

# Optical Engineering

SPIEDigitalLibrary.org/oe

## **Optimal complexity scalable H.264/AVC video decoding scheme for portable multimedia devices**

Hoyoung Lee  
Younghyeon Park  
Byeungwoo Jeon



# Optimal complexity scalable H.264/AVC video decoding scheme for portable multimedia devices

**Hoyoung Lee**  
**Younghyeon Park**  
**Byeungwoo Jeon**  
Sungkyunkwan University  
School of Electronic and Electrical Engineering  
300 Chunchun-Dong, Jangan-Gu  
Suwon 440-746, Gyonggi-Do, Republic of Korea  
E-mail: [bjeon@skku.edu](mailto:bjeon@skku.edu)

**Abstract.** Limited computing resources in portable multimedia devices are an obstacle in real-time video decoding of high resolution and/or high quality video contents. Ordinary H.264/AVC video decoders cannot decode video contents that exceed the limits set by their processing resources. However, in many real applications especially on portable devices, a simplified decoding with some acceptable degradation may be desirable instead of just refusing to decode such contents. For this purpose, a complexity-scalable H.264/AVC video decoding scheme is investigated in this paper. First, several simplified methods of decoding tools that have different characteristics are investigated to reduce decoding complexity and consequential degradation of reconstructed video. Then a complexity scalable H.264/AVC decoding scheme is designed by selectively combining effective simplified methods to achieve the minimum degradation. Experimental results with the H.264/AVC main profile bitstream show that its decoding complexity can be scalably controlled, and reduced by up to 44% without subjective quality loss. © The Authors. Published by SPIE under a Creative Commons Attribution 3.0 Unported License. Distribution or reproduction of this work in whole or in part requires full attribution of the original publication, including its DOI. [DOI: [10.1117/1.OE.52.7.071508](https://doi.org/10.1117/1.OE.52.7.071508)]

Subject terms: video decoder; complexity scalable decoding; H.264/AVC decoder.

Paper 121153SS received Aug. 9, 2012; revised manuscript received Apr. 4, 2013; accepted for publication Apr. 16, 2013; published online May 22, 2013.

## 1 Introduction

The proliferation of ubiquitous communication infrastructures makes various video services increasingly popular on portable multimedia devices. Although recent technological advancements have made real-time video playback possible on many portable multimedia devices, when it comes to high resolution and/or high quality video content, especially on mobile devices, there are still impending issues in real-time playback due to limited resources of portable devices in battery capacity, processor speed, and memory.

The problem addressed in this paper is slightly different from the conventional usage scenario of video decoders since it addresses what to do if an H.264/AVC video decoder is given a compressed bitstream exceeding its level specification. The level in a video coding standard specifies the minimal resources with which a given standard-conformant decoder shall be equipped. Encountering video contents exceeding a decoder's level specification is not rare in reality due to the recent ubiquity of various communication networks and the coexistence of portable devices with wide variability in their computation capacity. That is because video content available on mobile networks is accessible basically to every sort of devices, and some content may have a higher level specification of H.264/AVC than that of a receiving decoder. In such cases, currently a conventional decoder just refuses to decode. However, it would be much friendlier to users if the decoder were able to show the decoded pictures at a slightly lower quality that its available resources could provide with the best efforts. In order to do this, a video decoder should be able to flexibly decode the bitstream exceeding the level specification according to its available computing resources. This kind of simplified decoding capability is also quite essential

when a decoder knows *a priori* that fully compliant decoding is not necessary, e.g., in fast-forwarding of video, making thumbnails, or skimming through a video playback. As a whole, the issue in this problem is complexity scalable video decoding in accordance with the available resources in a decoder.

The aforementioned video playback capability itself is already implementable in a sense if the play-back quality is not concerned much. However, such a less careful approach would be practically useless due to the significant degradation of both objective and subjective quality resulting from the processing mismatch between the encoder and the decoder. A quality distortion caused by a complexity-reduced decoding process is propagated to subsequent pictures, and pretty soon, the quality degradation will become unbearable. Therefore, it is very important to carefully design a complexity scalable decoding algorithm which can manage optimal complexity control depending on the resource availability of the device.

Note that various complexity scalable video decoding algorithms have been already developed.<sup>1-19</sup> A major approach to complexity scalable decoding is to control the computational complexity of one or two decoding processes. Peng<sup>1</sup> proposed a discrete cosine transform (DCT)-based complexity scalable video decoder which controlled the decoding complexity by pruning out some DCT data in order to skip the inverse discrete cosine transform (IDCT) process. Peng and Zhong<sup>2</sup> proposed a selective B-residual decoding method based on computational resources and the energy level of B-residual blocks. Chen et al.<sup>3</sup> realized the complexity scalability by using both IDCT pruning and a simpler interpolation filter based on the frame type. Lei et al.<sup>4</sup> proposed a complexity scalable algorithm for

the audio and video coding standard in China (AVS) by using a loop filter along with a luminance interpolation filter scaling method. In this approach, the encoder sends complexity information about the loop filter and interpolation filter to the decoder for the complexity control. Meanwhile, Ji et al.<sup>5</sup> developed an energy scalable video decoding strategy for multimedia devices. Lee et al.<sup>6</sup> also worked towards complexity scalability by controlling the complexity of motion compensation and of the deblocking filter. Mahjoub et al.<sup>7</sup> proposed a complexity reduction method of the deblocking filter, and Lu et al.<sup>8</sup> optimized the context adaptive variable length coding (CAVLC) lookup tables to reduce the decoding complexity. de Oliveira et al.<sup>9</sup> optimized the inverse transform (IT) matrix for each frame, as a function of both content and quantization noise.

Several approaches also designed the complexity model of decoder for complexity scalable video decoding.<sup>10-13</sup> They include complexity models of H.264/AVC decoding parts such as motion compensation,<sup>10,11</sup> entropy decoding,<sup>12</sup> or the whole H.264/AVC decoder.<sup>13</sup>

As another approach to the complexity scalable algorithms, Park et al.<sup>14</sup> reduced the energy consumption of the MPEG-4 decoding process by using a re-quantization process that reduced the amount of data to process. Nam et al.<sup>15</sup> proposed a method using spatial downsizing decoding.

Some other approaches addressed the reduction of decoding complexity from the viewpoint of hardware architecture design.<sup>16-19</sup> Chao et al.<sup>16</sup> designed an optimized IT architecture to support multistandard video coding applications, and Wei et al.<sup>17</sup> proposed parallel decoding algorithms for multicore processors. Tsai et al.<sup>18</sup> designed a parallel level decoding method of CAVLC, and Sze and Chandrakasan<sup>19</sup> proposed a parallel context adaptive binary arithmetic coding (CABAC) decoding method. However, these hardware perspective approaches are not easy for achieving flexible control because of their hardwired characteristics.

In this paper, an algorithmic complexity scalable video decoding scheme for the H.264/AVC decoder of portable multimedia devices is investigated. First, each decoding element of H.264/AVC was studied in terms of both its complexity versus degradation performance from the viewpoint of complexity scalability and its complexity control parameters. By adjusting the complexity control parameters in a complexity-distortion (C-D) optimized way, the optimum complexity control levels which give the best performance in complexity reduction with minimal quality degradation were found.

The rest of this paper is structured as follows. Section 2 investigates the video decoding elements from the viewpoint

of complexity control and develops a complexity reduction method. Section 3 presents the proposed complexity scalable decoding method of each control element and the optimal complexity control level. Section 4 presents experimental results, and Sec. 5 concludes with discussion and some directions for possible future work.

## 2 Complexity Control of Video Decoding Elements

The H.264/AVC decoder is composed of several decoding elements as depicted in Fig. 1: a variable length decoder (VLD), an inverse quantizer (IQ), an IT, motion compensation, intra prediction, reconstruction, and deblocking filter. In previous research,<sup>6</sup> complexity of the H.264/AVC decoding elements in the main profile was evaluated, and its complexity profiling result<sup>6</sup> is summarized in Table 1 to show that the most complex decoding element is the motion compensation and variable length decoding (CABAC decoding), and that the second major complex element is the deblocking filter. "Others" in Table 1 represent operating system overheads such as file I/O for reading bitstreams from the file system.<sup>20</sup> Based on this profiling result, the motion compensation and deblocking filtering were chosen in this paper as targets for complexity control.

However, by noting the lossless nature of the entropy decoder, simplification of CABAC is not considered since even small degradation from compromised entropy decoding can result in fatal decoding errors. Since the complexity control range using the motion compensation and deblocking filter were not sufficient, a macroblock (MB) decoding skipping method was further developed to provide more flexibility in the complexity control.

### 2.1 Complexity Reduction of the Motion Compensation

In the H.264/AVC motion compensation, motion predicted values at half-pel samples were generated by horizontal or vertical one-dimensional 6-tap finite impulse response (FIR) interpolation filtering, while quarter-pel samples were generated by averaging the nearest half-pel and integer-pel samples.<sup>21</sup> Figure 2 depicts quarter-pel samples for luma components. Since the interpolation process accounts for most of the complexity of motion compensation, its complexity for the luma component is modeled as

$$C_{mc\_luma} = C_{Int\_pel} + C_{Half\_pel} + C_{Quarter\_pel}, \quad (1)$$

where  $C_{mc\_luma}$  is the total computational complexity of the motion compensation in a decoder;  $C_{Int\_pel}$ ,  $C_{Half\_pel}$ , and  $C_{Quarter\_pel}$  are the computational complexity for generating

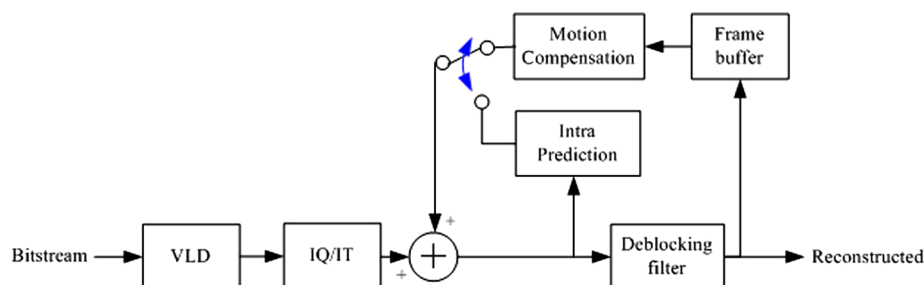
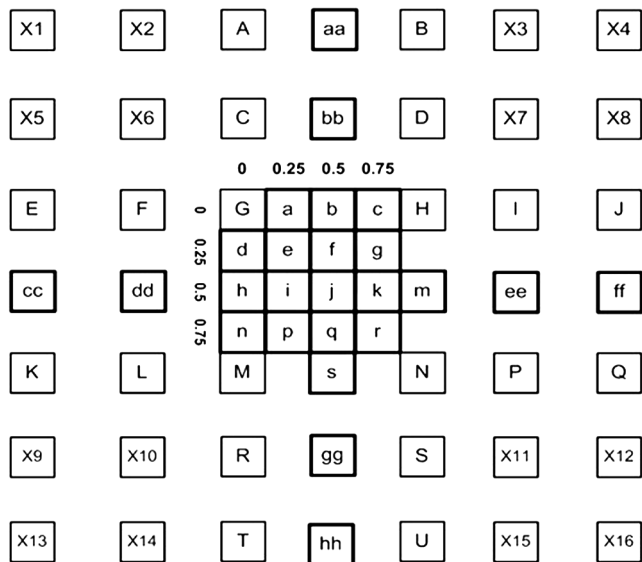


Fig. 1 Block diagram of the H.264/AVC decoding process.

**Table 1** Complexity profile result of H.264/AVC decoding elements (Ref. 6).

Decoding elements	Complexity (%)
Motion compensation	27.51
Variable length decoding(VLD)	25.19
Deblocking filter	16.65
inverse quantizer/inverse transform(IQ/IT)	10.65
Reconstruction	3.08
Intra prediction	0.57
Others	16.34

integer-pel samples (i.e., for sample copy from its reference picture), 6-tap FIR interpolation filtering to generate half-pel samples, and an additional averaging process of quarter-pel samples after generating half-pel samples, respectively. Table 2 shows the interpolation filtering process based on the quarter-pel sample positions in Fig. 2 and their normalized complexity with respect to  $C_{Int\_pel}$ .<sup>6</sup> Note the nonidentical computational complexity: it depends on the sample positions. Quarter-pel positions at  $f, i, k, q$  are the most complex since they require seven 6-tap filtering and one 2-tap filtering. On the other hand, those half-pel samples labeled  $b, h$  are the least complex ones — they require just one time of 6-tap filtering. This observation suggests that the motion compensation complexity depends critically on the complexity of the interpolation filter. Therefore, to reduce the complexity, the 6-tap interpolation filter can be replaced by 2-tap or 4-tap filter, depending on the subpel position, with filter coefficients as shown in Table 3. Note that the filter coefficients of the 4-tap filter are used in a scalable extension of



**Fig. 2** Fractional sample positions for quarter sample luma interpolation.

**Table 2** Complexity comparison of interpolation in H.264/AVC decoding.

Sample position	Interpolation operation	Normalized Complexity w.r.t. G(0,0)
G(0,0)	Integer (pixel copy)	1.00
b(0.5,0)	6-tap	2.96
h(0.5,0)	6-tap	2.87
a(0.25,0)	6-tap + 2-tap	3.64
c(0.75,0)	6-tap + 2-tap	3.50
d(0,0.25)	6-tap + 2-tap	3.49
n(0,0.75)	6-tap + 2-tap	3.60
e(0.25,0.25)	(6-tap) × 2 + 2-tap	5.59
g(0.25,0.75)	(6-tap) × 2 + 2-tap	5.70
p(0.75,0.25)	(6-tap) × 2 + 2-tap	5.64
r(0.75,0.75)	(6-tap) × 2 + 2-tap	5.82
j(0.5,0.5)	(6-tap) × 6 + 6-tap	7.19
f(0.5,0.25)	(6-tap) × 6 + 6-tap + 2-tap	7.79
i(0.5,0.75)	(6-tap) × 6 + 6-tap + 2-tap	7.28
k(0.25,0.5)	(6-tap) × 6 + 6-tap + 2-tap	7.11
q(0.75,0.5)	(6-tap) × 6 + 6-tap + 2-tap	7.78

H.264/AVC<sup>22</sup> for inter-layer intra prediction, and those of the 2-tap filter are generated by simplifying the H.264/AVC interpolation filter with adjacent int-pel samples.<sup>6</sup>

Prediction values for chroma sample positions are generated by bilinear interpolation of four neighboring integer samples using

$$a = ((8 - xFrac_c) \times (8 - yFrac_c) \times A + xFrac_c \times (8 - yFrac_c) \times B + (8 - xFrac_c) \times yFrac_c \times C + xFrac_c \times yFrac_c \times D + 32) / 64, \tag{2}$$

**Table 3** Complexity reduction method for interpolation filtering.

position	1-D filter coefficient	
	2-tap	4-tap
1/4	[48, 16]/64	[-3, 51, 19, -3]/64
1/2	[32, 32]/64	[-3, 19, 19, -3]/64
3/4	[16, 48]/64	[-3, 19, 51, -3]/64

where  $a$  is a predicted chroma sample value and  $A, B, C, D$  are the integer-pel samples.  $x\text{Frac}_c$  and  $y\text{Frac}_c$  are the fractional offsets of the predicted sample. To reduce the complexity for chroma interpolation filtering, the predicted chroma sample is just copied from the nearest neighboring integer samples.

### 2.2 Complexity Reduction of the Deblocking Filter

Since H.264/AVC performs block-based transform and lossy quantization of integer DCT coefficients, blocking artifacts occur in the reconstructed picture. To eliminate the blocking artifacts, a deblocking filter is applied both in the encoder and in the decoder. This deblocking filter has two processes: block boundary strength (Bs) decision and actual pixel filtering using the determined Bs. At each  $4 \times 4$  block boundary, Bs is determined as one integer between 0 and 4 according to the rules<sup>23</sup> based on whether or not it is a MB boundary, whether its blocks have intra/inter prediction mode, whether it has nonzero DCT coefficients, its motion vector, its reference picture index, etc. Subsequently, the actual filtering process is applied to each  $4 \times 4$  block boundary depending on the selected value of Bs. In case of Bs = 4, a special filter is applied according to the specific condition.<sup>23</sup> When Bs is from 1 to 3, a normal filter is applied.<sup>23</sup> Therefore, the computational complexity of the deblocking filter is modeled as

$$C_{\text{deblocking\_filter}} = C_{\text{Bs\_decision}} + C_{\text{filtering}}, \quad (3)$$

where  $C_{\text{deblocking\_filter}}$  is the computational complexity of the deblocking filtering,  $C_{\text{Bs\_decision}}$  is the computational complexity of the Bs decision process, and  $C_{\text{filtering}}$  is the computational complexity of the actual pixel filtering process. The complexity reduction of deblocking can be designed in two ways — by reducing  $C_{\text{Bs\_decision}}$  or  $C_{\text{filtering}}$ .

In order to reduce  $C_{\text{Bs\_decision}}$ , a simplified Bs decision process in the previous research<sup>6</sup> was proposed based on some observations.<sup>6,24</sup> The proposed simplified Bs decision process contained the same Bs decision rules as the previous

research,<sup>6</sup> but different Bs values according to the rules as depicted in Fig. 3.

A slightly different filtering method compared to the previous research<sup>6</sup> was also designed by adjusting filtering tap size and the number of samples to be filtered according to Bs to reduce  $C_{\text{filtering}}$ . When  $Bs \geq 3$ , the proposed simplified filtering method is the same as that of H.264/AVC. On the other hand, if  $Bs = 2$ , only one nearest pixel from each side of the block boundary (i.e.,  $p_0$  and  $q_0$ ) is filtered<sup>23</sup> — that is, unlike the H.264/AVC, the second immediate pixels from the block boundaries ( $p_1$  and  $q_1$ ) are not filtered.<sup>23</sup> In the  $Bs = 1$  case, the filtering complexity is reduced by using a 2-tap FIR filter which is applied only to the  $p_0$  and  $q_0$  samples.

### 2.3 Complexity Reduction by MB Decoding Skipping

For further complexity scalability, a MB was skipped from the whole decoding process after executing VLD. This method is similar to frame skipping for frame rate control in transcoders.<sup>25-28</sup> Reduction in temporal resolution due to the frame skipping may cause noticeably perceivable motion jerkiness and consequently significant subjective quality loss can follow. To prevent such mishaps, existing approaches selectively skip frames that satisfy some conditions, for examples, scene change,<sup>25</sup> motion activity,<sup>26,27</sup> or motion continuity.<sup>28</sup> Such skipping methods were also used for the proposed complexity scalable video decoder. That is, the MB decoding in a B slice was skipped when the MB satisfied three conditions of MB coding type, coded block pattern ( $cbp$ ) value, and motion activity. If an MB (in inter-coded slice or picture) was determined to be intra-coded, its correlation with the blocks in reference picture ( $s$ ) must be low. Therefore, an intra-coded MB should not be skipped from the decoding since otherwise its pixel values cannot be faithfully reproduced. On the other hand, if an inter-coded MB has no nonzero coded coefficients (that is,  $cbp = 0$ ), then it was safe to assume that the MB was highly correlated to its reference block, and it was possible to estimate the MB quite faithfully from its reference. Motion

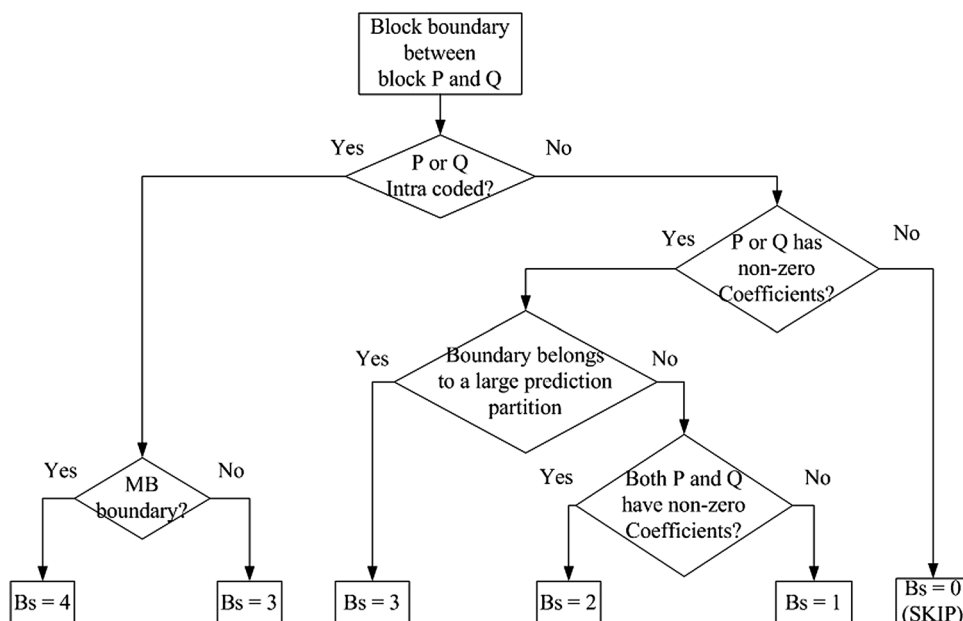


Fig. 3 Proposed simplified block boundary strength (Bs) decision.



activity can indicate whether the motion of the current MB is fast or slow. If an MB with fast motion is skipped from decoding, its estimated reconstruction is highly likely to have noticeable motion jerkiness. Therefore, the decoding skipping also needs to check the motion activity of the current MB which can be calculated by

$$mv_{MA} = \frac{1}{16} \sum_{n=0}^{15} (|mv_x^n| + |mv_y^n|), \quad (4)$$

where  $mv_{MA}$  is the motion activity of current MB, and  $(mv_x^n, mv_y^n)$  is the motion vector of the  $n$ th  $4 \times 4$  block ( $n = 0 \sim 15$ ) inside a current MB. In the proposed method, an inter-coded MB having  $cbp = 0$  and motion activity ( $mv_{MA}$ ) less than a threshold  $T_{MA}$  was skipped from further decoding after VLD. In the picture play-out, the skipped MB was generated by a very simple reconstruction method whose complexity was much lower than the actual decoding process. Such reconstruction methods for the skipped MB have already been studied extensively in error concealment problems.<sup>29-31</sup> Those skipped MBs were reconstructed by motion compensation using motion vectors and the reference index of the current MB. For example, if the current MB was predicted to be from list 0, its reconstructed MB was motion-compensated from the reference slice in the list 0 memory. If the current MB was bi-predicted, the reconstructed signal was formed by averaging the motion-compensated signals from the list 0 and list 1 memory.

### 3 Proposed Complexity Scalable Decoding Scheme

The complexity scalable video decoding scheme should satisfy the following optimality criterion

$$\max C(x_1, x_2, \dots, x_n) \text{ subject to } \min D(x_1, x_2, \dots, x_n), \quad (5)$$

where  $C(x_1, x_2, \dots, x_n)$  and  $D(x_1, x_2, \dots, x_n)$  denote, respectively the complexity reduction and consequent quality loss when the decoding complexity is controlled by the complexity control parameters  $x_1, x_2, \dots, x_n$ . By controlling the complexity control parameters, users can control the amount of complexity reduction. Under the constraints on available computing resources, the proposed complexity scalable video decoder can achieve the minimum quality loss while maximizing complexity reduction by controlling the selected control parameters. To find an optimal control level of those parameters, the C-D performance with selected complexity control parameters is evaluated first. Next three decoding complexity control parameters are discussed: motion compensation, the deblocking filter, and the MB decoding process skipping method.

#### 3.1 Complexity Scalable Method for the Motion Compensation

To reduce the complexity of motion compensation, simplified interpolation filtering methods were developed as depicted in Sec. 2. The most effective complexity scalability for motion compensation can be achieved by selectively using the simplified interpolation filtering methods according to a specified scalability level. The proposed four motion

**Table 4** Motion compensation complexity reduction (MCR) levels.

MCR <sub>Level</sub>	Interpolation filter
0	6-tap (no reduction)
1	4-tap (only <i>B</i> -slice)
2	2-tap (only <i>B</i> -slice)
3	4-tap ( <i>P</i> -slice) 2-tap ( <i>B</i> -slice)

compensation complexity reduction levels (MCR<sub>Level</sub>) are shown in Table 4. When MCR<sub>Level</sub> = 0, the motion compensation had the maximum complexity, which is the same as the conventional H.264/AVC motion compensation. When MCR<sub>Level</sub> = 1 or 2, simplified methods were applied to the *B* slice only. This gives the minimal degradation on the video quality since the degradation in the *B*-slice is not propagated to other slices unless the stored *B*-slice is used. When MCR<sub>Level</sub> = 3, simplified motion compensation methods were used for the *P*-slices (4-tap filter) and the *B*-slices (2-tap filter). Therefore, encoder-decoder mismatch could occur, however, in return, the decoder achieved a significant reduction in complexity. For chroma components, when MCR<sub>Level</sub> > 0, the simplified method from Sec. 2.1 for chroma was applied.

#### 3.2 Complexity Scalable Method for the Deblocking Filter

To control the complexity of the deblocking filter, a simplified Bs decision and deblocking method were applied as shown in Table 5, based on the six deblocking filter complexity reduction levels (DFR<sub>Level</sub>). When DFR<sub>Level</sub> = 0, the conventional H.264/AVC deblocking filter was used without any complexity reduction. As the DFR<sub>Level</sub> increased, simplified methods of deblocking in Sec. 2 were applied one by one. When DFR<sub>Level</sub> = 4, the deblocking filtering was applied only to the MB boundary with the simplified Bs decision and filtering method in Sec. 2. To achieve the maximum complexity reduction in the deblocking filter, the deblocking

**Table 5** Deblocking filter complexity reduction (DFR) levels.

DFR <sub>Level</sub>	Bs decision	Deblocking filtering
0	H.264/AVC method	H.264/AVC method
1	H.264/AVC method	simplified method
2	simplified method	H.264/AVC method
3	simplified method	simplified method
4	simplified method (only MB boundary)	simplified method (only MB boundary)
5	Forced deblocking filter off	Forced deblocking filter off

**Table 6** Complexity reduction (MDR) levels for macroblock decoding processing skipping.

MDR <sub>Level</sub>	Reduction method
0	No reduction
1	Conditional skip decoding of macroblock
2	Forced skip decoding of inter-coded macroblock

filter process to whole slices was switched off when DFR<sub>Level</sub> = 5.

**3.3 Complexity Scalable Method for MB Decoding Skipping**

The MB decoding process skipping has three levels which are described in terms of their MB decoding reduction level (MDR<sub>Level</sub>), as in Table 6. If MDR<sub>Level</sub> = 1, those MBs satisfying the skip conditions (inter-coded MB, *cbp* = 0, *mv*<sub>MA</sub> < *T*<sub>MA</sub>) were not decoded, that is, were skipped from decoding. On the other hand, if MDR<sub>Level</sub> = 2, the inter-coded MBs were skipped from the decoding process, thus saving a tremendous amount of computation.

**3.4 Proposed Complexity Scalable Decoding Scheme**

The previous subsections discussed how to realize decoding complexity scalability individually for each key decoding element. In this subsection, the same problem is discussed but focusing specifically on the best scalable control of total video decoding by optimally adjusting the three control parameters MCR<sub>Level</sub>, DFR<sub>Level</sub>, and MDR<sub>Level</sub> together. A block diagram of the proposed complexity scalable video decoder is shown in Fig. 4.

Let us define a total complexity reduction level (TCR<sub>Level</sub>) as a function of the three levels,

$$TCR_{Level} = f(MCR_{Level}, DFR_{Level}, MDR_{Level}), \quad (6)$$

where *f*(·) is a complexity control function with all three individual control levels as input. Note that each level has

different effectiveness in complexity control and that consequently the degradation varies in quality. To find the optimal combination of the control parameters to maximize the total complexity reduction subject to a constraint on the minimum quality degradation, the C-D performance<sup>6,32</sup> was compared by using AST and ΔPSNR which are respectively defined as

$$AST(\%) = \frac{DT(Anchor) - DT(proposed)}{DT(Anchor)} \times 100, \quad (7)$$

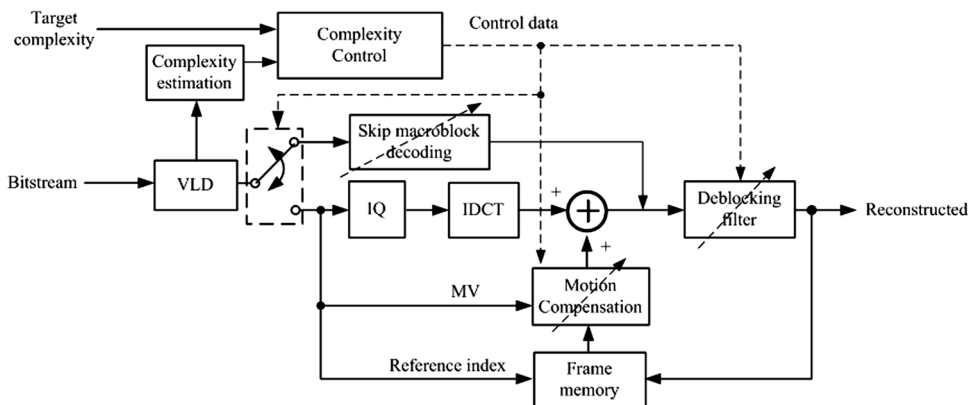
$$\Delta PSNR[dB] = PSNR(proposed) - PSNR(Anchor), \quad (8)$$

where AST(%) is an average saving time and DT(·) is a total decoding time of a decoder. ΔPSNR [dB] is the difference in objective quality as measured in the peak signal-to-noise ratio (PSNR) between the proposed method and the anchor. Here the “anchor” represents the conventional H.264/AVC decoder which corresponds to the maximum complexity (i.e., TCR<sub>Level</sub> = MCR<sub>Level</sub> = DFR<sub>Level</sub> = MDR<sub>Level</sub> = 0), and the “proposed” represents the complexity-reduced decoding scheme according to the selected control level.

The C-D curves are shown in Fig. 5; each curve represents the complexity reduction versus quality degradation obtained by controlling the individual level parameter independently (MCR<sub>Level</sub> in Table 4, DFR<sub>Level</sub> in Table 5, and MDR<sub>Level</sub> in Table 6, respectively). By controlling each parameter, the decoder complexity is reduced by up to 13.6% by MCR<sub>Level</sub>, 12.9% by DFR<sub>Level</sub>, and 25.9% by MDR<sub>Level</sub>. The quality degradations compared to the maximum complexity reduction are -1.66, -1.84, and -3.19 dB, respectively. As in Fig. 5, it was verified that these three parameters were suitable for complexity control of a decoder.

The joint C-D curve in terms of all three control parameters together (MCR<sub>Level</sub>, DFR<sub>Level</sub>, and MDR<sub>Level</sub>) is shown in Fig. 6. The optimal complexity control points which have the maximum complexity reduction subject to minimum distortion are specified by a line in Fig. 6. Therefore, if a decoder adjusts the control parameters following the line, the optimal decoding complexity scalability can be attained.

Table 7 shows the control level of each parameter according to TCR<sub>Level</sub>, while its relative complexity reduction and quality degradation are shown in Fig. 6. As shown in Table 7, the expected maximum complexity reduction was up to 41% compared with TCR<sub>Level</sub> = 0. Since the choice of



**Fig. 4** Block diagram of the proposed complexity scalable H.264/AVC decoder.

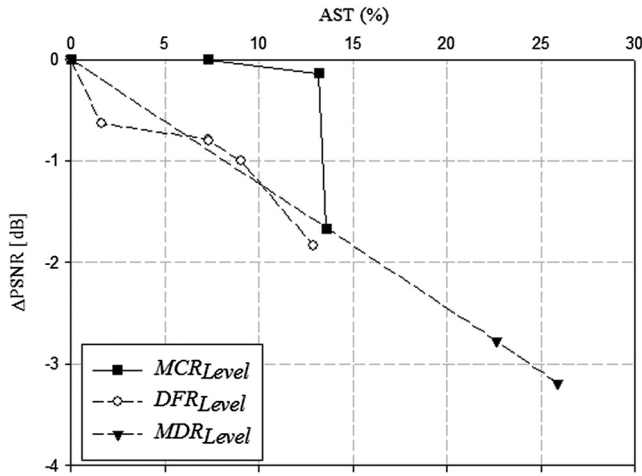


Fig. 5 Complexity-distortion curve of each individual control level.

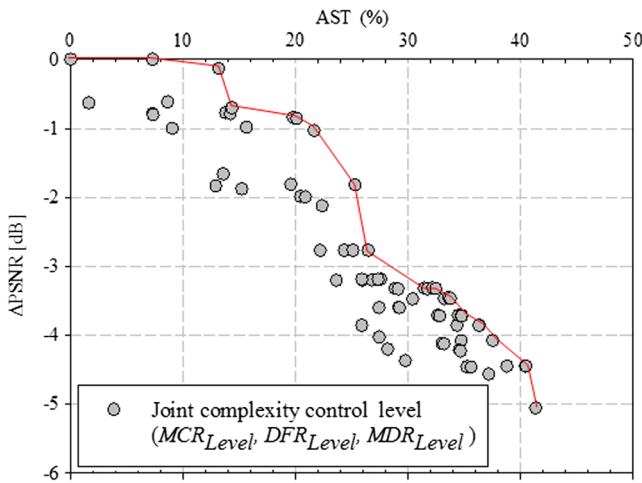


Fig. 6 Joint complexity-distortion curve.

$TCR_{Level} = 15$  in Table 7, compared to  $TCR_{Level} = 14$ , reduced the complexity by not more than 1% but incurred additional quality degradation of about 0.6 dB, the maximum complexity reduction level was set to 14 instead of 15. Therefore, the complexity of a decoder was adjusted in up to 15 steps. When  $TCR_{Level}$  was 0, the decoder has no complexity reduction, that is, the same complexity as conventional H.264/AVC decoder. As the  $TCR_{Level}$  increased, the complexity control level of each parameter was adjusted according to Table 7.

#### 4 Experimental Results

In order to assess the performance of the proposed scheme, it was implemented on JM18.0 H.264/AVC reference software. Bitstreams for experiments were coded for the H.264/AVC main profile with group of pictures (GOP) size = 60 under IBPBP structure (Here, I, B, and P represents Intra, Bi-predictive, and Predictive picture, respectively). The number of reference frames was 5, and one picture was coded as one slice. The quantization parameter (QP) was set to 22, 27, 32, and 37. The video sequences used for performance evaluation were *Bigships*, *City\_corr*, *Night*, and *Crew* (1280 × 720@60 fps). The performance of the proposed

Table 7 Joint complexity reduction level.

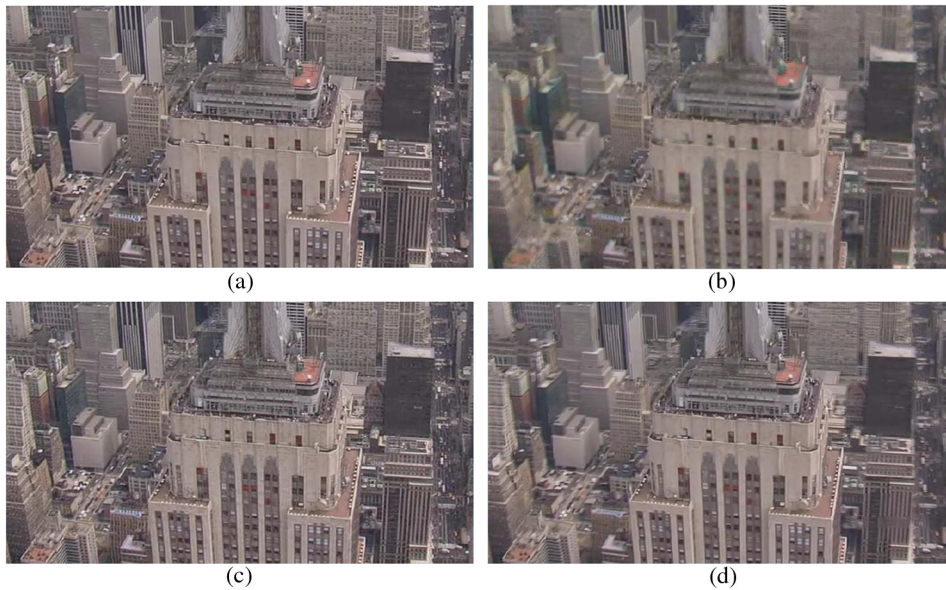
$TCR_{Level}$	$MDR_{Level}$	$MCR_{Level}$	$DFR_{Level}$	$\Delta PSNR$ [dB]	AST(%)
0	0	0	0	0.00	0.00
1	0	1	0	0.00	7.29
2	0	2	0	-0.13	13.17
3	0	2	1	-0.71	14.34
4	0	3	2	-1.98	20.45
5	0	2	4	-1.03	21.64
6	0	2	5	-1.82	25.29
7	0	3	5	-2.77	26.44
8	1	1	2	-3.32	31.43
9	1	2	3	-3.33	32.45
10	1	2	5	-3.47	33.76
11	2	1	3	-3.72	34.78
12	2	1	4	-3.86	36.33
13	1	1	5	-4.07	37.52
14	2	2	5	-4.45	40.47
15	2	3	5	-5.06	41.33

scheme was measured by AST (%) in Eq. (7) and  $\Delta PSNR$  [dB] in Eq. (8).

Figure 7 shows how the subjective quality changes when each control parameter of complexity ( $MCR_{Level}$ ,  $DFR_{Level}$ , and  $MDR_{Level}$ ) is individually adjusted. Figure 7(a) corresponds to the maximum complexity level ( $TCR_{Level} = 0$ , i.e.,  $MCR_{Level} = 0$ ,  $DFR_{Level} = 0$ , and  $MDR_{Level} = 0$ ) while the others, Fig. 7(b)–7(d), respectively, correspond to when each control parameter is changed to its maximum reduction value individually [Fig. 7(b):  $MCR_{Level} = 3$ ; Fig. 7(c):  $DFR_{Level} = 5$ ; Fig. 7(d):  $MDR_{Level} = 2$ ]. Figure 7 shows that while the maximum complexity reduction of deblocking filtering ( $DFR_{Level} = 5$ ) in Fig. 7(c) gives similar subjective quality as the no complexity reduction case in Fig. 7(a), the sharpness of the picture is degraded when  $MCR_{Level}$  or  $MDR_{Level}$  are adjusted to those of the minimum complexity level ( $MCR_{Level} = 3$ ,  $MDR_{Level} = 2$ ). The sharpness degradation appears in all regions in Fig. 7(b), but it is less apparent in the low motion area in Fig. 7(d) (see the middle building in the picture). As depicted in Fig. 7, although control parameters had some degradation in the minimum complexity level, they still maintained proper overall subjective quality.

Table 8 shows experimental results of the proposed scheme, indicating that it attains complexity scalability. The proposed scheme reduces the decoding complexity by up to 44% (in *City\_corr* sequence,  $TCR_{Level} = 14$ ) as compared





**Fig. 7** Subjective quality comparison between the maximum complexity level and the minimum complexity level of control parameters (*city* sequence, 118th frame, QP22) (a)  $TCR_{Level} = 0$  (maximum complexity), (b)  $MCR_{Level} = 3$ , (c)  $DFR_{Level} = 5$ , (d)  $MDR_{Level} = 2$ .

**Table 8** Experimental results of the proposed scheme.

$C_{target}$	$TCR_{Level}$	Bigships		City_corr		Night		Crew	
		$\Delta PSNR$ [dB]	AST(%)	$\Delta PSNR$ [dB]	AST(%)	$\Delta PSNR$ [dB]	AST(%)	$\Delta PSNR$ [dB]	AST(%)
0	0	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
10	2	-0.14	11.73	-0.35	16.31	-0.05	13.02	-0.11	9.94
15	3	-0.69	16.20	-0.80	18.14	-0.74	15.27	-0.81	12.00
20	4	-1.87	22.50	-2.70	22.08	-2.61	21.22	-1.72	17.72
25	6	-1.52	26.72	-1.25	27.72	-1.77	26.66	-2.32	22.37
30	8	-2.45	32.45	-4.42	32.09	-2.70	29.75	-2.28	25.95
35	11	-2.87	36.11	-4.77	36.43	-3.51	32.12	-3.35	28.81
40	14	-4.03	41.96	-5.91	44.50	-5.13	40.02	-4.66	37.03



**Fig. 8** Intra coded macroblock (shown in box) in an inter slice.

with the conventional H.264/AVC decoder. In Table 9,  $C_{target}$  represents the target complexity reduction and  $TCR_{Level}$  is the complexity control level to achieve the complexity reduction up to  $C_{target}$ . Reduced complexity for the  $TCR_{Level}$  is similar to the  $C_{target}$  in most cases. However, the reduced complexity does not reach the target complexity  $C_{target}$  in the *Crew* sequence. This is because the proposed scheme does not consider an intra-coded MB as a candidate for complexity control, therefore, there may exist a problem of insufficiency in the adjustable range of complexity especially if there are many intra MBs. The many intra-coded MBs (boxed areas) in the *Crew* sequence are shown in Fig. 8 in an inter slice. This behavior is due to flash lights). This is one area for future extension of the proposed method.

Table 9 shows the distortion and complexity reduction performance of the proposed method compared to the

**Table 9** Performance comparison of the proposed scheme with previous methods (Refs. 4, 6).

$C_{\text{target}}$	Method	Bigships		City_corr		Night		Crew	
		$\Delta\text{PSNR}$ [dB]	AST(%)	$\Delta\text{PSNR}$ [dB]	AST(%)	$\Delta\text{PSNR}$ [dB]	AST(%)	$\Delta\text{PSNR}$ [dB]	AST(%)
10	Method 1 (Ref. 4)	-1.33	11.83	-0.99	10.74	-1.63	14.36	-0.01	6.56
	Method 2 (Ref. 6)	-0.40	8.73	-0.70	12.89	-0.31	9.37	-0.06	5.91
	Proposed	-0.14	11.73	-0.35	16.31	-0.05	13.02	-0.11	9.94
20	Method 1 (Ref. 4)	-4.94	18.91	-7.54	16.27	-9.78	20.17	-2.22	15.48
	Method 2 (Ref. 6)	-2.23	23.89	-1.33	22.84	-2.64	24.88	-2.13	21.29
	Proposed	-1.87	22.50	-2.70	22.08	-2.61	21.22	-1.72	17.72



**Fig. 9** Comparison of subjective quality of decoded frames (*city* sequence, 118th frame, QP22) (a)  $\text{TCR}_{\text{Level}} = 0$  (maximum complexity), (b)  $\text{TCR}_{\text{Level}} = 2$ , (c)  $\text{TCR}_{\text{Level}} = 4$ , (d)  $\text{TCR}_{\text{Level}} = 6$ , (e)  $\text{TCR}_{\text{Level}} = 11$ , (f)  $\text{TCR}_{\text{Level}} = 14$  (minimum complexity).





**Fig. 10** Comparison of subjective quality on mobile device. (a)  $TCR_{Level} = 0$  (maximum complexity) (b)  $TCR_{Level} = 14$  (minimum complexity).

previous research.<sup>4,6</sup> For a fair comparison, previous research<sup>4,6</sup> was implemented on the JM18.0 H.264/AVC reference software. In Table 9, methods 1 and 2, respectively represent methods of research from Refs. 4 and 6. The reduced complexity according to the  $C_{target}$  and the relative distortion compared with the conventional H.264/AVC decoder were measured. Since method 2 reduces the complexity by up to 25% and method 1 exploits the  $B$  picture decoding skip method to reduce the complexity by more than 30%,  $C_{target}$  was only considered for 10% and 20% in this experiment. Compared to methods 1 and 2, the proposed method is better in objective quality loss. Since those methods reduce the complexity of the interpolation filtering process regardless of picture type — that is, through the  $P$  picture or stored- $B$  picture which are used as references of following pictures — error will be propagated to the following pictures and the objective quality loss was much higher than the proposed method. Complexity control of the proposed method was also more accurate than methods 1 and 2, and the reduced complexity of proposed method was more similar to  $C_{target}$  than these two methods.

Figure 9 shows a subjective quality comparison according to the  $TCR_{Level}$  represented in Table 8 from the 118th (the last decoded frame in one GOP) frame in *City\_corr* sequence coded with QP 22 (high quality). Even if the  $TCR_{Level}$  increases, the decoded pictures still have acceptable subjective quality, although the objective quality degradation becomes larger.

To verify the proposed scheme on mobile devices, it was also implemented on a mobile device. Figure 10 shows subjective quality with the maximum complexity ( $TCR_{Level} = 0$ ) and the minimum complexity ( $TCR_{Level} = 14$ ) on a mobile device. Since the mobile device is not able to show the original resolution of the video sequence, it only displays a downsized picture after performing a downsizing process. As shown in Fig. 10, the subjective quality of the decoded picture compared to the one of maximum complexity was acceptable even with the minimum complexity.

## 5 Conclusion

This paper presented a complexity scalable H.264/AVC decoding scheme for portable multimedia devices. The proposed method controls the motion compensation, deblocking filtering, and MB decoding skipping process by adjusting these three complexity control parameters. Its C-D performance was evaluated according to the controlling parameters, and the optimal complexity control levels for each parameter were sought. The proposed scheme can control

the decoding complexity with variable complexity control levels without significant subjective quality loss. Since the current scheme can adjust the decoding complexity of inter MBs only, future work may extend it to include also the decoding complexity controlling capability of intra MBs.

## Acknowledgments

This work was supported in part by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MEST) (No. 2011-001-7578).

## References

1. S. Peng, "Complexity scalable video decoding via IDCT data pruning," in *Proc. IEEE Conf. on Consumer Electron.*, pp. 74–75, IEEE, Los Angeles, California (2001).
2. S. Peng and Z. Zhong, "Resource-constrained complexity-scalable video decoding via adaptive  $B$ -residual computation," *Proc. SPIE* **4671**, 1165–1174 (2002).
3. Y. Chen et al., "Regulated complexity scalable MPEG-2 video decoding for media processors," *IEEE Trans. Circ. Syst. Video Technol.* **12**(8), 678–687 (2002).
4. C. Lei, Y. Chen, and W. Ji, "A complexity scalable decoder in an AVS video codec," in *Proc. the 6th Int. Conf. on Advances in Mobile Computing and Multimedia (MoMM '08)*, pp. 35–39, ACM, New York (2008).
5. W. Ji et al., "ESVD: an integrated energy scalable framework for low-power video decoding systems," *EURASIP J. Wireless Commun. Netw.* **2010**, 234131 (2010).
6. H. Lee et al., "Complexity scalable video decoding scheme for H.264/AVC," in *Proc. the Third Int. Conf. on Advances in Multimedia (MMEDIA '11)*, pp. 18–22, IARIA, Budapest, Hungary (2011).
7. W. H. Mahjoub, H. Osman, and G. M. Aly, "H.264 deblocking filter enhancement," in *Proc. Int. Conf. on Computer Engineering & systems (ICCES'11)*, pp. 219–224, IEEE, Cairo, Egypt (2011).
8. D. Lu, G. Liu, and L. Zhu, "An optimization for CAVLC code table lookup algorithm in H.264 decoder," in *Proc. Int. Symp. on Intelligence Info. Process. and Trusted Computing (IPTC'11)*, pp. 79–82, IEEE, Wuhan, China (2011).
9. R. G. de Oliveira, M. Trocan, and B. Pesquet-Popescu, "An H.264/AVC inverse transform adaptation method for video streaming applications," in *Proc. the 20th European Signal Processing Conference (EUSIPCO'12)*, pp. 2762–2766, IEEE, Bucharest, Romania (2012).
10. S. Lee and C. Jay Kuo, "Complexity modeling for motion compensation in H.264/AVC decoder," in *Proc. IEEE Int. Conf. on Image Processing (ICIP'07)*, pp. V313–V316, IEEE, San Antonio, Texas (2007).
11. M. Semsarzadeh et al., "Complexity modeling of the motion compensation process of the H.264/AVC coding standard," in *Proc. Int. Conf. on Multi. Expo (ICME'12)*, pp. 925–930, IEEE, Melbourne, VIC, Australia (2012).
12. S. Lee and C. Jay Kuo, "Complexity modeling of H.264/AVC CAVLC/UVLC entropy decoders," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS'08)*, pp. 1616–1619, IEEE, Seattle, Washington (2008).
13. Z. Ma, H. Hu, and Y. Wang, "On complexity modeling of H.264/AVC video decoding and its application for energy efficient decoding," *IEEE Trans. Multimedia* **13**(6), 1240–1255 (2011).
14. S. Park et al., "Quality-adaptive requantization for low-energy MPEG-4 video decoding in mobile devices," *IEEE Trans. Consum. Electron.* **51**(3), 999–1005 (2005).
15. H. Nam et al., "A complexity scalable H.264 decoder with downsizing capability for mobile devices," *IEEE Trans. Consum. Electron.* **56**(2), 1025–1033 (2010).

16. Y. C. Chao et al., "Efficient inverse transform architectures for multi-standard video coding applications," *IET Image Process.* **6**(6), 647–660 (2012).
17. Y. Wei, R. Zhang, and R. Lin, "A parallel computing algorithm for H.264/AVC decoder," in *Proc. Int. Conf. on Robot, Vision and Sig. Processing (RVSP'11)*, pp. 332–335, IEEE, Kaohsiung, Taiwan (2011).
18. T. Tsai, T. Fang, and Y. Pan, "A novel design of CAVLC decoder with low power and high throughput considerations," *IEEE Trans. Circ. Syst. Video Technol.* **21**(3), 311–319 (2011).
19. V. Sze and A.P. Chandrakasan, "A highly parallel and scalable CABAC decoder for next generation video coding," *IEEE J. Solid-State Circ.* **47**(1), 8–22 (2012).
20. S. Park, Y. Lee, and H. Shin, "An experimental analysis of the effect of the operating system on memory performance in embedded multimedia computing," in *Proc. ACM Int. Conf. on Embedded Software (EMSOFT'04)*, pp. 26–33, ACM, Pisa, Italy (2004).
21. T. Wiegand et al., "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circ. Syst. Video Technol.* **13**(7), 560–576 (2003).
22. H. Schwarz, D. Marpe, and T. Wiegand, "Overview of the scalable video coding extension of the H.264/AVC standard," *IEEE Trans. Circ. Syst. Video Technol.* **17**(9), 1103–1120 (2007).
23. P. List et al., "Adaptive deblocking filter," *IEEE Trans. Circ. Syst. Video Technol.* **13**(7), 614–619 (2003).
24. S. D. Kim et al., "A deblocking filter with two separate modes in block-based video coding," *IEEE Trans. Circ. Syst. Video Technol.* **9**(1), 156–160 (1999).
25. S. Tripathi and E. M. Piccinelli, "A scene change independent high quality constant bit rate control algorithm for MPEG4 simple profile transcoding," in *Proc. IEEE Int. Symp. on Broadband Multimedia Systems and Broadcasting 2008*, pp. 1–4, IEEE, Las Vegas, Nevada (2008).
26. M. J. Chen, M. C. Chu, and C. W. Pan, "Efficient motion-estimation algorithm for reduced frame-rate video transcoder," *IEEE Trans. Circ. Syst. Video Technol.* **12**(4), 269–275, (2002).
27. C. T. Hsu et al., "Arbitrary frame rate transcoding through temporal and spatial complexity," *IEEE Trans. Broadcast.* **55**(4), 767–775 (2009).
28. H. Shu and L. P. Chau, "Dynamic frame-skipping transcoding with motion information considered," *IET Image Process.* **1**(4), 335–342 (2007).
29. S. Liu, J. Kim, and C.-C. J. Kuo, "Nonlinear motion-compensated interpolation for low bit rate video," *Proc. SPIE* **4115**, 203–213 (2000).
30. J. Zhai et al., "A low complexity motion compensated frame interpolation method," in *Proc. IEEE Int. Symp. on Circuits and Systems (ISCAS'05)*, Vol. 5, pp. 4927–4930, IEEE, Kobe, Japan (2005).
31. Y. Wu, M. N. S. Swamy, and M. O. Ahmad, "Error concealment for motion-compensated interpolation," *IET Image Process.* **4**(3), 195–210 (2010).
32. H. Lee et al., "Computational complexity scalable scheme for power-aware H.264/AVC encoding," in *Proc. IEEE Int. Workshop on Multimedia Signal Process. (MMSP'09)*, pp. 1–6, IEEE, Rio De Janeiro, Brazil (2009).



**Hoyoung Lee** received the BS degree in electronics electrical engineering from Sungkyunkwan University, Suwon, Korea in 2007. He is currently a PhD candidate in the Digital Media Laboratory at Sungkyunkwan University. His research interests include video compression, resource-aware video coding and mobile multimedia framework.



**Younghyeon Park** received the BS degree in electronics electrical engineering from Sungkyunkwan University, Suwon, Korea in 2011. He is currently a PhD candidate in the Digital Media Laboratory at Sungkyunkwan University. His research interests include video compression and compressed sensing.



**Byeungwoo Jeon** received his BS degree in 1985 and an MS degree in 1987 from the Department of Electronics Engineering, Seoul National University, Seoul, Korea. He received his PhD degree in 1992 from the School of Electrical Engineering at Purdue University, Indiana, United States. From 1993 to 1997 he was in the Signal Processing Laboratory at Samsung Electronics in Korea, where he worked on video compression algorithms, designing digital broadcasting satellite receivers, and other MPEG-related research for multimedia applications. Since September 1997, he has been with the faculty of the School of Information and Communication Engineering, Sungkyunkwan University, Korea, where he is currently a professor. His research interests include multimedia signal processing, video compression, statistical pattern recognition, and remote sensing.