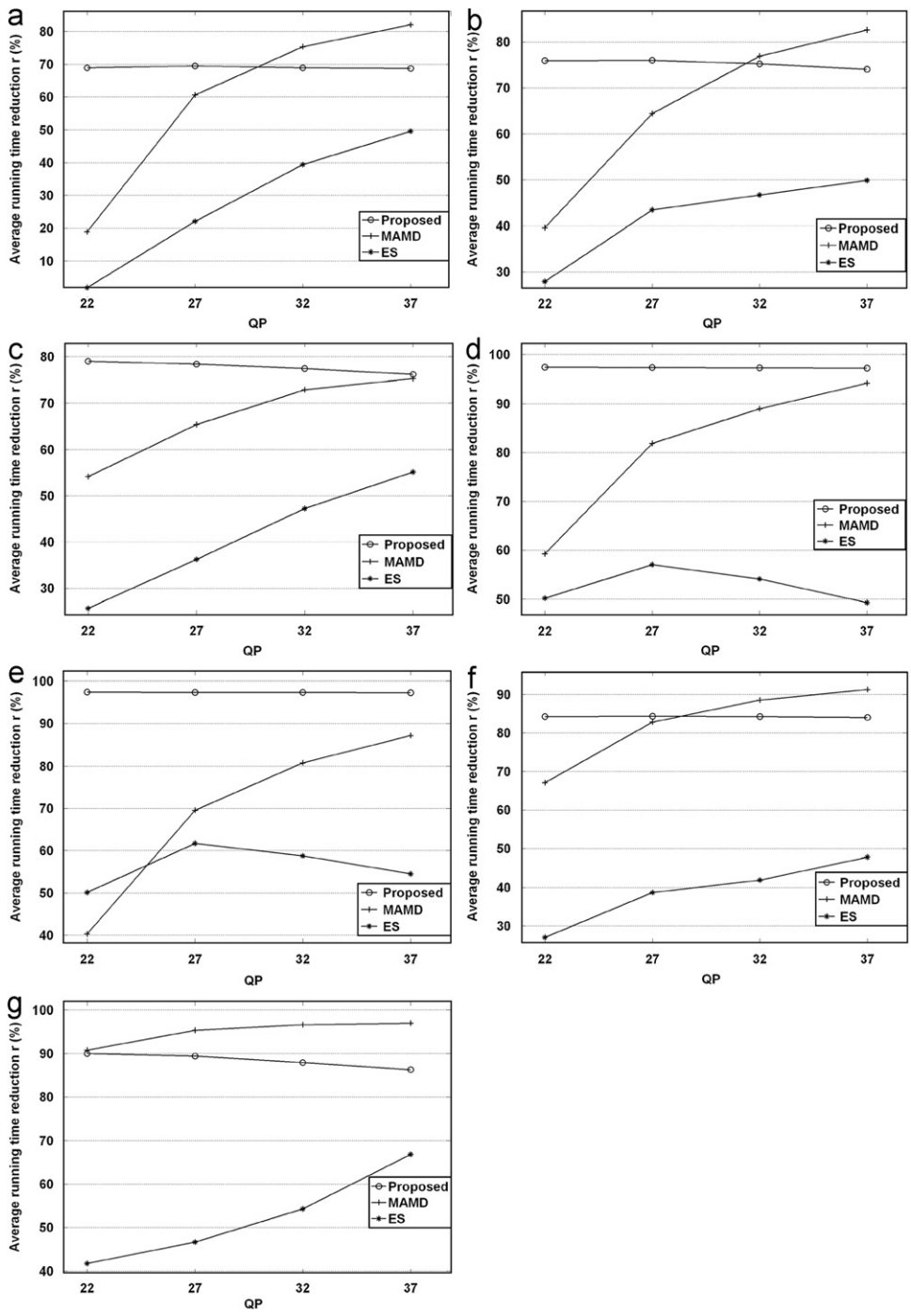

Fig. 7. Average running time reduction ratio comparison for ES, MAMD and the proposed at QP(I)=22, 27, 32, 37 for: (a) H-M; (b) IS01; (c) AB114; (d) SV0; (e) SV3; (f) Street and (g) HOC1.

quality and computational complexity of the proposed algorithm (Proposed) is compared with ES and MAMD in Table 4.

As shown in the table, the proposed algorithm outperforms both ES and MAMD in reducing the computational complexity and in preserving coding efficiency. It provides an average of 84% running time reduction for JM, which is 40% higher than ES and 10% higher than MAMD. Note that even though the computational complexity is greatly reduced by the proposed algorithm, the compression efficiency is not degraded for most of the sequences. The average gains in PSNR over that of JM is 0.032 dB. The largest PSNR improvement is more than 0.26 dB. Such

performance improvement is mainly due to the fact that the proposed algorithm produces more skipped MBs compared to JM, ES and MAMD.

The total running time reductions of ES, MAMD and our proposed DD at each QP are shown in Fig. 7. As shown in the figure, our proposed algorithm always provide a stationary reduction in total running time for all of the test sequences at all of the QP points in comparison with ES and MAMD. The total running time reduction introduced by the proposed algorithm is almost independent of the QP variation, or in another word, the running time reduction is almost not influenced by the coding bit rate. The total running time reductions of ES and MAMD always increase with the decrease in bit rate. However, to be efficient at the high coding bit rate is important for video surveillance application to provide compressed video with good quality for storage and future check.

Table 5
Average VIF difference comparison for ES [26], MAMD [28] and the Proposed.

Seq.	ES [26] vs. JM	MAMD [28] vs. JM	Proposed vs. JM
H-M	−0.0008	−0.0042	−0.0018
IS01	0.0000	−0.0019	−0.0009
AB114	0.0000	−0.0005	−0.0010
SV0	−0.0001	−0.0002	−0.0008
SV3	−0.0001	−0.0002	−0.0007
Street	−0.0003	−0.0101	−0.0013
HOC1	−0.0001	−0.0021	−0.0011
Average	**−0.0002**	**−0.0028**	**−0.0011**

Table 5 compares the average VIF difference. As listed in the table, the average loss in VIF introduced by the proposed algorithm against JM is only 0.0011, which represents virtually no loss in visual quality. It is smaller than that introduced by MAMD. Even though its VIF drop is 0.0009 larger than that introduce by ES, visually, no quality difference can be detected. Due to limitation of paper length, Fig. 8 provides the subjective quality comparison only for the decoded sequence of Hall Monitor, which has the largest loss in PSNR and VIF when the proposed Difference Detection algorithm is applied. As shown in the figure, no visual quality difference can be detected, which is consistent with the VIF results.

To evaluate the encoder adaptability of the proposed algorithm and to prove that the algorithm can further reduce the encoding complexity even if the encoder itself has integrated fast algorithm in mode decision module, the proposed algorithm has also been integrated with a fast version of JM15.1 encoder (FJM), which uses fast high

Fig. 8. Decoded frame 104 of Hall Monitor at QP(I)=27 by: (a) JM; (b) ES; (c) MAMD and (d) Proposed.

complexity RDO (setting *RDOptimization*=2) and in the meantime turns on two fast algorithms as early skip mode determination and selective Intra mode decision [26] in the mode decision module, to test its performance. Compared with JM, the FJM provides an average of 74% of computational complexity reduction with practically the same objective performance. Following the same testing methods described above (except setting *RDOptimization*=2, *EarlySkipEnable*=1 and *SelectiveIntraEnable*=1), the compression performances of DD working with FJM (DD+FJM) are compared with those of FJM and JM in Table 6. Also, the compression performance difference between FJM and JM is listed in the table to prove the acceleration property of FJM. As shown in the table, although FJM reduces the complexity of JM by approximately 74%, DD+FJM can further reduce the coding complexity of FJM by 82% and in the meantime preserves the objective and subjective quality. Compared with JM, DD+FJM provides an average of 95% reduction in coding complexity without objective quality loss. Fig. 9 provides the subjective quality comparison for the decoded sequence of Hall Monitor for JM, FJM, DD working with JM (DD+JM) and DD+FJM. As revealed in the figure, no visual

Table 6
Performance comparison for DD+FJM and FJM (Results of FJM vs. JM is to show the acceleration performance of FJM).

Seq.	FJM vs. JM			DD+FJM vs. FJM			DD+FJM vs. JM		
	ARTRR (%)	BD-PSNR (dB)	ΔVIF	*ARTRR* (%)	BD-PSNR (dB)	ΔVIF	*ARTRR* (%)	BD-PSNR (dB)	ΔVIF
H-M	62.64	−0.018	−0.0011	71.26	−0.092	−0.0018	89.21	−0.110	−0.0029
IS01	73.05	0.028	0.0000	77.68	−0.015	−0.0014	93.99	0.017	−0.0014
AB114	72.54	0.008	0.0001	77.13	−0.003	−0.0009	93.75	0.004	−0.0009
SV0	78.49	0.008	−0.0001	92.25	0.258	−0.0007	98.34	0.275	−0.0008
SV3	80.42	0.004	−0.0001	92.34	0.085	−0.0006	98.51	0.089	−0.0008
Street	72.02	0.025	−0.0004	81.72	−0.077	−0.0014	94.86	−0.051	−0.0018
HOC1	78.76	−0.010	−0.0002	84.14	−0.014	−0.0010	96.70	−0.023	−0.0014
Average	**73.99**	**0.006**	**−0.0003**	**82.36**	**0.020**	**−0.0011**	**95.05**	**0.029**	**−0.0014**

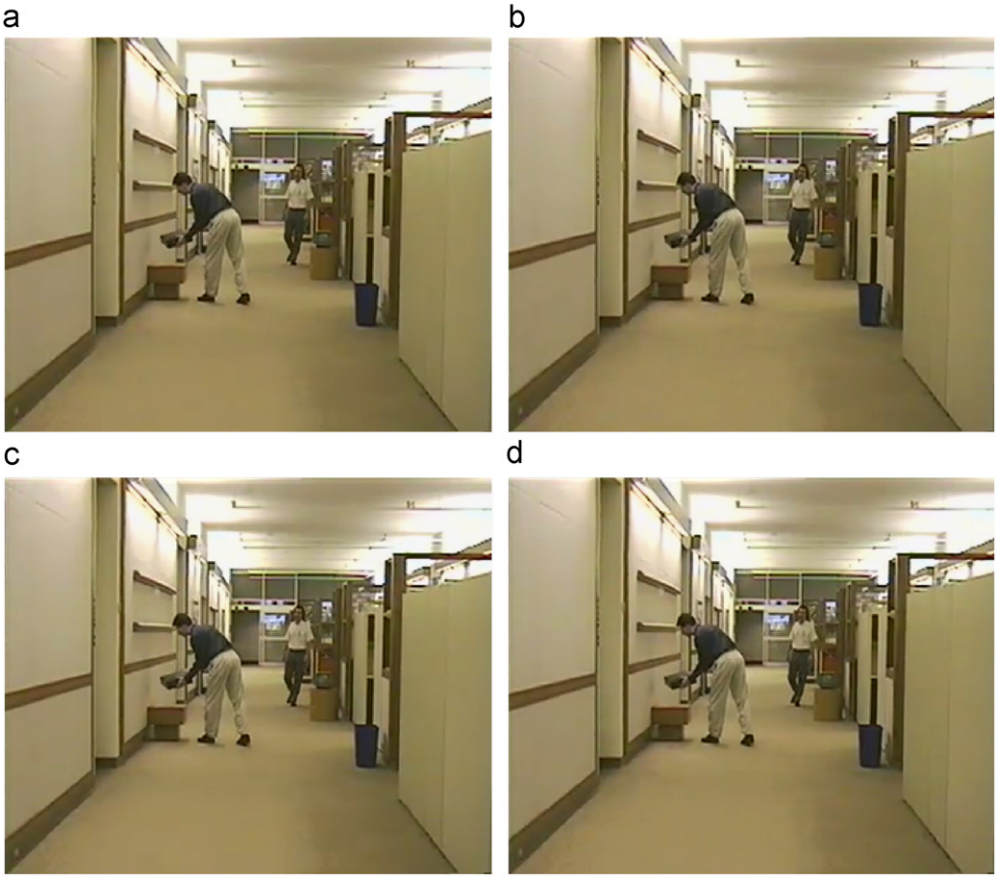

Fig. 9. Decoded frame 104 of Hall Monitor at QP(I)=27 by: (a) JM; (b) FJM; (c) DD+JM and (d) DD+FJM.

quality difference can be detected, which is consistent with the VIF results.

5. Conclusions

In this paper, an encoder adaptable difference detection algorithm is proposed to reduce the computational complexity and power consumption in surveillance video compression. Without any requirement in changing the encoder hardware, the proposed algorithm automatically distributes the input MBs to different modules of the video encoder according to the content difference in chrominance and motion feature. Experimental results revealed that the proposed algorithm is effective at both the encoders with and without fast algorithms. The overall computational complexity of the whole encoding system is significantly reduced, and no subjective or objective video quality loss is observed. The proposed algorithm can potentially have part of the encoder circuit operated at a lower clock speed to further reduce the power consumption, which will be investigated further.

Acknowledgement

This work is sponsored by a grant from the Core Research for Evolution Science and Technology (CREST), Japan Science and Technology Agency (JST), JAPAN.

Thanks for the video data coming from the ViSOR repository, found at URL: http://www.openvisor.org.

References

[1] See Privacy International, Leading Surveillance Societies in the EU and the World 2007, 〈http://www.privacyinternational.org/article.shtml?cmd[347]=x-347-559597〉.
[2] M. Gill, A. Spriggs, Assessing the impact of CCTV, Home Office Research, Development and Statistics Directorate, Feb. 2005.
[3] Brandon C. Welsh, David P. Farrington, Evidence-based crime prevention: the effectiveness of CCTV, Crime Prev. Community Saf.—Int. J. 6 (2) (2004) 21–33.
[4] Britain is 'surveillance society', BBC News. November 2, 2006. 〈http://news.bbc.co.uk/1/hi/uk/6108496.stm〉.
[5] Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC), March 2003.
[6] Inform. Technol.: Digital Compression and Coding of Continuous-Tone Still Images: Requirements and Guidelines, JPEG Std., ISO/IEC 10918-1 and ITU-T T.81, 1991.
[7] Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to about 1.5 Mbits/s - Part 2: Video, ISO/IEC JTC1 ISO/IEC 11172-2 (MPEG-1), March 1993.
[8] Generic Coding of Moving Pictures and Associated Audio Inform.–Part 2: Video, ITU-T and ISO/IEC JTC1 ITU-T Recommendation H.262 and ISO/IEC 13818-2 (MPEG-2), November 1994.
[9] Coding of Audiovisual Objects-Part 2: Visual, ISO/IEC ISO/IEC 14496-2 (MPEG-4), 1999.
[10] Video Codec for Audiovisual Services at $p \times 64$ kbits/s, ITU-T Recommendation H.261 Version 2, Mar. 1993.
[11] Video Coding for Low Bit Rate Commun., ITU-T Recommendation H.263 Version 2, January 1998.
[12] T. Wiegand, G. Sullivan, G. Bjøntegaard, A. Luthra, Overview of the H.264/AVC video coding standard, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (2003) 560–576.
[13] T. Wiegand, G. Sullivan, The H.264/AVC video coding standard [Standards in a Nutshell], IEEE Signal Process. Mag. 24 (2) (Mar. 2007) 148–153.
[14] Y. Yu, D. Doermann, Model of object-based coding for surveillance video, in: Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Process.(ICASSP), vol. 2, 2005, pp. 693–696 (March 2005).
[15] T. Nishi, H. Fujiyoshi, Object-based video coding using pixel state analysis, in: Proceedings of 17th International Conference on Pattern Recognition (ICPR), vol. 3, 2004, pp. 306–309 (August 2004).
[16] D. Venkatraman, A. Makur, A compressive sensing approach to object-based surveillance video coding, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2009, 2009, pp. 3513–3516 (April 2009).
[17] S. Yaman, G. AlRegib, A low-complexity video encoder with decoder motion estimator, in: Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP) 2004, vol. 3, 2004, pp. iii - 157–60 (May 2004).
[18] L. Liu, Z. Li, E.J. Delp, Efficient and low-complexity surveillance video compression using backward-channel aware Wyner-Ziv video coding, IEEE Trans. Circuits Syst. Video Technol. 19 (4) (2009) 453–465.
[19] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, Comput. Vision Pattern Recognition 2 (1999) 246–252.
[20] A. Elgammal, D. Harwood, L. Davis, Non-parametric model for background subtraction, in: Proceedings of Sixth European Conference on ComputerVision, (June 2000).
[21] D. Gutchess, M. Trajkovics, E. Cohen-Solal, D. Lyons, A. Jain, A background model initialization algorithm for video surveillance, in: Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, Canada, 2001, vol. 1, pp. 733–740 (July 2001).
[22] I. Haritaoglu, D. Harwood, L. Davis, A fast background scene modeling and maintenance for outdoor surveillance, in: Proceedings of the 15th International Conference on Pattern Recognition, Barcelona, Spain, 2000, vol. 4, pp.179–183 (September 2000).
[23] Horng-Horng Lin, Tyng-Luh Liu, Jen-Hui Chuang, Learning a scene background model via classification, IEEE Trans. Signal Process. 57 (5) (May 2009) 1641–1654.
[24] Janaka Liyanage, GMM based background subtraction on GPU, 〈http://www.cs.ucf.edu/~janaka/gpu/GMM_report.doc〉, (April 2009).
[25] JM reference software 15.1, downloaded at 〈http://iphome.hhi.de/suehring/〉.
[26] I. Choi, J. Lee, B. Jeon, Fast coding mode selection with rate-distortion optimization for MPEG-4 Part-10 AVC/ H.264, IEEE Trans. Circuits Syst. Video Technol. 16 (12) (2006) 1557–1561.
[27] F. Pan, X. Lin, S. Rahadja, K. P. Lim, Z. G. Li, D. Wu, S. Wu, Fast Intra mode decision algorithm for H.264/AVC video coding, in: Proceedings of the International Conference on Image Processing (ICIP) 2004, vol. 2, 2004, pp. 781–784 (October 2004).
[28] H. Zeng, C. Cai, Ma Kai-Kuang, Fast mode decision for H.264/AVC based on macroblock motion activity, IEEE Trans. Circuits Syst. Video Technol. 19 (4) (Apr. 2009) 491–499.
[29] B. Jung, B. Jeon, Adaptive slice-level parallelism for H.264/AVC encoding using pre macroblock mode selection, J. Vis. Commun. Image R. 19 (8) (2008) 558–572.
[30] D Wu, F. Pan, K.P. Lim, S Wu, Z.G. Li, X. Lin, S. Rahardja, C.C. Ko, Fast intermode decision in H.264/AVC video coding, IEEE Trans. Circuits Syst. Video Technol. 15 (7) (2005) 953–958.
[31] Z. Chen, P. Zhou, Y. He, Fast integer pel and fractional pel motion estimation for JVT, JVT of ISO/IEC MPEG & ITRU_T VCEG, 6th meeting, Awaji, Island, JP, 5–13 December, 2002.
[32] A. M. Tourapis, Enhanced predictive zonal search for single and multiple frame motion estimation, in: Proceedings of the Visual Communications and Image Processing (VCIP) 2002, 2002, pp.1069–1079 (January 2002).
[33] T.K. Tan, G. Sullivan, T. Wedi, Recommended simulation common conditions for coding efficiency experiments revision 4, ITU-T SC16/Q6, in: Proceedings of the 36th VCEG Meeting, San Diego, USA, 8th–10th October 2008, Doc. VCEG-AJ10r1.
[34] 〈http://www.multitel.be/~va/candela/intersection.html〉.
[35] 〈http://www.openvisor.org/video_categories.asp〉.
[36] V. Lappalainen, A. Hallapuro, T.D. Hämäläinen, Complexity of optimized H.26 L video decoder implementation, IEEE Trans. Circuits Syst. Video Technol. 13 (7) (July 2003) 717–725.
[37] G. Bjøntegaard, Improvements of the BD-PSNR model, ITU-T SC16/Q6, in: Proceedings of the 35th VCEG Meeting, Berlin, Germany, 16th–18th July, 2008, Doc. VCEG-AI11.
[38] H.R Sheikh, A.C Bovik, Image information and visual quality, IEEE Trans. Image Process 15 (2) (Feb. 2006) 430–444.
[39] Pixel domain version of VIF, 〈http://live.ece.utexas.edu/research/Quality/index.htm〉.