EURASIP Journal on Advances in Signal Processing
a SpringerOpen Journal

RESEARCH Open Access

Video coding with dynamic background

Manoranjan Paul[1*], Weisi Lin[2*], Chiew Tong Lau[2] and Bu-Sung Lee[2]

Abstract

Motion estimation (ME) and motion compensation (MC) using variable block size, sub-pixel search, and multiple reference frames (MRFs) are the major reasons for improved coding performance of the H.264 video coding standard over other contemporary coding standards. The concept of MRFs is suitable for repetitive motion, uncovered background, non-integer pixel displacement, lighting change, etc. The requirement of index codes of the reference frames, computational time in ME & MC, and memory buffer for coded frames limits the number of reference frames used in practical applications. In typical video sequences, the previous frame is used as a reference frame with 68–92% of cases. In this article, we propose a new video coding method using a reference frame [i.e., the most common frame in scene (McFIS)] generated by dynamic background modeling. McFIS is more effective in terms of rate-distortion and computational time performance compared to the MRFs techniques. It has also inherent capability of scene change detection (SCD) for adaptive group of picture (GOP) size determination. As a result, we integrate SCD (for GOP determination) with reference frame generation. The experimental results show that the proposed coding scheme outperforms the H.264 video coding with five reference frames and the two relevant state-of-the-art algorithms by 0.5–2.0 dB with less computational time.

Keywords: Motion estimation, Video coding, H.264, Multiple reference frame, Scene change detection, Adaptive GOP, Uncovered background, Motion compensation

1. Introduction

The H.264/AVC video coding standard improves rate-distortion performance significantly compared to its predecessors and competitors by introducing a number of innovative ideas in Intra- and Inter-frame coding [1-3]. Major performance improvement is taken place by means of *motion estimation* (ME) and *motion compensation* (MC) using variable block size, sub-pixel search, and *multiple reference frames* (MRFs) [3-8]. It has been demonstrated that MRFs facilitate better predictions than using just one reference frame, for video with repetitive motion, uncovered background, non-integer pixel displacement, lighting change, etc. Moreover, better error-resilient coding can be obtained using MRFs [9] where Zheng and Chau showed that referencing some macro-blocks of the current frame from the furthest reference frame improves error resilience. The requirement of index codes (to identify the particular reference frame used), computational time in ME & MC (which increases almost linearly with the number of reference frames), and memory buffer size (to store decoded frames in both encoder and decoder) limits the number of reference frames used in practical applications. The optimal number of MRFs depends on the content of the video sequences. Typically, the number of reference frames varies from one to five. If the cycle of repetitive motion, exposing uncovered background, non-integer pixel displacement, or lighting change exceeds the number of reference frames used in MRFs coding system, there will be not any improvement and therefore, the related computation (mainly that of ME) and bits for index codes are wasted. Moreover, the existing MRFs-based system experiences disaster in decoded picture quality if any frame is lost during transmission.

To tackle with the major problem of MRFs, a number of techniques [5-8,10] have been developed for reducing the computation associated with. Huang et al. [5] searched either the previous or every reference frame based upon the result of the intra prediction and ME from the previous frame. This approach can reduce 76–96% of computational complexity by avoiding unnecessary search for reference frames. Moreover, this approach is orthogonal

* Correspondence: MPaul@csu.edu.au; WSLIN@ntu.edu.sg
[1]School of Computing and Mathematics, Charles Sturt University, Charles Sturt, Australia
[2]School of Computer Engineering, Nanyang Technological University, Singapore, Singapore

to conventional fast block matching algorithms, and they can easily be combined to achieve further efficient implementation. Shen et al. [6] proposed an adaptive and fast MRF selection algorithm based on the hypothesis that homogeneous areas of video sequences probably belong to the same video object, move together as well, and thus have the same optimal reference frame. Simulation results show that this algorithm deducts 56–74% of computation time in ME. Kuo et al. [7] proposed a fast MRF selection algorithm based on the initial search results using 8 × 8-pixel block. Hachicha et al. [8] used *Markov Random Fields* algorithm relying on robust moving pixel segmentation, and saved 35% of coding time by reducing the number of reference frames to three instead of five without image quality loss. Saponara et al. [10] added a low complexity context-aware controller to a basic ME engine to avoid unnecessary computations and memory accesses while keeping unaltered coding efficiency for a wide range of applications.

Most of the fast MRFs selection algorithms including the above-mentioned techniques used one reference frame (in the best case) when their assumptions on the *correlation* of the MRFs selection procedure are satisfied or five reference frames (in the worse case) when their assumptions completely fail. But it is obvious that in terms of rate-distortion performance, these techniques cannot outperform the H.264 with five reference frames which is considered as optimal [1]. Moreover, they also suffer image quality degradation if any frame is missing during transmission.

In H.264, a *group of picture* (GOP) comprises one *Intra* (I-) frame with subsequent *predicted* (P-) and/or *bi-directional* (B-) frames. Typical size of a GOP is 30 in the American NTSC television standard and 25 in the European PAL standard. With regular interval (i.e., at the beginning of a GOP) an I-frame is inserted for error propagation prevention, backward/forward play, indexing, etc. We have observed that I-frame requires two to three times more bits compared to the inter (i.e., P or B)-frame for the same image quality. Figure 1 shows frame-level bits and PSNR performance using the H.264 for I-frame and P-frame with *Paris* video sequence. The figure demonstrates that I-frame requires around three times (3.03 and 2.88 times when quantization parameter QP = 30 and QP = 20, respectively) more bits compared to that of P-frame. In general, if a sequence does not contain any scene changes or extremely high motion activity compared to the previous frames, insertion of I-frames reduces the coding performance. Therefore, we need to insert optimal I-frames based on the *adaptive GOP* (AGOP) determination and scene change detection (SCD) algorithms.

A number of algorithms [4,11-14] are proposed in the literature for AGOP and SCD. Dimou et al. [11] used dynamic threshold based on the *mean* and *standard deviation* of the previous frames for SCD. Their reported

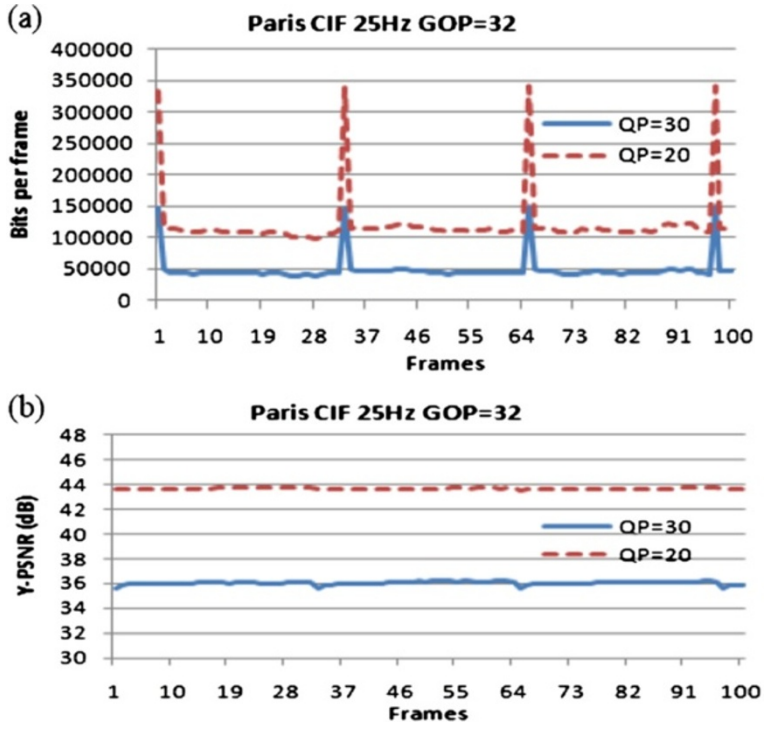

Figure 1 Frame level bits (a) and PSNRs (b) by I-frame and P-frame using the H.264 standard using two QPs for Paris video sequence.

accuracy is 94% on average. Alfonso et al. [12] used ME & MC to find the SCD. To avoid repetitive scene change, they imposed lower limit of scene change as four frames. The success rate of this method is 96% with 7.5–15% more compression and 0.2-dB quality loss.

Matsuoka et al. [13] proposed a combined SCD and AGOP method based on fixed thresholds generated from the accumulated difference of luminance pixel components. They used the *number of the intensive pixels* (NIP) to investigate the frame characteristics. A pixel of a frame is considered as an intensive one if the luminance pixel difference between the adjacent frames is bigger than 100. If NIP exceeds a pre-defined threshold between two frames, then insert an I-frame at that position assuming the occurrence of SCD; otherwise they restricted GOP size to either 8 or 32 based on the NIP and another threshold. Song et al. [14] proposed another SCD method based on [13] focusing on the hierarchical B-picture structure.

Ding and Yang [4] also combined AGOP and SCD for better coding efficiency based on different video content variations (VCVs), which can be extracted from temporal deviation between two consecutive frames. The VCVs are measured using the *sum of absolute motion vectors* (SAMV) and the *sum of absolute transformed differences* (SATD) with 4 × 4-pixel blocks. For AGOP, this method used SAMV with the previously processed frames in a GOP to determine one of the pre-defined GOP sizes among {16, 32, 64, 128, and 256}. They determined the SCD if the ratio of SATD of tth frame and $(t-1)$th frame is greater than 1.7, and inserted an I-frame if SCD occurs. This method ensured 98% accuracy of SCD with 0.63-dB image quality improvement.

The above-mentioned AGOP and/or SCD techniques require comparison between the current frame and a number of previous frames for better rate-distortion performance. We believe that a joint AGOP and SCD technique can be developed using only one *appropriate* frame containing *enough* scene information for computationally efficient and better rate-distortion performance. In this article, we generate a *most common frame in scene* (McFIS), for SCD and AGOP, and finally as an effective reference frame for better rate-distortion performance in coding.

Moreover, due to the limited number of reference frames (the maximum is five in practical implementations), uncovered background may not be encoded efficiently using the existing techniques. Some algorithms [15-18] determined and exploited uncovered background using pre- and/or post-processing and computationally expensive video segmentation for coding. Uncovered background can also efficiently be encoded using *sprite* coding through object segmentation. Most of the video coding applications could not tolerate inaccurate video/object segmentations and expensive computational complexity incurred by segmentation algorithms. Ding et al. [18] used a background-frame for video coding. The background frame is made up of blocks which keep *unchanged* (based on the zero motion vector) in a certain number of continuous frames. Due to the dependency on block-based motion vectors and lack of adaptability in multi-modal backgrounds for dynamic environment, this background frame could not perform well.

Recently, *dynamic background modeling* (DBM) [19-21] using *Gaussian mixture model* (GMM) has been introduced for robust and real-time object detection from the so-called *dynamic environment* where *ground-truth* background (GTB) is impossible. Moreover, static background model does not remain valid due to illumination variation over time, intentional or unintentional camera displacement, shadow/reflection of foreground objects, and intrinsic background motions (e.g., waving tree leaves, etc.) [21]. Object can be detected more accurately by subtracting background frame (generated from the background model) from the current frame. In this article, we have incorporated DBM into the video coding to improve the SCD for AGOP, coding performance, and error concealment. First we generate an McFIS from the pre-decoded frames using DBM, and then use it as second reference frame (first reference frame is the immediate previous frame). The same McFIS generation is used at the encoder and decoder so that we do not need to send background model to the decoder.

Using McFIS as a reference frame we have the following advantages compared to the existing methods based on MRFs and SCD for AGOP:

- Only one McFIS is used instead of a number of reference frames so the overheads of index codes are reduced.
- An McFIS enables the possibility of capturing a whole cycle of repetitive motion, exposing uncovered background, non-integer pixel displacement, or lighting change.
- The new frame referencing scheme is designed with clearer purpose: the immediate previous frame is meant for moving areas, and the McFIS is meant for background regions.
- Since an McFIS is generated from the already decoded frames, intrinsically it has better error recovery capacity for error-prone channel transmission as the McFIS model has already contained pixel intensity history of the frames.
- A simple mechanism for AGOP and SCD determination is possible using McFIS as it is the most common frame in that scene. Thus, any mechanism for SCD and AGOP determination by

comparing difference between McFIS and the current frame is more effective. In fact, the SCD (therefore AGOP) is integrated with reference frame generation.
- Less computation in ME & MC is required using McFIS compared to the multiple frames (true for the comparison with more than two reference frames).
- A better error-resilient coding can be obtained due to the referencing some macroblocks of the current frame from the furthest reference frame (i.e., McFIS) as described in [9].
- Due to the direct referencing from the long-term reference frame (i.e., McFIS) less variable (i.e., more consistent) bit rate and PSNR [22] can be obtained so that GOP-boundary artifacts would be reduced [23].
- The main contributions of the proposed technique are
- A new background modeling technique has been proposed using decoded frames for coding gain.
- A new skip mode is defined using newly developed dynamic background frame.
- A new SCD technique is derived using McFIS.
- Comprehensive analysis and simulation results [on computational time, SCD, amount of referencing based on dynamic background (i.e., McFIS), and rate-distortion performance] are provided to understand the effectiveness of McFIS in video coding.

The rest of the article is organized as follows. Section "GMM-based DBM" describes the existing DBM and their limitations for processing using distorted video frames. Section "Proposed video coding algorithm" proposed McFIS-based method. The overall experimental set up and results for the proposed scheme are presented in Section "Overall experimental results", while Section "Conclusions" concludes the article.

2. GMM-based DBM

GMM-based DBM [19-21] has been proved effective for object detection from the dynamic environment. The DBM is performed at pixel level, i.e., each pixel of a scene is modeled independently by a mixture of K (normally at most three models are used in the existing techniques [19-21]) Gaussian distributions. Each Gaussian model represents the intensity distribution of one of the different environment components, e.g., moving objects, waiving trees, static background, etc., observed with the pixel in frames. If we assume that kth Gaussian at time t representing a pixel intensity is $\eta_{k,t}$ with mean $\mu_{k,t}$, variance $\sigma^2_{k,t}$, and weight $w_{k,t}$ such that $\sum_{\forall k} w_{k,t} = 1$. The learning parameter α is used to balance the contribution between the current and past values of parameters such as weight, variance, mean, etc. Obviously, $1/\alpha$ defines the time constant which determines the speed at which the distribution's parameters change. Contribution of the current change of pixel in the model is minimal and *tailing effect* of the previous object/background can be visible if we use very low α (e.g., 0.001). For the real-time processing and integrating SCD in the proposed algorithm, we need faster learning rate. The system starts with an empty set of models. The fixed initial parameters are suggested in [21] as follows: maximum number of model for a pixel is 3, learning rate is 0.1, weight is 0.001, and variance is 900. The Gaussians are always ordered based on the w/σ in descending order assuming that the top Gaussian will provide most stable background [21].

After initialization, for every new observation X_t at the current time t, it is first matched against the existing models in order to find one (say the kth model) such that $|X_t - \mu_{k,t}| \le 2.5\sigma_{k,t}$. If such a model exists, its associated parameters are updated as follows [19] where $\alpha < 1$ is the learning rate

$$\mu_{k,t} = (1-\alpha)\mu_{k,t-1} + \alpha X_t; \tag{1}$$

$$\sigma^2_{k,t} = (1-\alpha)\sigma^2_{k,t-1} + \alpha\left(X_t - \mu_{k,t}\right)^T\left(X_t - \mu_{k,t}\right); \tag{2}$$

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1} + \alpha, \tag{3a}$$

and the weights of the remaining Gaussians are updated as

$$\omega_{k,t} = (1-\alpha)\omega_{k,t-1}. \tag{3b}$$

After this approximation, the weights are renormalized. If such a model does not exist, a new Gaussian is introduced with $\mu = X_t$, arbitrarily high σ, and arbitrarily low ω by evicting η_K if it exists.

From the above-mentioned models, background and foreground are determined using different techniques. Stauffer and Grimson [19] used a user-defined threshold based on the background and foreground ratios. A predefined threshold does not perform well in object/background detection because the ratio of background and foreground varies from video-to-video. Lee [20] used two parameters (instead of a threshold used in [19]) of a *sigmoid* function by modeling the posterior probability of a Gaussian to be background. This method also depends on the proportion by which a pixel is going to be observed as background. Moreover, the generated background has delay response due to using the weighted mean of all the background models [21]. To avoid *mean effect* (mean is considered as an *artificially*

generated value and sometimes far from the original value) and delay response, Haque et al. [21] used a parameter called *recentVal*, *m* to store recent pixel intensity value when a pixel satisfies a model in the Gaussian mixture. They used *classical* background subtraction method which identifies an object if the value of the current intensity differs from the *recentVal*, *m* of the *best* background model by a well-studied threshold. This method reduces not only delay response, but also learning rates, which are sometimes desirable criteria for real-time object detection.

The existing GMM-based DBM (using pixel intensity from the original videos, i.e., lossless video) with its associated background generation (using recent value, *m* of the pixel intensity) performs well for robust object detection scheme. However, in the video coding applications, the above-mentioned strategy for DBM does not perform well as we need to model using *distorted* (i.e., decoded using quantization) video frames for better compression. Thus, the above-mentioned approach for background generation using recent value (i.e., distorted recent value) also loses its meaning.

3. Proposed video coding algorithm

The primary purpose of the existing background modeling is to detect object, however, in the video coding applications, the primary purpose is to compress video data without degrading image quality. Thus, straightforward application of the existing background modeling is not effective for compression. In the proposed method, we have proposed a new technique for background modeling as well as incorporated an SCD scheme based on the newly generated background frame for coding performance gain.

An McFIS is generated using real-time DBM based on the GMM [19-21]. Obviously, *traditional* DBM (tDBM) would be different from our *proposed* DBM (pDBM) as the tDBM primarily focuses on object detection, whereas the pDBM focuses on rate-distortion optimization when an McFIS is used as an extra reference frame for encoding uncovered background, repetitive motion, non-integer pixel displacement, light change, etc. Moreover, the tDBM has used original video frames (i.e., lossless) to construct background frame, whereas the pDBM will use decoded frames (i.e., lossy), which are quantized at different levels based on the available bit rate. The McFIS is also used for SCD toward AGOP for efficient video coding. The subsequence subsections will describe McFIS generation, AGOP determination through SCD, and the proposed coding scheme.

3.1. Generation of McFIS

Figure 2 shows the original and decoded pixel intensities for Frame 1 to Frame 100 as O(row, column) and $D1$ (row, column), respectively, of three different diagonally positioned pixels (where row and column are counted from the top-left). For example, in original or undistorted frames (i.e., 1 to 100 frames), O(36, 44) indicates a pixel position at row 36 and column 44 and the pixel intensities are 104 at Frame 1, 102 at Frame 23, and 115 at Frame 90. In decoded frames (i.e., after coding and decoding) $D1$(36, 44) indicates the same pixel position (as O(36, 44)) and the decoded pixel intensities are 114 at Frame 1, 101 at Frame 23, and 115 at Frame 90. These data are collected from the first 100 frames of *Hall Monitor* video sequence when encoded using the proposed coding technique (described later) with the state of the art tDBM [21] method at QP = 40. The solid lines and dotted lines of the figure represent original and decoded pixel intensities, respectively. From the figure, one can easily observe that decoded pixel intensities differ from the corresponding original pixel intensities. It is due to the quantization and block-based ME & MC used in the coding system. This pixel intensity discrepancy increases with the quantization, and especially is a severe problem at low bit rates. Note that according to Equations (1) to (3), all $(t - 1)$ previous decoded frames are (somehow) used to generate a tth McFIS unless there is a scene change. Obviously, the contribution of the older frames diminishes with the time (depends on the learning parameter α). If there is scene change, then all parameters are reset and modeling starts again.

We have also observed that there is pixel intensity similarity among neighboring pixels. This relationship is also observed by the other researchers, and thus, pre-/post-filtering techniques were introduced by exploiting neighboring pixels to reduce pixel intensity discrepancy in decoded frames due to the quantization and/or block-based ME & MC [24,25]. We have also exploited neighboring pixel intensities to the pDBM. Let D_t and M_{t-1} be the tth decoded frame and $(t - 1)$th McFIS,

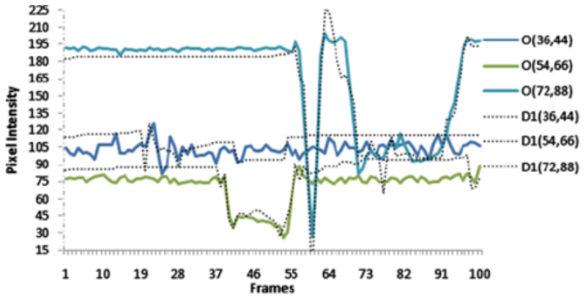

Figure 2 Pixel intensities in the original frames, *O*(row, column) and decoded frames, *D*1(row, column), for the first 100 frames of *Hall Monitor* video sequence where decoded pixel intensities (from Frame 1 to Frame 100) *D*1 is reconstructed by the H.264 video coding standard using McFIS generated by the tDBM at quantization parameter QP = 40.

respectively, to generate tth McFIS, M_t. For a given pixel position (x,y) in D_t, we modify $D_t(x, y)$ as $D_t^{'}(x, y)$

$$D_t^{'}(x,y) = \begin{cases} \tau D_t(x,y) + (1-\tau)\bar{D}_t(x,y) & \text{if } |D_t(x,y) - \bar{D}_t(x,y)| < T_p \\ D_t(x,y), & \text{otherwise} \end{cases} \tag{4}$$

where τ and T_p are the weighting factor and threshold, respectively. $\bar{D}_t(x,y)$ is defined as follows

$$\bar{D}_t(x,y) = 1/4 \sum_{i=0}^{1} \sum_{j=0}^{1} D_t(x+i, y+j). \tag{5}$$

Note that all existing DBM algorithms [19-21] use original pixel intensities for their dynamic modeling, whereas, the proposed technique uses decoded pixel intensities (but modified using Equation (5)). In our experiment, we have used τ = 0.5 and T_p = 3. This minimizes the *trailing effect* (i.e., some portion of objects remains in background) of moving objects in the McFIS generation using a small threshold T_p. Note that we have only used right and bottom neighboring pixels for the possible modification of McFIS in the proposed scheme. We do not considered left and upper neighboring pixels to restrict the number of pixels to make the McFIS smooth. If we consider more neighboring pixels, it may make the McFIS more blur and eventually the reconstructed image. However, selection of neighboring pixels is still an open question to be investigated in the future for efficient coding performance.

We have observed that generation of background image, i.e., McFIS using *recentVal* sometimes does not work properly. It is due to the pixel intensity fluctuation causes by the course quantization. To minimize this variation, we have used same (i.e., τ) weighting factor between the *mean* and *recentVal*, m to get McFIS (i.e., background) pixel intensity $m^{'}$, i.e., $m^{'} = \tau\mu + (1 - \tau)m$ where μ and m are the mean pixel intensity and recent pixel intensity (i.e., *recentVal*), respectively, of the background model selected for McFIS generation as defined in GMM-based DBM.

Figure 3 shows original (i.e., O), decoded (i.e., $D1$ denotes tDBM and $D2$ denotes pDBM) pixel intensities of three diagonal pixel positions for *Hall Monitor* video sequence at QP = 40. The figure clearly shows that proposed pDBM provides closer pixel intensities to the original compared to that of tDBM. Closer pixel intensity approximation enables better quality of background generation.

We call the resultant frame by pDBM as the McFIS because pDBM preserves the most stable pixel intensity for a pixel over the time; this is the frame which has the most *similarity* with the other frames within the corresponding scene. We also believe that a *properly* generated McFIS can replace the I-frame (the first frame of a GOP or a scene) for better coding performance under a given bit budget.

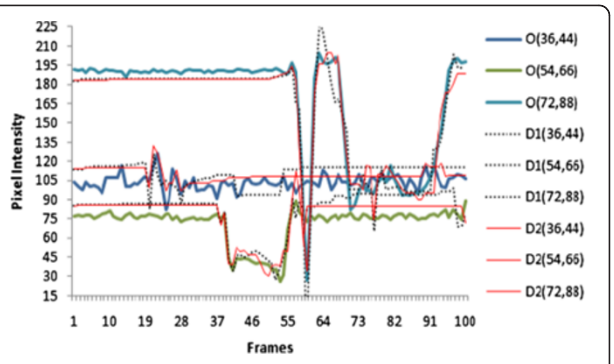

Figure 3 Pixel intensities in original frames *O*(row, column), decoded frames *D*1(row, column), and decoded frames *D*2(row, column) using first 100 frames of *Hall Monitor* video sequence; the decoded frame *D*1 and *D*2 are reconstructed by the H.264 video coding standard using McFIS generated by the tDBM and pDBM, respectively, at quantization parameter, QP = 40.

Figure 4 shows subjective comparison of different McFISes generated by the tDBM, pDBM, and motion vector-based [18] techniques with encoding *Hall Monitor* video sequence at QP = 40. Figure 4a shows the original 45th frame, and Figure 4b,c shows McFISes using tDBM and pDBM. In the figure, at the man's position there is debris in Figure 4b whereas in Figure 4c it is less obvious. For clear visualization we have also included two images (see Figure 4e,f) using difference between GTB and the McFISes generated by the tDBM and the pDBM, respectively. We have multiplied the absolute different by 10 for clear view. The area enclosed by a circle in the Figure 4e is the most distinctive area between two McFISes. We also found that the absolute different matrix between GTB and the McFIS generated by tDBM has maximum 135 and average 5.3 values, whereas the counterpart generated by the McFIS by pDBM has 125 and 4.8. All the above-mentioned evidences indicate that the proposed pDBM generates more accurate background compared to the tDBM, and this leads to efficient encoding of uncovered background regions. We have also created background frame shown in Figure 4d using motion vector-based technique [18]. This background does not capture uncovered background (i.e., no background at the man's position (black regions) due to the non-zero motion vectors for those regions). Thus, this background frame is not suitable for efficient coding compared to the background generated using DBM.

We have also generated the McFIS using the first 25 original (undistorted) frames of a scene, and then encoded it as an Intra-frame with finer quantization. The rate-distortion performance is improved significantly when we have used the McFIS as a second reference frame for encoding the rest of the inter-frames of

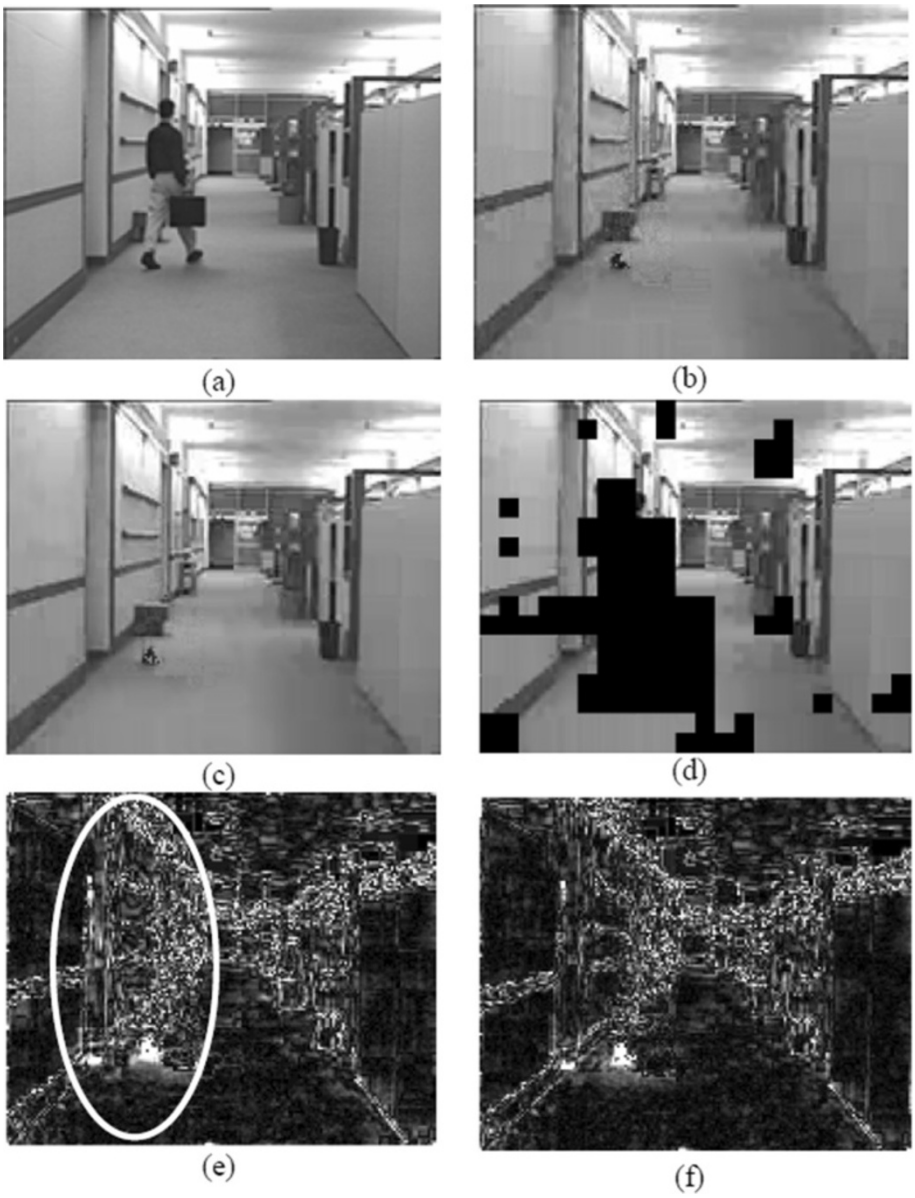

Figure 4 Comparison of different McFISes using tDBM, pDBM, and [18] techniques. (a) Original 45th frame of *Hall Monitor* video sequence, **(b)** McFIS using tDBM, **(c)** McFIS using pDBM, **(d)** background-frame using [18], **(e)** difference between (b) and GTB, and **(f)** difference between (c) and GTB (both images multiplied by 10 for clear visualization) when encoded at QP = 40.

the scene [26]. However, there is little or no improvement of the approach [26] for the video sequences with frequent scene changes and camera motions because in these cases a large number of McFISes need to be encoded (thus increases bits requirements) to keep the McFIS relevant for referencing to encode the inter-frames. Note that the McFIS is not a displayable frame, thus, whenever a scene change has been detected, an extra high-quality frame (i.e., McFIS) needs to be encoded for the scheme in [26] and results poor rate-distortion performance. For example, for a video with 100 frames and 10 scene changes we need to encode 111 frames (extra 11 McFIS frames with high quality) for the scheme in [26], whereas we need to encode 100 frames for the proposed scheme. Currently, 25 frames are used to generate an McFIS after SCD in the scheme in [26]; however, it is difficult to find the optimal number of frames requirement for McFIS generation. Selection of suitable quantization levels for McFIS coding at different bit rates is also a difficult problem. Moreover, this approach [26] is not suitable if the application could not tolerate any decoding delay or play back delay due to the time requirement for generating McFIS with few frames before actual coding. In the proposed approach, we have

overcome this shortcoming by dynamically updated the McFIS with the recent encoded frame before encoding the inter-frame.

3.2. SCD and AGOP

As we mentioned in "Introduction" section that proper insertion of I-frame makes rate-distortion performance better. Most of the existing methods used some metrics computed with already processed frames and the current frames. The McFIS is the most similar frame comprising stable portion of the scene (mainly background) compared to the individual frame in a scene. Thus, the SCD is determined by a *simple* metric computed using the McFIS and the current frame. This would be effective compared to the existing algorithms as the McFIS is *equivalent* to a group of already processed frames. According to the free dictionary (http://www.thefreedictionary.com/scene), a scene is defined as the place where an action or event occurs. Thus, a scene change means the change of background or stable portion (not the foreground or the objects) of a video sequence. Through background modeling McFIS captures entire background of a scene (without moving objects) (see Figure 4b,c), thus, for SCD, McFIS would be appropriate frame to compare with for scene change determination against the current frame. As the McFIS has the *history* of the scene we do not need a *rigorous* process (like Ding's AGOP and SCD algorithm) to determine scene change. Unlike the other existing algorithm, we do not need any explicit algorithm for AGOP, and it can be achieved as an integrated part of the McFIS process, as to be described next.

For SCD using McFIS, we find the *sum of absolute difference* (SAD) between the McFIS and the current frame. If the SAD for the current frame is 1.7 greater than that of the previous frame, then we consider SCD occur and insert an I-frame, otherwise we continue inter-coding. The threshold 1.7 is initially set by Ding and Yang [4], we also find effective in our implementation. Due to dynamic nature of scene complexity and variations in videos, SAD variations using McFIS against a current frame are least compared to that of using the immediate previous frame as McFIS does not contain moving objects. Thus, McFIS would be better choice for SCD compared to the immediate previous frame. We do not need any ME (unlike Ding's algorithm) for the current frame before taking intra/inter-frame decision. Obviously, we need computation to generate the McFIS, but this can be paid off by avoiding ME time in AGOP determination as for Ding's algorithm. We do not need to compute NIP for the pre-decoded frames (unlike Matsuoka's algorithm).

To see the effectiveness of the proposed technique, we have created two mixed video sequences: *Mixed A* and *Mixed B* of 700 frames comprising 11 different standard video sequences (like in the existing algorithms [4,13]). *Mixed A* and *Mixed B* video sequences comprise the first 50/100 frames of the specified QCIF and CIF videos, respectively, as shown in Table 1. From the table, it is clear that for both mixed sequences, total 10 scene changes are occurred at 101, 151, 201, 251, 351, 401, 501, 551, 601, and 651 frames. We have compared our results with two most recent and effective AGOP and SCD algorithms [4,13] for efficient video coding.

Figure 5 shows the SCD results by the proposed, Ding's and Matsuoka's methods using three QPs = {40, 28, and 20} for *Mixed A* video sequence. We have plotted SAD ratio (see Figure 5a), SATD ratio (see Figure 5b), and NIP (see Figure 5c) for the proposed, Ding's and Matsuoka's algorithms, respectively. As we mentioned earlier, for the proposed method an SCD occurs if the SAD ratio is above 1.7 (i.e., the SAD for the current frame is 70% greater than that of the previous). For the Ding's algorithm, an SCD occurs if the SATD ratio is more than 1.7 [4]. For the Matsuoka's algorithm, an SCD occurs if the NIP is more than 1,000 [13] for QCIF sequences. Thus, it is clear from the figure that for each of the SCD positions (i.e., 101, 151, 201, 251, 351, 401, 501, 551, 601, and 651 frames), the proposed and Ding's methods successfully detect all scene changes. On the other hand, Matsuoka's method successfully detects all scene changes except at the 501 frame for QP = 40 and 28 due to the similarity in background between *salesman* and *grandma* video sequences.

Similar curves are also drawn in Figure 6 using *Mixed B* video sequence. The figure shows that the proposed and Ding's algorithms successfully identify all SCD locations but Matsuoka's algorithm detects 30, 30, and 29 extra locations for three cases, QPs = {40, 28, and 20}, respectively, being false SCD. The majority of the extra SCDs occur from 551 to 600 frames due to the high-motion *Football* sequence. Note that for CIF sequences, Matsuoka's algorithm identifies SCD if NIP > 4,000.

Table 1 Mixed video sequences for SCD and AGOP

Mixed A (QCIF)	Mixed B (CIF)	Frames	Frames in mixed sequence
Akiyo	Silent	100	1–100
Miss America	Waterfall	50	101–150
Claire	Coastguard	50	151–200
Car phone	Paris	50	201–250
Hall Monitor	Hall Monitor	100	251–350
News	Container	50	351–400
Salesman	Bridge far	100	401–500
Grandma	Highway	50	501–550
Mother	Football	50	551–600
Suzie	Bridge close	50	601–650
Foreman	Tennis	50	651–700

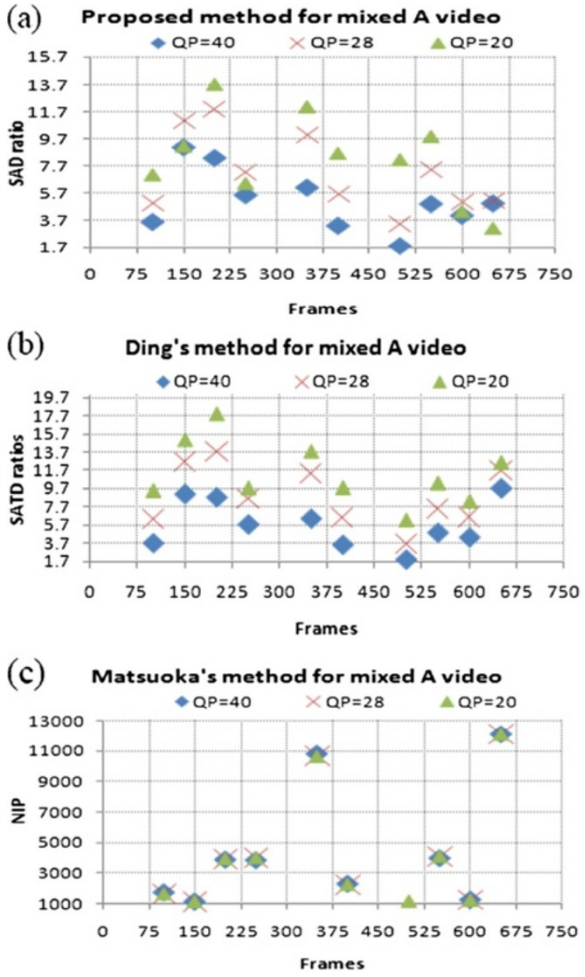

Figure 5 SCD using three algorithms namely the proposed method, Ding's, and Matsuoka's methods for Mixed video A comprising 11 QCIF video sequences.

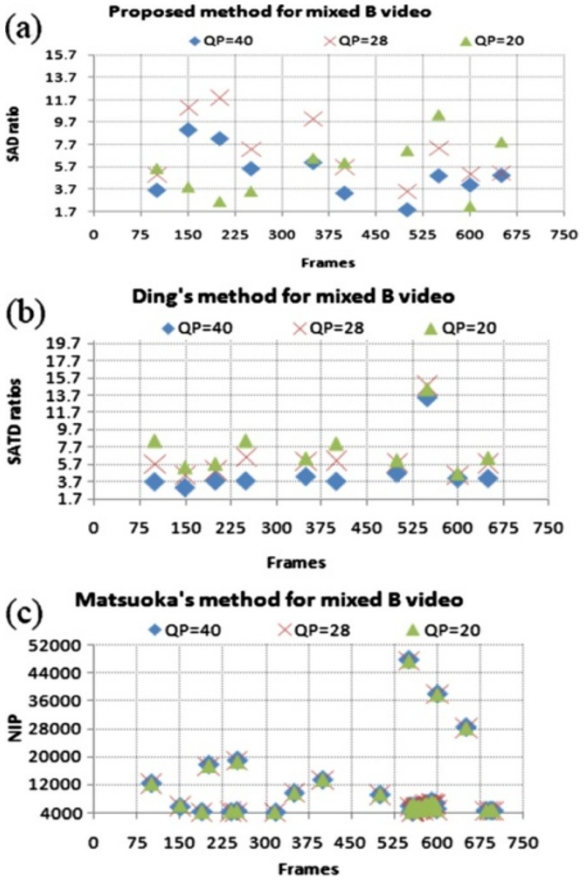

Figure 6 SCD using three algorithms namely the proposed method, Ding's, and Matsuoka's methods for Mixed video B comprising 11 CIF video sequences.

Although the proposed scheme performs similarly with Ding's algorithm in SCD for the two sequences, it outperforms when applied to actual coding due to the AGOP differences. Table 2 shows total I-frame insertion based on the SCD and AGOP using three methods for *Mixed A* and *Mixed B* video sequences at three QPs = {40, 28, and 20}. While the proposed method only inserts ten I-frames at the SCD locations (based on the SAD ratios) for all cases, Ding's method sometimes inserts extra I-frames (e.g., 11 I-frames for Mixed B video at QP = 40) besides SCD locations for their AGOP technique. This extra I-frame insertion does not help to improve rate-distortion coding efficiency (later we will show with rate-distortion performance) as there is no SCD. Matsuoka's algorithm inserts extra I-frames not only for the AGOP (e.g., 21 I-frames for *Mixed A* or *Mixed B* sequences at QP = 40), but also their false SCD (e.g., 40 I-frames for Mixed B video sequence at QP = 40). This algorithm sometimes even misses SCD (e.g., at QP = 40 and 28 for *Mixed A* video sequence, it detects only 9 cases whereas there are 10 cases of SCDs).

With the results from Figures 5, 6, and Table 2, we see that the proposed approach using McFIS is more

Table 2 Number of I-frames for mixed video A and B of 700 frames

Methods	QPs	Number of I-frames			
		Mixed video A		Mixed video B	
		SCD	AGOP	SCD	AGOP
Proposed algorithm	40	10	0	10	0
	28	10	0	10	0
	20	10	0	10	0
Ding's algorithm	40	10	0	10	11
	28	10	0	10	4
	20	10	0	10	4
Matsuoka's algorithm	40	9	21	40	21
	28	9	21	40	21
	20	10	21	39	21

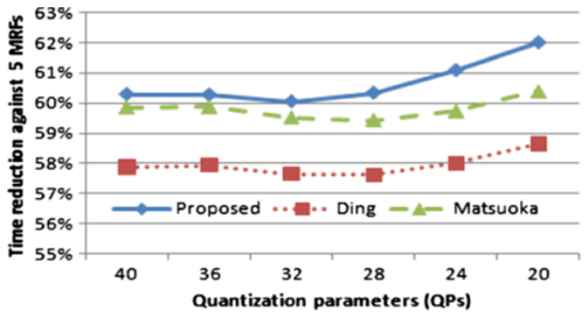

Figure 7 Average computational time reduction of the proposed, Ding's, and Matsuoka's algorithms against the H.264 5 MRFs using different standard video sequences.

effective in SCD and AGOP, compared to the two recent algorithms.

3.3. The proposed coding system

As we have mentioned earlier, the proposed pDBM is incorporated into the encoder and decoder in the same way to generate McFIS from the decoded frames so that we do not need to encode an McFIS. This also provides more error resilience in the frame/packet-loss situation as the McFIS (i.e., instead of furthest reference frame) can be used to restore the lost information at the decoder [9].

First, an McFIS is used as the second reference frame in addition to the immediate previous frame. The H.264 encoder and decoder are employed in the proposed scheme with the only difference being that an McFIS is used as the second reference frame instead of five previous frames as reference frames. That is, the proposed scheme has two reference frames: the immediate previous frame and McFIS, based on the rate-distortion Lagrangian optimization, the final reference frame is selected from these two for each block.

As the proposed McFIS would be a better choice of a reference frame especially for smooth areas, true background and uncovered background areas compared to the other four previous frames, we have introduced a new skip *macroblock* (MB) condition by comparing current MB and its co-located MB in the McFIS. If the difference between these two MBs is small then we consider the current MB as skipped MB. The rationality of the new skipped MB is that small changes between the current MB and the co-located MB in the McFIS indicate that the current MB is a part of a stable background, thus no need to encode (due to no motion and small residual error) it rather than just simply copy it from the McFIS.

Pattern-based video coding techniques [27,28] used a definition for static MB (SMB) which is equivalent to the skip MB, with a fixed threshold for various bitrates. We have used same kind of definition but in *dynamic* fashion, i.e., a function of QP to cope with different bitrates. We have observed that for coarse quantization we can use a large threshold and for fine quantization we need to use a small threshold to maintain better rate-distortion performance in the proposed algorithm.

Let C_k (x, y) and R_k (x, y) denote the kth MB of the current frame (with frame size of $W \times H$) and the corresponding McFIS, respectively. The moving region M_k (x, y) of the kth MB in the current frame is obtained as [27]:

$$M_k(x,y) = \begin{cases} 1, & \text{if } |(C_k(x,y) \bullet B) - (R_k(x,y) \bullet B)| > 2; \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

where $0 \le x, y \le 15$, $0 \le k < W/16 \times H/16$, and B is a 3×3 unit (i.e., containing only '1') matrix for the *morphological closing* operation (denoted by "$\bullet$" in (6)),

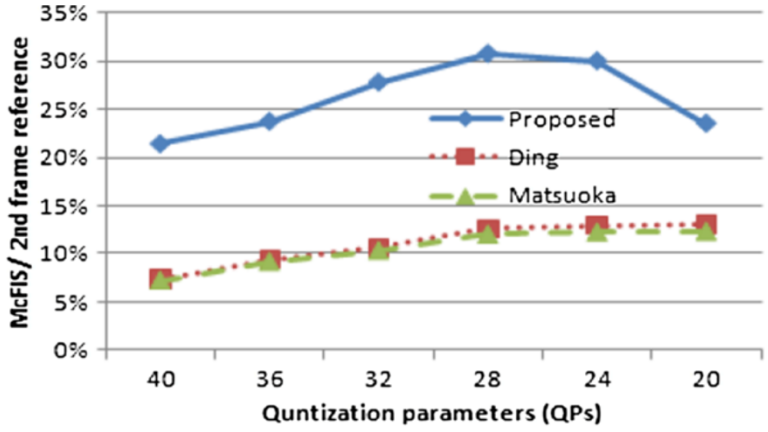

Figure 8 Percentages of references of the McFIS (for proposed method) and second reference frame (for Ding's and Matsuoka's algorithms) using Mixed A and Mixed B video sequences.

which is applied to reduce noise. Wong et al. [27] chose SMB if $\sum M_k(x, y) < 8$.

In the proposed scheme, an MB is skipped if $\sum M_k(x, y) < QP/2$. By this new definition, the proposed coding technique classified more MBs as SMBs. This does not jeopardise image quality as the McFIS is a better reference frame. Note that if any MB is classified as an SMB, we do not process any other modes to speed up the encoding.

4. Overall experimental results

Overall experimental results are performed using 23 standard video sequences, comprising of 4CIF (720 × 576), CIF (352 × 288), and QCIF (176 × 144) digital video format. All sequences are encoded at 25 frames per second. Full-search fractional ME with ±15 as the search length and IPPP... format are used. We have compared the proposed method with three relevant existing algorithms, namely Ding's algorithm [4], Matsuoka's algorithm [13], and the H.264 fixed GOP (32 as the GOP size for fixed GOP) using five reference frames, in terms of rate-distortion and computational complexity. We have found that Ding's algorithm is the best existing method in rate-distortion, SCD, and AGOP performance, while Matsuoka's algorithm is the latest and simplest technique for SCD and AGOP. For the complete comparison, we have also selected the H.264 standard video coding using fixed GOP and five reference frames. For Ding's and Matsuoka's algorithms we have used two reference frames (the immediate previous and the second immediate previous frames). As mentioned earlier, the proposed algorithm uses the immediate previous frame and the McFIS as the two reference frames. We use H.264 with five reference frames and fixed GOP to prove that the proposed scheme outperforms the state-of-the-art method. We use Ding's and Matsuoka's algorithms to prove that the proposed scheme is better in terms of rate-distortion, computational time, and SCD.

4.1. Computational complexity

The ME, irrespective of a scene's complexity, typically comprises more than 60% of the processing overhead required to inter-encode a frame with a software codec using the DCT [29], when full search is used. Obviously, ME computational time is also varied with the number of reference frames, precision of ME, etc. A comprehensive performance and complexity analysis on a tool-by-tool basis is provided in [30]. The proposed technique takes some extra operations to generate McFIS and interpolate McFIS for encoding each frame. Ding's algorithm needs extra ME and SATD calculations for SCD and AGOP. Matsuoka's algorithm has only extra computation for NIP and sum of total NIP for 32 frame calculations. But it is very difficult to analyze theoretical computational complexity of each algorithm because of too many parameters and coding conditions. Thus, we have compared their computational performance based on the empirical data. Note that the proposed technique needs extra time in the encoder and decoder compared to the other relevant techniques due to the background modeling. The background modeling time is fixed and does not depend on the search length. The experimental result shows that extra 2% of encoding time is needed when we encode 100 frames with 15 search length.

Figure 7 shows experimental results of computation reduction of the proposed, Ding's, and Matsuoka's algorithms against the H.264 with five references frames,

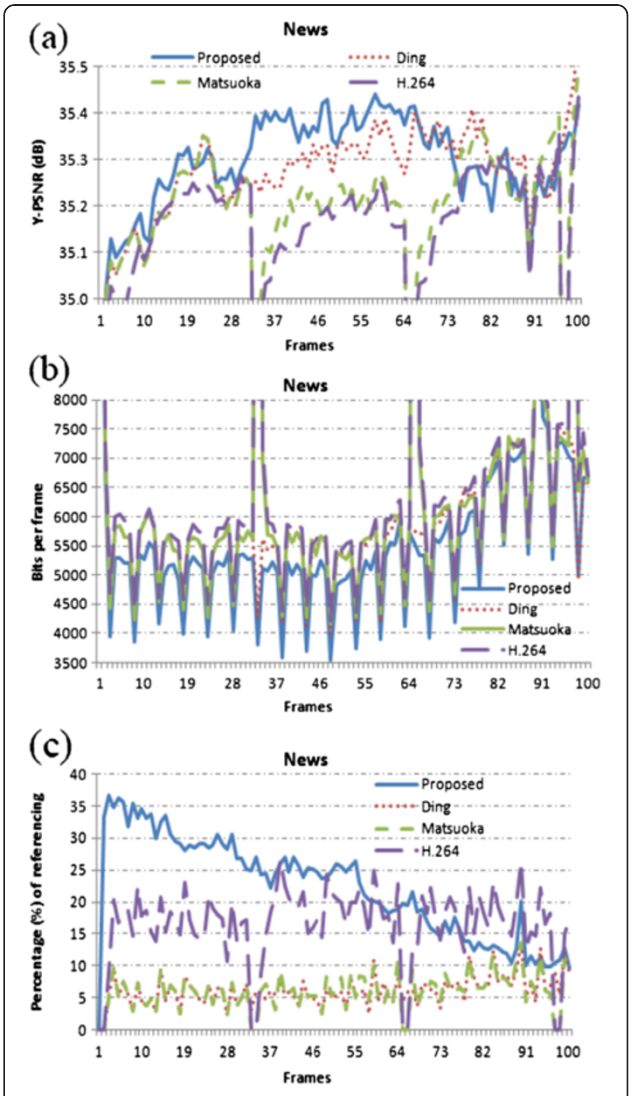

Figure 9 Frame level information. (a) Luminance PSNR, **(b)** Bits per frame, and **(c)** Percentage of references using McFIS/second previous frame for 100 frames in *News* video sequence with the proposed, Ding's, Matsuoka's, and the H.264 (fixed GOP and five reference frames) algorithms.

using a number of video sequences (*Mixed A*, *Mixed B*, *Silent*, *Hall Monitor*, *Salesman*, *News*, *Paris*, and *Susie*) over different QPs, i.e., 40, 36, 32, 28, 24, and 20. This figure confirms that the proposed, Ding's, and Matsuoka's algorithms reduce 61, 58, and 60% on average, respectively. Thus, we can conclude that the proposed scheme is comparable with (in fact, slightly better than) the two state-of-art methods in complexity while it can save 61% of computational time on average compared to the H.264 with five reference frames. Actually, we have observed that the proposed technique generates more skip modes (due to the new definition of skip mode) compared to the existing methods. For example, for *Paris* video sequence, the proposed technique generates 2/3 times more skip modes compared to Ding's algorithm at high bit rates. As we mentioned at the end of Proposed video coding algorithm that if any MB is classified as a skip mode, we do not process any other modes to speed up the encoding. That's why we get better coding time compared to the other schemes. The skip mode mainly comes from the McFIS references (see Figure 8, McFIS references are higher).

4.2. Performance comparisons in other perspectives

Figure 8 shows the average percentages of using the McFIS as the reference frame for the proposed method and the second previous frame for the other two methods, with Mixed A and Mixed B video sequences. The figure demonstrates that the proposed method has 26% of the cases using McFIS on average whereas the other two have only about 11% of cases using the immediate

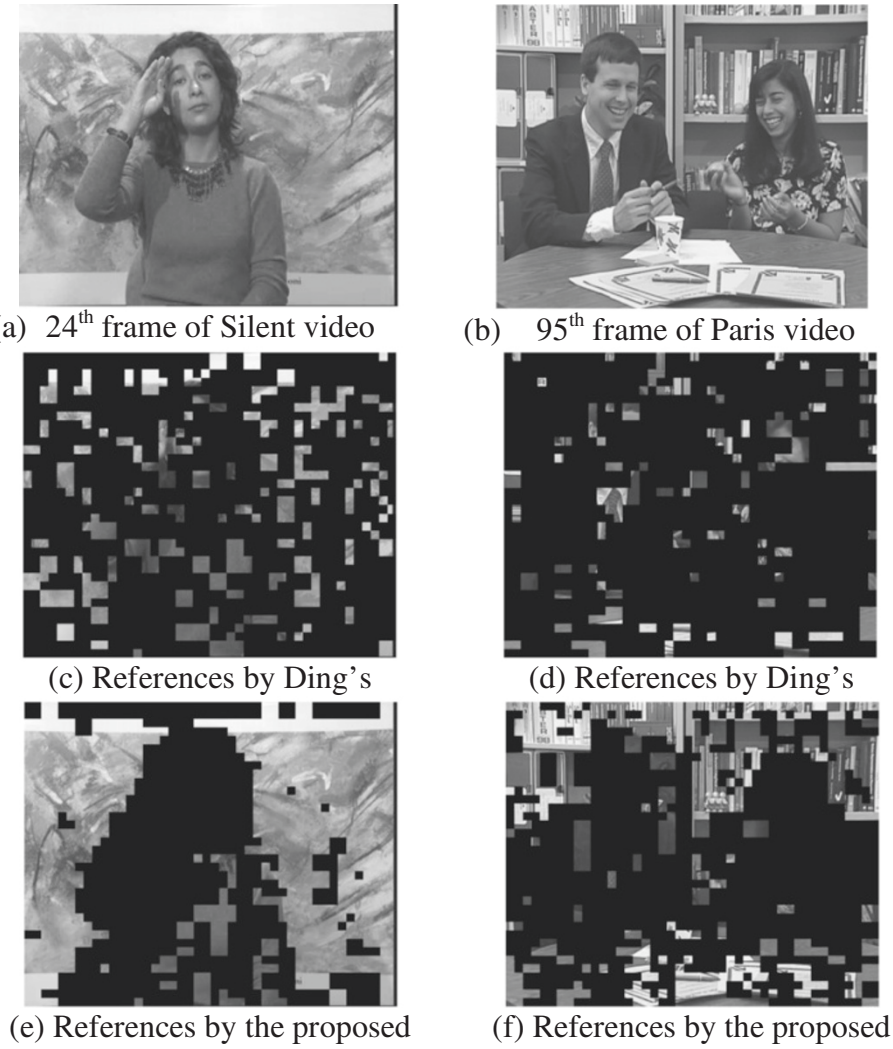

Figure 10 Rate-distortion performance by the proposed, Ding's, Matsuoka's, and the H.264 standard video coding (with fixed GOP and five reference frames) algorithms using Mixed A (comprising 700 frames of 11 various QCIF standard video sequences), Mixed B (comprising 700 frames of 11 various CIF standard video sequences), and other four QCIF, two CIF, and one 4CIF video sequences with at least 300 frames from each.

second previous frame on average. The significantly larger referencing frequency indicates rate-distortion improvement using McFIS as a reference frame against using the second previous frame. Moreover, a better error-resilient coding can be obtained due to the large number of referencing from the McFIS as described by Zheng and Chau [9]. They showed that referencing some macroblocks of the current frame from the furthest reference frame improves error resilience. Instead of using the furthest reference frame, if we use McFIS as the reference frame, we can achieve better error-resilient coding using Zheng and Chau's approach.

For detailed understanding we have provided frame level data for *News* video sequence. Figure 9 shows detailed data using luminance PSNR (Y-PSNR), bits per frame, and percentage of references using McFIS/second previous frame by the proposed, Ding's, Matsuoka's, and the H.264 (with fixed GOP and five reference frames) algorithms, respectively. Figure 9a demonstrates that the proposed algorithm is the best to produce higher PSNR compared to the other three algorithms. It is due to the use of McFIS. The standard H.264 (with fixed GOP and five reference frames) produces the worst PSNR. Between the other two algorithms, Ding's algorithm

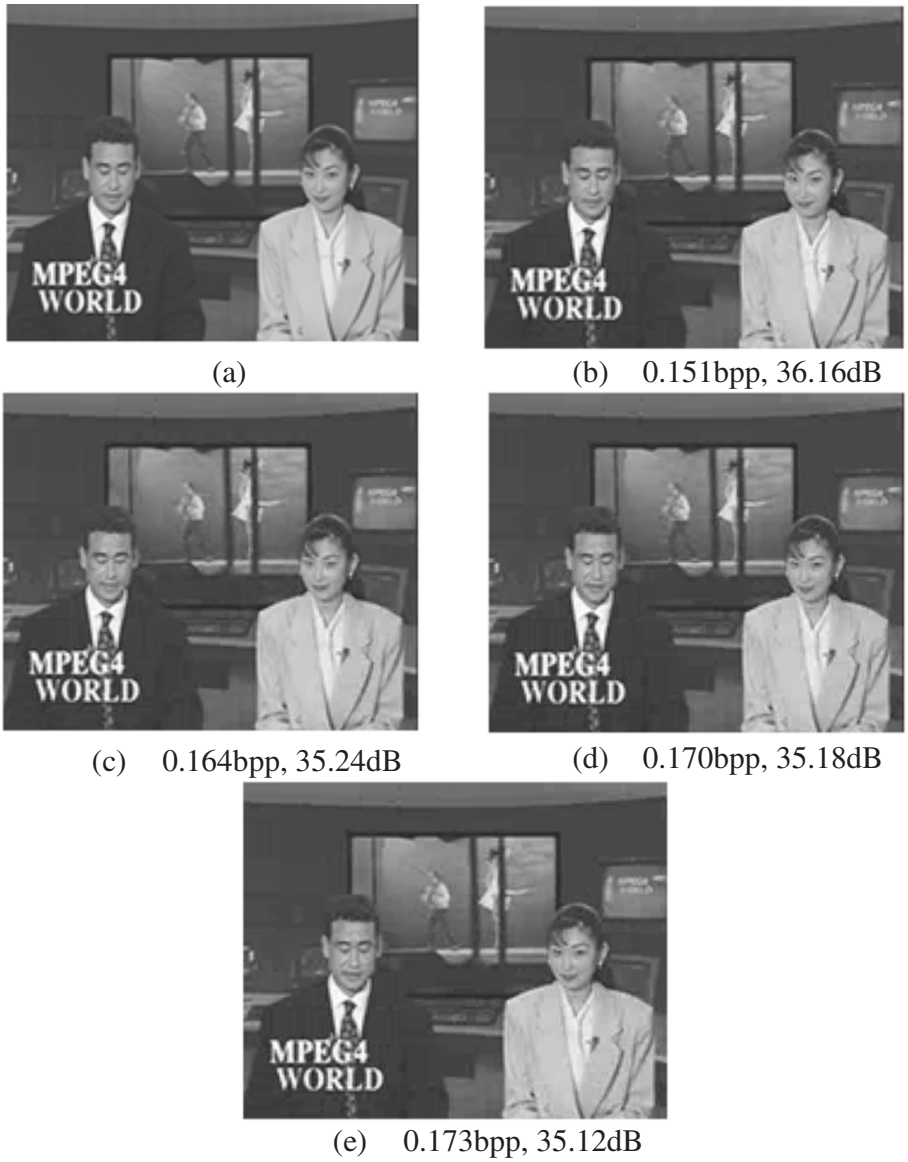

Figure 11 Frame level reference maps by the proposed Ding' method for *Silent* and *Paris* video sequences. (a, b) Two decoded 24th frame of *Silent* and 95th frame of *Paris* videos, **(c, d)** reference maps by the Ding's algorithm, and **(e, f)** reference maps by the proposed algorithm where black and other regions are referenced from the immediate previous frame and the McFIS (for the proposed)/second frame (for the Ding's), respectively.

performs better compared to Matsuoka's algorithm. This is due to the relatively less I-frame insertion in the Ding's algorithm compared to the Matsuoka's algorithm. Both algorithms insert I-frame at the beginning of a GOP (GOP size being 8 or 32 in Matsuoka's algorithm and 16, 32, 64, 128, or 256 in Ding's algorithm, all based on the AGOP) and at the SCD locations. For the proposed method, we have only inserted an I-frame if SCD occurs.

Figure 9b,c shows frame-level bits and percentage of references (the second previous frame for Matsuoka's and Ding's algorithms, McFIS for the proposed algorithm, and second to fifth previous frames for the H.264). Fewer bits per frame are needed for the proposed method, while the highest bits per frame are used for the standard H.264. Since the reference frame is always selected from the candidate frame pool to achieve the best performance for any encoder, the higher referencing rate for one particular frame means better for coding and more effective for the associated index codes. With the higher referencing rate of McFIS in Figure 9c, the proposed method outperforms the other three methods in terms of compression (lower bits per frame) and image quality (higher PSNR) (see Figure 9a,b). This means if we keep bit rates constant, PSNR would be even higher for the proposed method, as will be demonstrated in Figure 10. The percentage of references using McFIS diminishes with the time (see Figure 9c). In *News* sequence, there is a dancing behind the readers, as we know that McFIS only captures the background, and thus, the percentage of the McFIS referencing for the object area (due to dancing) diminishes with the time of background modeling. The other reason is that when we select a mode (whether from the first reference or the second reference), we prefer McFIS if the cost functions for both are the same.

Due to the direct referencing from the long-term reference frame (i.e., McFIS) less variable (i.e., more consistent) bit rate and PSNR [22] can be obtained by the proposed approach. Figure 9a,b shows the evidence of better bits and PSNR consistency by the proposed method compared to the other relevant methods. This is a desirable property for better perceptual quality [31]. The proposed adaptive GOP determination based on the SCD provides longer GOP compared to that of relevant algorithms. This also provides pleasant perceptual video quality [31] by reducing GOP-boundary artifacts [23].

Figure 11 shows reference mapping using *Silent* and *Paris* video sequences by the proposed scheme and Ding's algorithm. A scattered referencing takes place using Ding's algorithm for the immediate previous and second previous frames. For the proposed method, moving object areas (black regions in Figure 11e,f) are referenced using the immediate previous frame whereas background regions are referenced using McFIS (normal area in Figure 11e,f). A large number of areas (normal regions in Figure 11e,f) are referenced by the McFIS, and this indicates the effectiveness of the McFIS for improving coding performance (as discussed above for Figure 9).

Figure 12 shows decoded frames for subjective viewing tests by the proposed, Ding's, Matsuoka's, and H.264 (with fixed GOP and five reference frames) algorithms at QP = 32. The 38th frame of *News* sequence is shown as an example. They are encoded using 0.151, 0.164, 0.170, and 0.173 bits per pixel (bpp) and resulting in 36.16, 35.24, 35.18, and 35.12 dB in Y-PSNR, respectively. From the viewing tests with ten people, the decoded video by the proposed scheme is with the best subjective quality. It is due to the fact that the proposed method spends relatively more bits at the moving areas and fewer bits for the smooth/background areas compared to the other methods. Thus, the quality of the moving areas (i.e., area comprising objects) is better in the proposed method.

The proposed technique with SCD encodes the first frame and the frames at the point of SCD as the I-frames. Thus, for a video sequence with no/small camera motion, the proposed scheme may have fewer numbers of I-frames; on the other hand, for a video with high camera motion, it may have higher number of I-frames compared to H.264. Figure 13 shows rate-distortion performance of the proposed scheme with SCD (i.e., flexible GOP) and fixed GOP size against the scheme in [26] and the H.264 with two reference frames using *Tennis* video sequence. We have selected *Tennis* sequence as it has camera motions and scene change. The figure confirms the superiority of the proposed scheme with SCD and fixed GOP over the algorithm in [26] and the H.264 with two reference frames. The figure also demonstrates

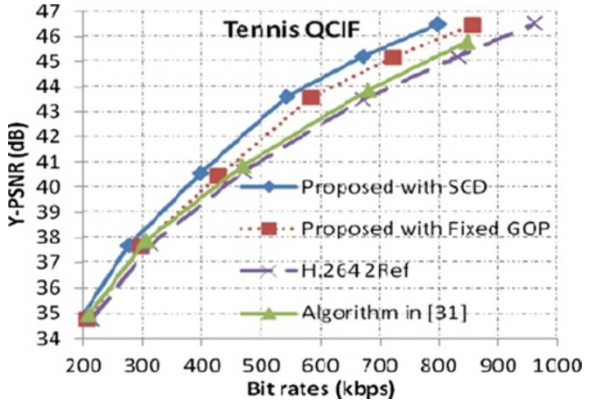

Figure 12 Decoded 38th frame of *News* video sequence: (a) original frame, (b) the proposed, (c) Ding's, (d) Matsuoka's, and (e) the H.264 (with fixed GOP and five reference frames) algorithms at QP = 32.

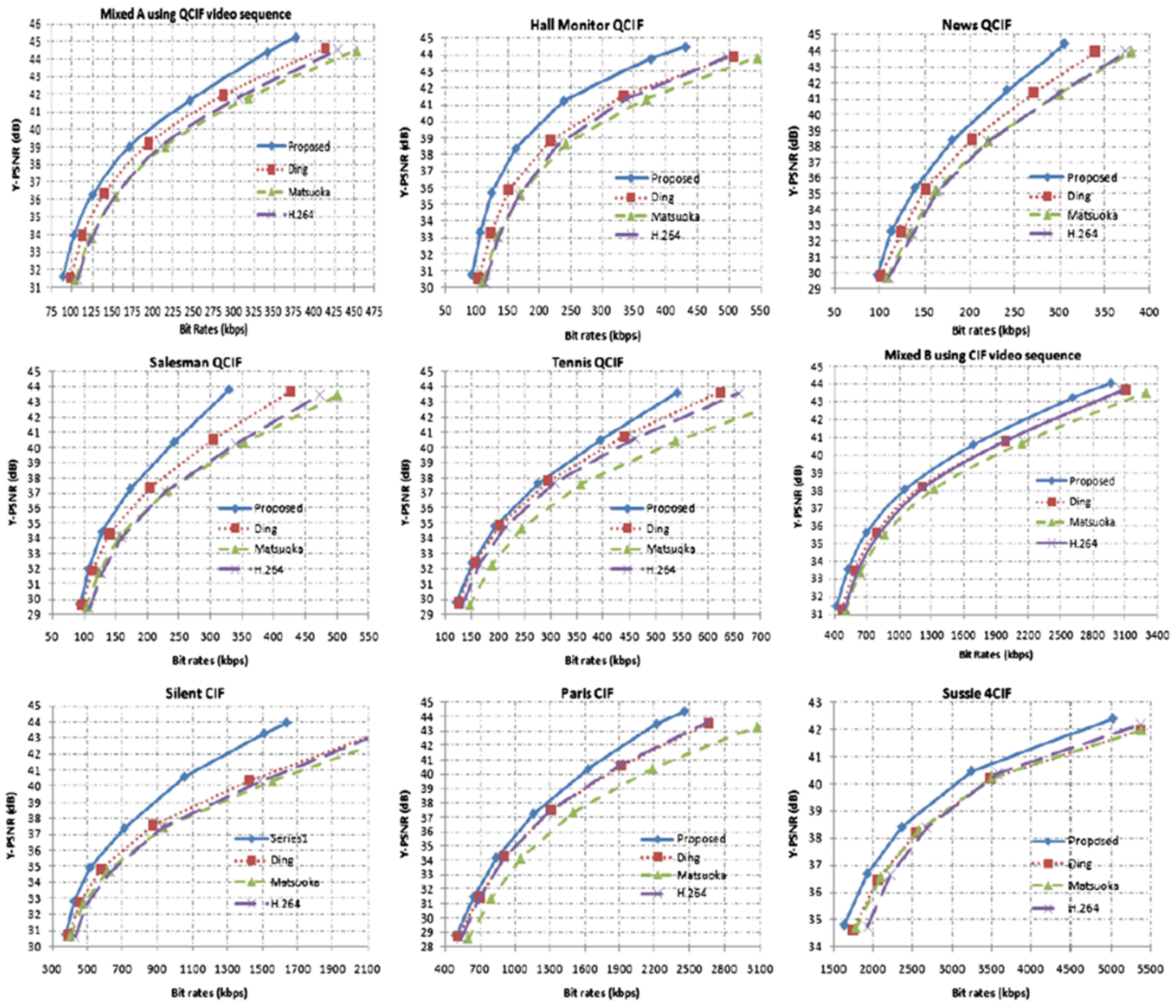

Figure 13 Rate-distortion performance of the proposed scheme with SCD, proposed scheme with fixed GOP, H.264 with two reference frame (H.264 2Ref), and Algorithm used in [26] using Tennis video sequence.

that a significant portion of coding gain is coming using McFIS as a second reference frame.

For the overall evaluation of the proposed scheme, Figure 10 shows the rate-distortion curves using the proposed (with SCD), Ding's, Matsuoka's, and the H.264 (with fixed GOP and 5 reference frames) algorithms for 2 mixed (each consisting of 11 CIF/QCIF videos) and 7 individual (4 QCIF, 2 CIF, and 1 4CIF) video sequences. The results from the figure confirm that the proposed scheme outperforms the H.264 as well as other two relevant state-of-the-art algorithms by 0.5–2.0 dB. The performance improvement by the proposed scheme is relatively high for *Salesman*, *Silent*, and *Hall Monitor* video sequences compared to the other sequences. This is due to the relatively larger background areas in these three cases, and hence a larger number of references are selected from the McFIS. On the other hand, the performance improvement by the proposed scheme is relatively lower for the *Tennis* and *Mixed B* video sequences due to the less number of reference MBs coming from the McFIS for camera movement.

5. Conclusions

In this article, the issue of effective, dynamic I-frame insertion, and reference frame (termed as the McFIS) generation in video coding has been tackled simultaneously with a Gaussian mixture-based model for dynamic background. To be more specific, the proposed method used the generated McFIS's inherent capability of SCD and adaptive GOP determination for integrated decision for efficient video coding. The McFIS is generated using real-time GMM. We have used dynamic

background (i.e., McFIS) as the second reference frame for efficient encoding of background. In essence, the new scheme allows moving object areas being referenced with the immediate previous frame while background regions are being referenced with McFIS.

We have proposed a DBM using decoded or distorted frames instead of original frames. This allows wider scope of use with DBM because raw video feeds (without any lossy compression) are usually not available and noise/error is inevitable especially in the case of wireless transmission.

By foreground and background referencing, we can improve rate-distortion performance in the uncovered background region which is almost impossible by the traditional multiple reference schemes. The proposed scheme effectively reduces computational complexity by limiting the reference frames into only two without sacrificing rate-distortion performance (actually it improves compared to the relevant existing algorithms). By introducing McFIS as a reference frame, we can avoid the complication of selecting long-term reference frame.

The proposed video coding technique outperforms the existing relevant schemes, in terms of rate-distortion and computational requirement. The experimental results show that the proposed technique detects scene changes more effectively compared to the two state-of-the-art algorithms, and outperforms them by 0.5–2.0 dB PSNR for coding quality. The proposed technique outperforms the H.264 with fixed GOP and five reference frames by 0.8–2.0 dB in PNSR and around 60% of reduced computational time.

Competing interests

This work is supported by the SINGAPORE MINISTRY OF EDUCATION Academic Research Fund (AcRF) Tier 2, Grant Number: T208B1218. A number of modifications and research works have been completed in Charles Sturt University when the first author, Manoranjan Paul, has been joined in Charles Sturt University, Australia.

Acknowledgements

This study was supported by the Singapore Ministry of Education Academic Research Fund (AcRF) Tier 2, Grant No. T208B1218.

Received: 16 March 2012 Accepted: 3 December 2012
Published: 31 January 2013

References

1. T Wiegand, GJ Sullivan, G Bjøntegaard, A Luthra, Overview of the H.264/AVC video coding standard. IEEE Transactions on Circuits and Systems for Video Technology **13**(7), 560–576 (2003)
2. ITU-T Recommendation H.264, *Advanced Video Coding for Generic Audiovisual Services*, 03/2009
3. M Paul, M Murshed, Video coding focusing on block partitioning and occlusions. IEEE Transactions on Image Processing **19**(3), 691–701 (2010)
4. J-R Ding, J-F Yang, Adaptive group-of-pictures and scene change detection methods based on existing H.264 advanced video coding information. IET Image Processing **2**(2), 85–94 (2008)
5. Y-W Huang, B-Y Hsieh, S-Y Chien, S-Y Ma, L-G Chen, Analysis and complexity reduction of multiple reference frames motion estimation in H.264/AVC. IEEE Transactions on Circuits and Systems for Video Technology **16**(4), 507–522 (2006)
6. L Shen, Z Liu, Z Zhang, G Wang, An adaptive and fast multi frame selection algorithm for H.264 video coding. IEEE Signal Processing Letters **14**(11), 836–839 (2007)
7. T-Y Kuo, H-J Lu, Efficient reference frame selector for H.264. IEEE Trans. Circuits Syst. Video Technol. **18**(3), 400–405 (2008)
8. K Hachicha, D Faura, O Romain, P Garda, Accelerating the multiple reference frames compensation in the H.264 video coder. J. Real-Time Image Process. (Springer) **4**(1), 55–65 (2009)
9. J Zheng, L-P Chau, Error-resilient coding of H.264 based on periodic macroblock. IEEE Trans. Broadcasting **52**(2), 223–229 (2006)
10. S Saponara, M Casula, F Rovati, D Alfonso, L Fanucci, Dynamic control of motion estimation search parameters for low complex H.264 video coding. IEEE Trans. Consum. Electron. **52**(1), 232–239 (2006)
11. A Dimou, O Nemethova, M Rupp, Scene change detection for H.264 using dynamic threshold techniques, in *Proceedings of the EURASIP Conference on Speech and Image Processing, Multimedia Communications and Service* (Slovak Republic, Smolenice, 2005), pp. 80–227
12. D Alfonso, B Biffi, L Pezzoni, Adaptive GOP size control in H.264/AVC encoding based on scene change detection, in *Signal Processing Symposium* (, Rejkjavik, 2006), pp. 86–89
13. S Matsuoka, Y Morigami, S Tian, T Shimamoto, Coding efficiency improvement with adaptive GOP size selection for H.264/SVC, in *International Conference on Innovative Computing Information and Control (ICICIC)* (Dalian, Liaoning, Dalian, Liaoning, 2008), pp. 356–359
14. T Song, S Matsuoka, Y Morigami, T Shimamoto, Coding efficiency improvement with adaptive GOP selection for H.264/SVC. Int. J. Innovative Comput. Inf Control **5**(11), 4155–4165 (2009)
15. D Hepper, Efficiency analysis and application of uncovered background prediction in a low bit rate image coder. IEEE Trans. Communication **38**, 1578–1584 (1990)
16. S-Y Chien, S-Y Ma, L-G Chen, Efficient moving object segmentation algorithm using background registration technique. IEEE Trans. Circuits Syst. Video Technol. **12**(7), 577–586 (2002)
17. T Totozafiny, O Patrouix, F Luthon, J-M Coutellier, Dynamic background segmentation for remote reference image updating within motion detection JPEG2000. IEEE International Symposium on Industrial Electronics, Montreal, Que **1**, 505–510 (2006)
18. R Ding, Q Dai, W Xu, D Zhu, H Yin, Background-frame based motion compensation for video compression, in *IEEE International Conference on Multimedia and Expo (ICME)* (, Taipei, vol. 2, 2004), pp. 1487–1490
19. C Stauffer, WEL Grimson, Adaptive background mixture models for real-time tracking, in *IEEE Conference on Computer Vision and Pattern Recognition* (, Fort Collins, CO, vol. 2, 1999), pp. 246–252
20. D-S Lee, Effective Gaussian mixture learning for video background subtraction. IEEE Trans. Pattern Anal. Mach. Intell. **27**(5), 827–832 (2005)
21. M Haque, M Murshed, M Paul, Improved Gaussian mixtures for robust object detection by adaptive multi-background generation, in *IEEE International Conference on Pattern Recognition* (, Tampa, FL, 2008), pp. 1–4
22. GVD Auwera, PT David, M Reisslein, Traffic and quality characterization of single-layer video streams encoded with the H.264/MPEG-4 advanced video coding standard and scalable video coding extension. IEEE Trans. Broadcast. **54**(3), 698–718 (2008)
23. D Wang, L Zhang, A Vincent, New method of reducing GOP-boundary artifacts in wavelet-based video coding. IEEE Trans. Broadcast. **52**(3), 350–355 (2006)
24. P List, A Joch, J Lainema, G Bjøntegaard, M Karczewicz, Adaptive deblocking filter. IEEE Trans. Circuits Syst. Video Technol. **13**(7), 614–619 (2003)
25. H Kimata, Y Yashima, N Kobuyashi, Edge preserving pre–post filtering for low bitrate video coding, in *IEEE International Conference on Image Processing* (, Thessaloniki, vol. 3, 2001), pp. 554–557
26. M Paul, W Lin, CT Lau, B-S Lee, McFIS: better I-frame for video coding, in *IEEE International Symposium on Circuits and Systems (IEEE ISCAS-10)* (Paris, Paris, 2010), pp. 2171–2174
27. K-W Wong, K-M Lam, W-C Siu, An efficient low bit-rate video-coding algorithm focusing on moving regions. IEEE Trans. Circuits Syst. Video Technol. **11**(10), 1128–1134 (2001)
28. M Paul, M Murshed, L Dooley, A real-time pattern selection algorithm for very low bit-rate video coding using relevance and similarity metrics. IEEE Trans. Circuits Syst. Video Technol. **15**(6), 753–761 (2005)
29. T Shanableh, M Ghanbari, Heterogeneous video transcoding to lower spatio-temporal resolutions and different encoding formats. IEEE Transaction Multimedia **2**(2), 101–110 (2000)

30. S Saponara, C Blanch, K Denolf, J Bormans, The JVT advanced video coding standard: complexity and performance analysis on a tool-by-tool basis, in *IMEC*, 2003
31. G Zhai, J Cai, W Lin, X Yang, W Zhang, Three-dimensional scalable video adaptation via user-end perceptual quality assessment. IEEE Transaction Broadcasting **54**(3), 719–727 (2008)

doi:10.1186/1687-6180-2013-11
Cite this article as: Paul *et al.*: Video coding with dynamic background. *EURASIP Journal on Advances in Signal Processing* 2013 **2013**:11.