SPIHT Algorithm With Adaptive Selection of Compression Ratio Depending on DWT Coefficients

Hyun Kim, *Member, IEEE*, Albert No, *Member, IEEE*, and Hyuk-Jae Lee, *Member, IEEE*

***Abstract*—In mobile multimedia devices, the frame memory compression (FMC) technique by embedded compression (EC) is becoming an increasingly important video-processing method for reducing the external data bandwidth requirement, which, in turn, results in power savings. Among various EC schemes, the combination of discrete wavelet transform (DWT) and set partitioning in hierarchical trees (SPIHT) is widely used for FMC because it achieves high compression efficiency with low computational complexity. However, there is room for improvement in the conventional DWT and SPIHT algorithm because it compresses all blocks with the same compression ratio without taking into account the correlation between DWT coefficients and the SPIHT algorithm. This study proposes a novel one-dimensional (1-D) DWT and SPIHT algorithm, which enhances the quality of the compressed video by internally applying an adaptive compression ratio for the SPIHT algorithm based on DWT coefficients while keeping the same bit-stream size. The block complexity is predicted from the distribution of DWT coefficients. Then, simple blocks are aggressively compressed with a low compression ratio, while the complex blocks are compressed with a high ratio. Furthermore, to achieve the best video quality, each compression ratio is decided by an optimization technique based on mathematical formulation. Precisely, the logarithm of mean squared error by the SPIHT algorithm is assumed to be linearly correlated with the logarithm of processed DWT coefficients. Experimental results are provided that support the aforementioned model. Compared to the conventional 1-D DWT and SPIHT algorithm, the proposed scheme remarkably improves the video quality by an average of 2.23 dB in peak signal-to-noise ratio when the target compression ratio for the SPIHT algorithm is 5/16.**

***Index Terms*—Adaptive compression ratio, discrete wavelet transform, embedded compression, frame memory compression, set partitioning in hierarchical trees, optimization, video compression.**

Manuscript received October 25, 2017; revised February 6, 2018 and April 3, 2018; accepted April 15, 2018. Date of publication May 2, 2018; date of current version November 15, 2018. This work was supported in part by "The Project of Industrial Technology Innovation" through the Ministry of Trade, Industry and Energy under Grant 10082585,2017, and in part by the National Research Foundation of Korea grant funded by the Korea Government (MSIT) under Grant NRF-2017R1C1B5018298. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Zixiang Xiong. *(Corresponding author: Hyuk-Jae Lee.)*

H. Kim is with the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul 01811, Korea (e-mail: hyunkim@seoultech.ac.kr).

A. No is with the Department of Electronic and Electrical Engineering, Hongik University, Seoul 04066, Korea (e-mail: albertno@hongik.ac.kr).

H.-J. Lee is with the Inter-university Semiconductor Research Center, Department of Electrical and Computer Engineering, Seoul National University, Seoul 08226, Korea (e-mail: hyuk_jae_lee@capp.snu.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2018.2832604

I. Introduction

THE demand for portable multimedia devices that support video recording or video processing, such as smart phones and video cameras, is increasing. In order to reduce the memory consumed for video storage, the video recording system must contain video compression techniques. However, use of these techniques results in high computational complexity and external memory bandwidth requirement, causing significant power consumption. Video processing also involves a large number of computations and an increase in external memory traffic due to the usage of high-complexity vision algorithms such as deep neural networks. While many of these applications operate on battery power [1], [2], power management for video recording and video processing becomes a critical issue. Especially, as the resolution of video frames increases, the size of raw video data and the corresponding memory bandwidth increases as well. Thus, a frame memory compression (FMC) method based on embedded compression (EC) is becoming increasingly important for reducing the external data bandwidth requirement in video recording and video processing systems [3]–[8]. Note that the amount of the power consumed as a result of data transfer to external memory is very significant [9], although the internal power consumption can be greatly reduced by optimizing the operating conditions [10].

There are various EC schemes that have been used in recent years, and it is very important to select the appropriate schemes according to their use. Several EC schemes have been proposed for video compression and display systems. Transform-based schemes [11]–[13] take advantage of the high compression efficiency of image transform techniques such as discrete cosine transform [14] and discrete wavelet transform (DWT) [15], but they result in a considerable amount of computation. There are non-transform-based schemes such as differential pulse code modulation-variable length coding [6] and block truncation coding [16]. These schemes can be implemented with very simple computation, but the compression efficiency is relatively low and the bit-stream length of these schemes is variable. Therefore, this paper focused on transformed-based schemes to target higher compression efficiency.

In case of the FMC for video recording and video processing, EC performs the real-time compression of image/video frames before storing them into the external DRAM. After reading the stored data from external DRAM, the EC performs the corresponding decompression. In this scenario, EC for the FMC should be carefully selected for the following reasons: First, in

order to ensure the generality that can be used for all algorithms, the design of the EC module should be independent of the main video compression module in the video recording system and the video computer vision module in the video processing system. Second, the addition of the FMC using EC should have minimal impact on the main operation in video recording and video processing. In other words, the operation delay increased by the FMC should be minimized. Third, the trade-off between hardware resource and compression efficiency should be taken into account. For example, lossless compression [17], [18] is not appropriate for the FMC because of high memory bandwidth, large memory size, and high power consumption. Furthermore, it is difficult to precisely meet the target bit length (TBL) due to its variable bit-stream length. To achieve the maximum memory bandwidth reduction with acceptable video quality, lossy compression which achieves significant power saving with negligible quality loss is much more suitable for the FMC [8], [11], [13]. Fourth, even if lossy compression schemes are used with good compression efficiency and minimal hardware resources, the complexity of the compression algorithm for the FMC needs to be much lower than that of the video coding standards such as H.264 or High Efficiency Video Coding (HEVC) to avoid a significant increase in hardware cost. This requirement prohibits an aggressive compression, and therefore the target compression ratio (CR) for the FMC is relatively high, ranging from 10% to 50% [11].

In order to meet the aforementioned requirements for the FMC encoder and decoder, it is suitable that one-dimensional (1-D) three-level synthesis lifting DWT coefficients are compressed by a set partitioning in hierarchical trees (SPIHT) coding that achieves a high compression efficiency with low computational complexity [11], [19]–[21]. There are three main reasons why the combination of the 1-D DWT and SPIHT is selected for the FMC. First, it generates a bit-stream with the fixed target CR. This property enables a simple address calculation for the compressed data and thus minimizes the additional resource requirement for the address calculation. Second, the 1-D structure can support the raster-scan processing order so that it minimizes the operation delay by the FMC. Furthermore, it requires fewer hardware resources and less power compared to two-dimension (2-D) operations such as JPEG2000 [15], which requires the internal memory with a size of coding block height $\times$ frame width. Therefore, it can minimize the power consumption by additional FMC modules. Third, the combination of the 1-D DWT and SPIHT can support various coding granularities and CRs. This property facilitates the adaptive application of the appropriate CR to each compression unit according to its characteristic. The existing DWT and SPIHT algorithms [11] with these advantages show good compression efficiency, but the correlation between DWT coefficients and the SPIHT algorithm is not considered.

To compensate for this drawback, this study focuses on the correlation between DWT coefficients and characteristics of the SPIHT algorithm. The key idea is that for relatively simple coding blocks, DWT coefficients are well integrated into the low-pass band, while the relatively complex coding blocks still have a large number of DWT coefficients in the high-pass band [22]. Since the SPIHT algorithm compresses DWT coefficients only up to the TBL in the bit-plane domain, the degree of quality degradation varies according to the distribution of DWT coefficients even at the same CR. Therefore, the video quality can be improved by internally applying an adaptive CR according to DWT coefficients. Based on this observation, this study proposes a novel scheme that significantly enhances the video quality while maintaining the final bit-stream size. The proposed scheme aggressively compresses coding blocks with well-integrated DWT coefficients and passively compresses coding blocks with distributed DWT coefficients in order to enhance the video quality. Furthermore, each adaptive CR for the coding blocks is allocated by optimization techniques based on mathematical modeling between the video quality and DWT coefficients. Precisely, linear relation between quality degradation by the SPIHT algorithm and processed DWT coefficients is assumed, which can be justified from test video sequences. Experimental results show that compared to conventional 1-D DWT and SPIHT schemes [11], the proposed scheme significantly improves the Peak Signal to Noise Ratio (PSNR) by an average of 2.23 dB in the case of 5/16 target CR with only a negligible degree of increase in complexity.

The remainder of this paper is organized as follows. In Section II, an extensive background for the DWT coefficients and the SPIHT algorithm is explained. Section III describes the proposed approach for adaptive application of the compression depending on the results of DWT coefficients. In Section IV, the formulation and optimization schemes for the optimal CRs are explained. Experimental results are shown in Section V and finally, the conclusions are presented in Section VI.

II. Previous Works

This section briefly introduces high-throughput 1-D DWT and SPIHT algorithms [11] on which this study is based. Especially, the data structure of each algorithm and the relation between DWT coefficients and characteristics of the SPIHT algorithm are described in detail. The discussion about previous works shows the challenges associated with improving the performance of the 1-D DWT and SPIHT algorithms. The concepts introduced in this section are used in the remaining sections of this paper.

Basically, the SPIHT algorithm receives the DWT coefficients as input. The DWT coefficients are organized as a binary tree structure on which significance tests are performed [15]. For this study, the 1-D 3-level synthesis lifting DWT with integer Le Gall 5/3 filter [20] is selected from among various DWT schemes. In Fig. 1(a) and (b), the binary tree structure of 1×64 block size (pixels) when the DWT decomposition level is three is depicted. Each coefficient represents a set of coefficients for one byte (i.e., eight bits). In this binary tree, if the current node i is not at the lowest level, two children of coefficient c_i are defined as the two coefficients c_{2i} and c_{2i+1}. If the node $2i$ is also not at the lowest level, the children of c_{2i} are defined as c_{4i} and c_{4i+1} in succession. Therefore, in this case, c_{4i}, c_{4i+1}, c_{4i+2}, and c_{4i+3} are also descendants of coefficient c_i. This tree structure depends on the DWT decomposition level. Fig. 1(b) shows the tree structure more clearly than Fig. 1(a). The coefficient c_0 is

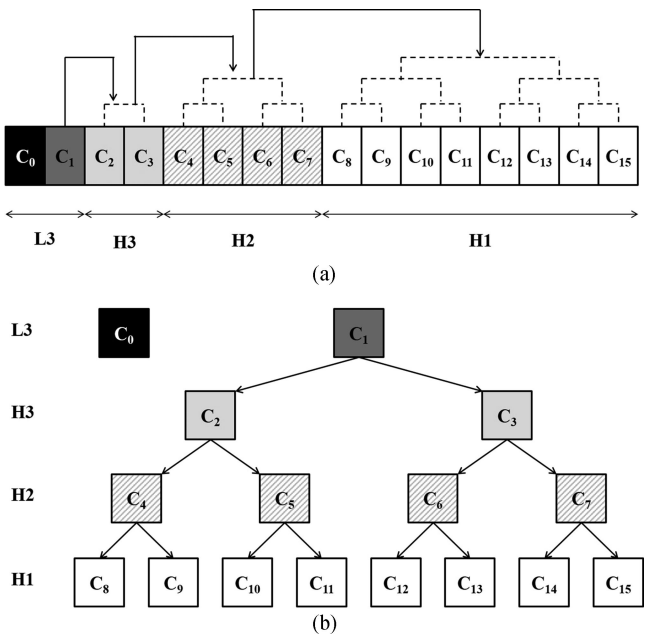


Fig. 1. Correlation between 1-D DWT coefficients when DWT decomposition level is three. (a) Array structure. (b) Corresponding binary tree structure.

the 0th low-pass band coefficient and consequently, it contains the most important information. The coefficient c_1 is the 1st low-pass band coefficient with decomposition level three and it becomes a root node of the binary tree structure. Both c_0 and c_1 are classified into the L3 level. Coefficients c_2 and c_3 in the H3 level are the ancestor nodes of high-pass band coefficients with decomposition level three. The relationship between the parent coefficients and all other offspring coefficients (i.e., from c_4 to c_{15}) is represented by arrows in Fig. 1(a) and (b).

The SPIHT is a representative wavelet transform-based algorithm that encodes in descending order starting from the upper bit-plane to the lower bit-plane in the bit-plane level [19]. Therefore, the more significant bits are coded first according to the results of DWT coefficients. In general, for processing the bit-plane operation, the SPIHT performs the significance test on sets of DWT coefficients and uses three data structures: the list of insignificant sets, the list of insignificant pixels, and the list of significant pixels, depending on the results of the significance test. Then, each structure is processed by three passes: the insignificant set pass (ISP), the insignificant pixel pass (IPP), and the refinement pass (RP), respectively. Among various SPIHT algorithms, this study focuses on the 1-D block-based pass-parallel SPIHT (BPS) algorithm proposed in [11]. It offers fast processing times for both the encoder and the decoder by reorganizing the existing passes of the SPIHT into the RP, the sorting pass (SP), and the first refinement pass (FRP) as shown in Fig. 2(a). In detail, the RP and the IPP are combined into one pass while the ISP is divided into two passes (i.e., the SP and the FRP).

There are two kinds of dependencies that prevent individual paths from being processed at the same time, as indicated by the dashed arrows in Fig. 2(b). The first dependence exists between the SP and the FRP. It can be avoided by delaying the FRP

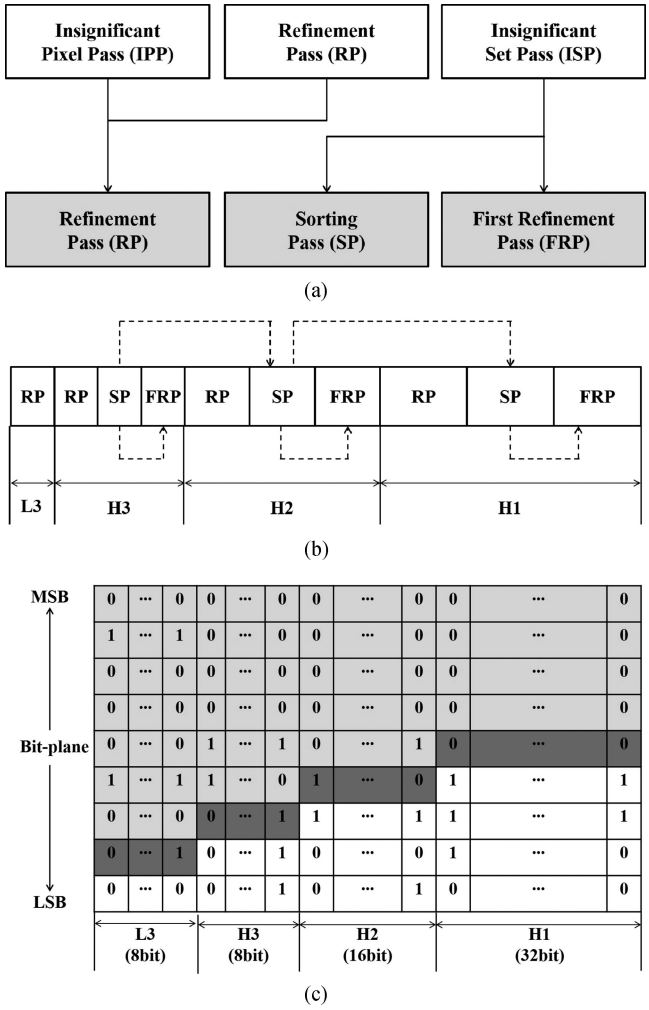


Fig. 2. Previous survey for the SPIHT algorithm. (a) Pass reorganization by BPS algorithm. (b) Structure of the 1-D SPIHT algorithm with DWT decomposition level three based on coding passes. (c) Example of the processing order in the 1-D SPIHT algorithm.

execution by one cycle compared to the SP execution. The second dependence exists between the SP of the higher-pass band and that of the lower-pass band and it can be avoided by delaying the SP execution phase of the lower-pass band by one cycle compared to that of the higher-pass band. As a result, the BPS algorithm enables three passes to be processed in parallel and pipelined manners. Therefore, the throughputs of the encoder and the decoder are relatively large compared to the existing methods.

Fig. 2(c) shows a processing order of the 1-D BPS algorithm. During descending order from the most significant bit (MSB) to the least significant bit, the encoded bit length is compared to the TBL each time every encoded bit is generated in the current bit-plane which is marked with dark gray color. If the encoded bit length has reached the TBL, the algorithm stops running and the remaining data in the lower bit plane with white color blocks are discarded. The great advantage of the SPIHT algorithm is a considerable compression efficiency that works by encoding important data first while maintaining a very simple and fast hardware structure. In fact, it demonstrates a performance

TABLE I
EXPERIMENTAL CONDITIONS

Test Video Sequences HD (1280 × 720)	Aspen	Blue_sky	Controlled_burn
	Factory	In_to_tree	Pedestrian_area
	Snow_mnt	Sunflower	Tractor
Encoding Configuration	Compression ratios		From 3/16 to 9/16
	Processing Unit		1 × 64

similar to other wavelet transform-based algorithms: embedded zerotree wavelet algorithm [23] and embedded block coding with optimized truncation algorithm [24], despite the much simpler hardware structure without arithmetic coding [25].

However, the compression efficiency of the 1-D DWT and SPIHT is relatively low because 1-D DWT that cannot make use of the redundancy in the vertical direction lowers the compression performance compared to that in the 2-D SPIHT [11]. Therefore, the compression efficiency of the 1-D SPIHT needs to be improved while maintaining its small memory size and short latency. In order to enhance the compression efficiency of the 1-D DWT and SPIHT algorithm, an adaptive CR scheme for the SPIHT is proposed in the next section which works by analyzing the correlation between the DWT coefficients and the characteristics of the SPIHT algorithm.

III. ADAPTIVE SPIHT BASED ON DWT COEFFICIENTS

A. Analysis of the Correlation between DWT and SPIHT

The objective of this study is to obtain the highest quality at a fixed CR for each frame. In this study, as a measure of the video quality, the most common method, PSNR, is used, which can be equated as follows [26]:

$$PSNR = 20 \cdot \log_{10}\left(\frac{255}{\sqrt{MSE}}\right). \quad (1)$$

MSE is the mean squared error between the original videos (Orig) and the damaged videos (Dam) whose video quality is degraded by the compression. In a block with $m \times n$ pixels, MSE is expressed as follows [26]:

$$MSE = \frac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}[Orig(i,j) - Dam(i,j)]^2. \quad (2)$$

The key observation here is that the amount of degradation in MSE is quite dependent on DWT coefficients. The DWT coefficients vary according to the complexity of the coding blocks [22] and the SPIHT algorithm compresses these DWT coefficients at the bit-plane level with priority to the low-pass band coefficients only up to the TBL. Therefore, for simple blocks where the most DWT coefficients are located in the low-pass band, SPIHT compresses well and thus, results in small MSE even with lower CR. On the other hand, for complex blocks with scattered DWT coefficients, the SPIHT algorithm results in a large MSE, even with a higher CR.

To prove this claim, several experiments are conducted. For these experiments, Table I summarizes the experimental conditions, depicting nine HD video sequences [27] in the top row

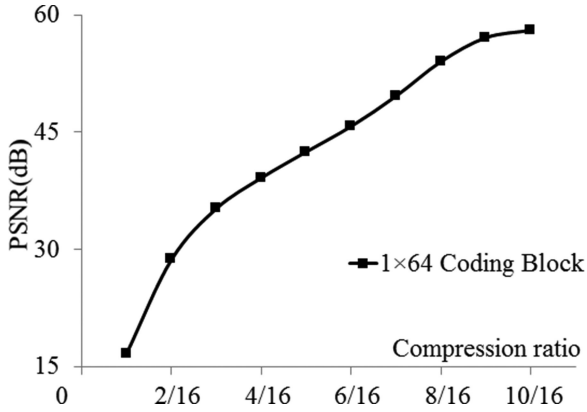


Fig. 3. Average PSNR of the 1-D DWT and SPIHT algorithm according to various compression ratios.

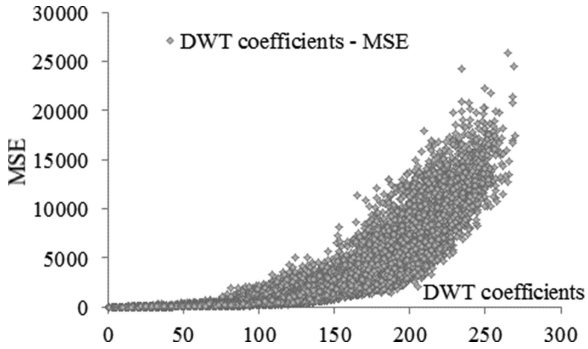


Fig. 4. Correlation between DWT coefficients and MSE.

and the encoding configurations of the 1-D DWT and SPIHT algorithms in the bottom row. Based on the Fig. 3 which shows an average PSNR of the 1-D DWT and SPIHT algorithms, the CR is limited from 3/16 to 9/16 at the processing unit of 1×64 because the PSNR significantly degrades at a CR lower than 3/16 and converges at a CR higher than 9/16. Fig. 4 shows a correlation between DWT coefficients in the horizontal axis and MSE in the vertical axis when the CR is 4/16. The correlations of 129,600 $(1,280 \times 720/64) \times 9$ sequences) coding blocks are marked by each point. In order to express DWT coefficients of each coding block as a numerical value, a new variable is obtained by using the values of 56 pixels belonging to the H3 level, the H2 level, and the H1 level (i.e., 8 pixels in the H3 level $+16$ pixels in the H2 level $+32$ pixels in the H1 level):

$$Cost_{DWT} = \sum_{p=H1}^{H3} TRUNC(\log_2 p). \quad (3)$$

The $TRUNC(\cdot)$ function is an operation that discards the decimal point. Since $TRUNC(\log_2 p)$ means the position of the MSB of each pixel in hardware design, expressing the DWT coefficients by (3) not only represents DWT coefficients accurately, but also makes hardware implementation very easy. The L3 level is not included in (3) for $Cost_{DWT}$ because the position of the MSB in the L3 level is always large, regardless of the complexity of the coding block [22]. It should be noted that $Cost_{DWT}$ for the complex coding block becomes relatively

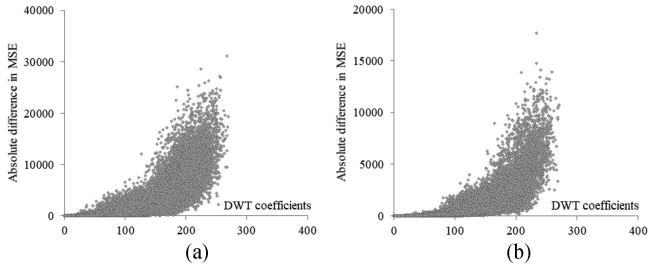


Fig. 5. Correlation between DWT coefficients and the absolute difference in MSE when CR varies. (a) CR $4/16 \rightarrow 3/16$. (b) CR $4/16 \rightarrow 5/16$.

large because the complex coding block has large values from the H3 level to the H1 level compared to the uncomplicated coding block. The experimental results in Fig. 4 clearly show that MSE increases as the DWT coefficients (i.e., $Cost_{DWT}$) increase. This motivates the adaptive CR scheme which applies higher CRs to blocks with high $Cost_{DWT}$ values.

To see the effectiveness of the adaptive CR scheme, Fig. 5 shows the correlation between the DWT coefficients and the absolute amount of difference in MSE when the CR is modified in the corresponding block. In each figure, similar to Fig. 4, 129,600 coding blocks are indicated by points. The x-axis represents the DWT coefficient of each block, and the y-axis is the absolute amount of MSE difference of each block when CR changes. In Fig. 5(a), the CR is reduced from 4/16 to 3/16, and it can be seen that MSE varies more in blocks with high DWT coefficients. Similarly, in Fig. 5(b), CR increases from 4/16 to 5/16, and MSE of high DWT coefficient blocks varies more. This implies that even with the same change of the CR, MSE of the coding block with high $Cost_{DWT}$ varies much more rapidly. In summary, a relatively complex coding block with a high $Cost_{DWT}$ has a larger MSE and is more sensitive to the change of the CR, while a relatively simple block with low $Cost_{DWT}$ has a smaller MSE and is less sensitive to the change in the CR.

The cause of this tendency can be found in the operation of the SPIHT algorithm. As shown in Fig. 2, in each level, the SPIHT algorithm maintains the SP execution from the highest bit-plane until the current bit-plane reaches the MSB (i.e., when "1" appears for the first time), and then operates the RP execution through the FRP execution. It should be noted that the FRP and RP executions generate a much larger amount of output streams compared to the SP execution [11]. Therefore, the coding block with large DWT coefficients (i.e., the position of the MSB is formed in the high bit-plane) starts the RP execution much earlier, and the size of the bit-stream increases rapidly and eventually reaches the TBL at the higher bit-plane. When the TBL is reached, all pixel values in the lower bit-plane are discarded. Hence, the performance degradation increases significantly as the SPIHT algorithm terminates at the higher bit-plane.

To clarify the correlation between DWT coefficients and the SPIHT algorithm, average values of $Cost_{DWT}$ based on the termination point of the SPIHT algorithm are shown in Table II. The results when the CR of the SPIHT algorithm is 4/16 are

TABLE II
AVERAGE OF $Cost_{DWT}$ ACCORDING TO TERMINATION POINT OF THE SPIHT ALGORITHM WHEN THE COMPRESSION RATIO IS 4/16

Termination bit-plane	7	6	5	4	3	2
$\boldsymbol{Cost_{DWT}}$	234	169	110	68	37	6

tested with nine video sequences given in Table I. In Table II, the first and second rows represent the bit-plane in which the SPIHT algorithm terminates and the average values of $Cost_{DWT}$ corresponding to each bit-plane, respectively. The original bit-plane for each pixel is from 7 to 0 (i.e., 8 bits), but as shown in Fig. 2(c), the range of the bit-plane varies from 10 to 0 because it has a pipelined manner from the L3 level to the H1 level for achieving the high throughput and performance. In this experiment with the CR of 4/16, the termination point ranges from 7 to 2. Experimental results show that the coding blocks at which the SPIHT algorithm terminates at the relatively high bit-plane have a relatively high $Cost_{DWT}$. From these results, it can be seen that there is a clear correlation between DWT coefficients and the video quality based on the manner of operation of the SPIHT algorithm.

B. Adaptive Compression Ratio of the SPIHT Algorithm Based on DWT Coefficients

As mentioned in Section III-A, the complexity of the coding blocks affects the video quality. Hence, it is possible to improve the video quality by applying the adaptive CRs to each coding block within the frame based on DWT coefficients. However, even if various CRs are used in a frame, the final TBL should be maintained to take advantage of the fixed CR. Therefore, while maintaining the TBL, coding blocks with low $Cost_{DWT}$ (i.e., well-integrated DWT coefficients) should be aggressively compressed with the low CR and those with high $Cost_{DWT}$ (i.e., distributed DWT coefficients) should be passively compressed with the high CR for enhancing the video quality. In other words, the key idea of this scheme is that the space margins that occur due to the reduction in the CR in the simple coding blocks are utilized for compressing more complex coding blocks. It should be noted that even with the same amount of change in the CR, the performance gain obtained by increasing the CR in the complex coding blocks is relatively large compared to the performance degradation caused by reducing the CR in the simple coding blocks, as shown in Fig. 5.

For using various CRs in a frame, the CR information must be stored together in the encoding process because the CR of the corresponding coding block must be known in the decoding process. In this study, 3 bits are allocated to the output stream for utilizing 7 CRs in 1/16 unit from 3/16 to 9/16 to store the information of the CR, as shown in Fig. 6. As a result, the space for storing the SPIHT results in each CR is reduced by 3 bits. In YUV422 format, the size of uncompressed raw data in the 1×64 coding block is $1024\,(= 1 \times 64 \times 16)$ bits. This bit allocation for the CR ensures identical operation of the encoder and the decoder even if various CRs are used. Such bit allocation causes some loss of compression efficiency, but the efficiency

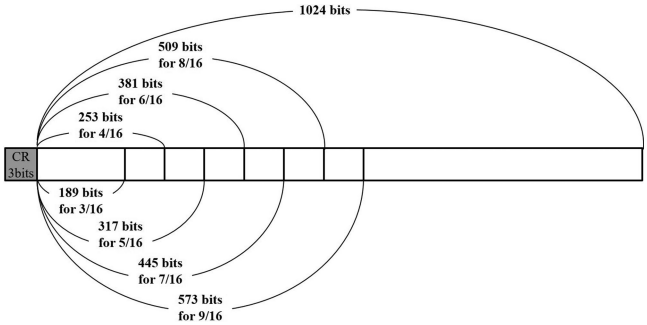


Fig. 6. Structure of the output stream for the 1×64 SPIHT coding block according to various compression ratios.

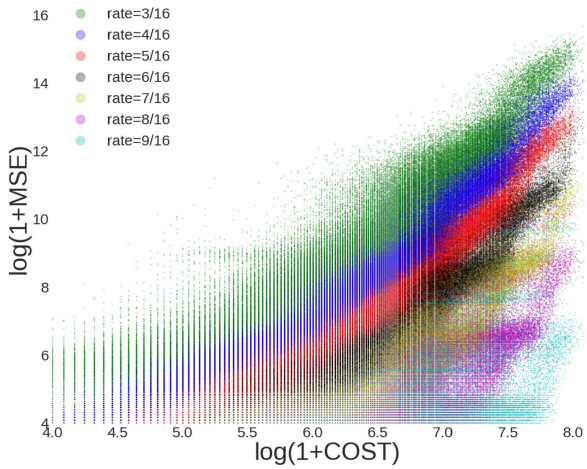


Fig. 7. Linear correlation between DWT coefficients and MSE.

loss is negligible because the proportion of 3 bits to total bitstream is very small.

Obviously, it is better to allocate a higher CR to the more complex coding block, but identifying the suitable CRs for all coding blocks is a challenging problem. This problem can be stated as follows: "Within a given final TBL, assign each of the CRs to each coding block so that the video quality is improved as much as possible." Since the maximum improvement of the video quality ultimately translates into minimizing the sum of the MSEs of all coding blocks, the goal of this study can be expressed as follows:

$$\min \sum_{i=0}^{N-1} MSE_i \tag{4}$$

where MSE_i denotes the MSE of the i-th coding block and N (=14,400 for the HD resolution) denotes the number of coding blocks in a frame. There is a constraint to maintain the final TBL in achieving this goal and it can be expressed as follows:

$$\frac{1}{N} \sum_{i=0}^{N-1} CR_i = CR_{Fixed} \tag{5}$$

where CR_i denotes the CR of the i-th coding block and CR_{Fixed} denotes the CR given to the encoder to satisfy the final TBL. In other words, if the adaptive CRs are not used, all coding blocks are compressed with CR_{Fixed}. In addition, CR_i has the following range limitation as mentioned in the previous subsection.

$$\frac{3}{16} \le CR_i \le \frac{9}{16} \quad \text{(discrete values with } \tfrac{1}{16} \text{ unit)}. \tag{6}$$

In order to achieve the goal in (4) within the constraints of (5) and (6), mathematical modeling and optimization schemes are proposed in Section IV.

IV. Optimization for the Best Performance

In this section, a novel method for allocating optimized CRs based on $Cost_{DWT}$ is proposed. To state the problem in (4) more formally, a mathematical model that describes the correlation between $Cost_{DWT}$ and MSE is necessary. For simplicity, let $Cost_i$ be $Cost_{DWT}$ of the i-th coding block, and $MSE_i(CR_i)$ be MSE of the i-th block when the CR of the i-th coding block is CR_i. Fig. 7 shows the correlation between $\log_2(1 + Cost_i)$ in the horizontal axis and $\log_2(1 + MSE_i(CR_i))$ in the vertical axis for the various CRs of nine video sequences given in Table I. In order to prevent the terms in log from becoming zero, 1 is added to both MSE_i and $Cost_i$. In this paper, the linear model between $\log_2(1 + Cost_i)$ and $\log_2(1 + MSE_i(CR_i))$ is assumed, which clearly can be seen in Fig. 7. Since different CR yields a different linear relation, $\log_2(1 + MSE_i(CR_i))$ should be a function of CR as well. The proposed model further assumes that $\log_2(1 + MSE_i(CR_i))$ is linearly related with CR_i as well. This implies that the gaps between linear functions of different rates (which are plotted with different colors in Fig. 7) are the same. More precisely, the proposed model can be expressed as follows:

$$\log_2(1 + MSE_i(CR_i)) \approx a \log_2(1 + Cost_i) - bCR_i + c \tag{7}$$

where a is the slope, and b represents the constant gap between linear functions.

Note that the linearity assumption between $\log_2(1 + MSE_i)$ and $\log_2(1 + Cost_i)$ is not crucial. For example, polynomial or exponential functions can be used. However, the linear model is chosen in this work because it is the simplest model that describes the data well. Since it has a few parameters so that it is immune to overfitting problem, which is the common problem of complex models. On the other hand, the linearity assumption between $\log_2(1 + MSE_i)$ and CR is necessary to get analytic solution of optimum rate allocation, which we will discuss later in this section.

Using (7), MSE_i can be estimated based on CR_i and $Cost_i$. Constants a, b, and c should minimize the deviations from the plotted points depicted in Fig. 7 and the linear representation in (7) for all discrete CRs in the range of (6). Therefore, they minimize the following linear regression problem:

$$\min \sum_{i=0}^{N-1} \sum_{\frac{3}{16} \le CR \le \frac{9}{16}} [\log_2(1 + MSE_i(CR_i)) - (a \log_2(1 + Cost_i) - bCR_i + c)]^2 . \tag{8}$$

By solving (8), coefficients a, b, and c are obtained as 2.1, 20.07, and -0.229, respectively. These values can be changed slightly based on the test sequences. However, the essence of this method remains the same even with different constants.

With these constants, the original problem of allocating the optimized CRs to each coding block can be resolved. First, the linear equation in (7) can be expressed as follows:

$$MSE_i(CR_i) \approx \exp_2\left(a\log_2(1+Cost_i) - bCR_i + c\right) - 1 \tag{9}$$

where $\exp_2(x) = 2^x$. The original optimization problem in (4) requires a solution of the difficult combinatorial problem. Instead, by using (9), it can be relaxed into the following form which provides a simple analytic solution.

$$\min \frac{1}{N}\sum_{i=0}^{N-1} \exp_2\left(a\log_2(1+Cost_i) - bCR_i + c\right). \tag{10}$$

It should be noted that the last term on the right side in (9), i.e., -1, is dropped in (10) since it does not affect the method for achieving the optimized CRs. Then,

$$\begin{aligned}
&\frac{1}{N}\sum_{i=0}^{N-1} \exp_2\left(a\log_2(1+Cost_i) - bCR_i + c\right) \\
&\quad \geq \exp_2\left(\frac{1}{N}\sum_{i=0}^{N-1} a\log_2(1+Cost_i) - bCR_i + c\right) \\
&\quad = \exp_2\left(\frac{1}{N}\sum_{i=0}^{N-1} a\log_2(1+Cost_i) - bCR_{Fixed} + c\right)
\end{aligned} \tag{11}$$

where the first inequality is based on Jensen's inequality. The last equality is derived from (5), based on the overall rate constraint, which dictates that the average rate should be equal to the overall rate. Note that the right hand side of the last equality is not a function of CR_i's since b is constant. An equality condition of Jensen's inequality is

$$a\log_2(1+Cost_i) - bCR_i + c = K \tag{12}$$

for some constant K for all i. The above equation can be solved by averaging them all:

$$a\frac{1}{N}\sum_{i=0}^{N-1}\log_2(1+Cost_i) - b\frac{1}{N}\sum_{i=0}^{N-1} CR_i + c = K. \tag{13}$$

For simplicity, let S be the average of $\log_2(1+Cost_i)$:

$$S = \frac{1}{N}\sum_{i=0}^{N-1}\log_2(1+Cost_i). \tag{14}$$

Then, (13) implies that

$$aS - bCR_{Fixed} + c = K. \tag{15}$$

Finally, from (12) and (15), the rate allocation should be

$$CR_i = CR_{Fixed} + \frac{a}{b}(\log_2(1+Cost_i) - S). \tag{16}$$

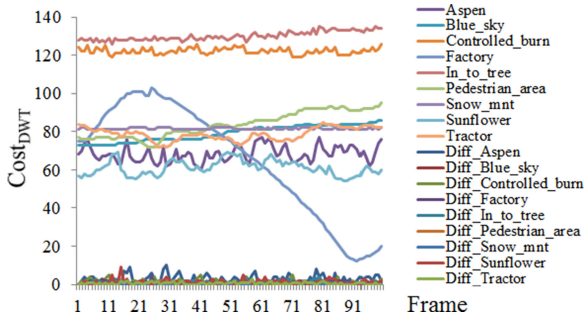


Fig. 8. Similarity of DWT coefficients between consecutive frames.

These results establish the proposition that assigns a higher CR value to the coding block with higher DWT coefficients. By using (14) and (16), the optimized CR for each coding block is determined based on the $Cost_i$ value derived from DWT coefficients. However, only the discrete CRs between 3/16 and 9/16 are allowed to be used as shown in (6). Hence, it is necessary to quantize CR_i to the nearest value. In this study, the suitable discrete CRs are allocated as follows:

$$CR_i = \left[\left(CR_{Fixed} + \frac{a}{b}(\log_2(1+Cost_i) - S)\right) \times 16\right] \Big/ 16 \tag{17}$$

where $[x]$ is the rounding function. If the optimized result from (17) is out of range in (6) (i.e., $\frac{3}{16} > CR_i$ or $\frac{9}{16} < CR_i$), the nearest values (i.e., 3/16 or 9/16) are selected for those CRs beyond the boundary.

The pending issue is a method of obtaining the value of S, which needs all $Cost_i$ values within a given frame. This implies that the DWT coefficients of all coding blocks must be computed, but this causes a significant delay that hampers real-time operation. Therefore, instead of this process, the value of S is estimated using the previous frame. It assumes that S does not vary significantly from the previous frame to the current frame. Fig. 8 supports this assumption by showing the trends of the average of $Cost_{DWT}$ ($Cost_{av}$) over nine test sequences in Table I. The number of frames in each sequence is 100. It should be noted that S is determined based on the average of $Cost_i$ as shown in (14). The horizontal axis represents the flow of the frame and the vertical axis represents the corresponding change in $Cost_{av}$. The results show that the $Cost_{av}$ values of neighboring frames are similar in most test sequences. In the "Factory" sequence, the change in $Cost_{av}$ is significant because of the rapid change of the scene. However, because there is no abrupt slope change across the previous and current frames, the change between consecutive frames is negligible. To support this assumption, the difference between the previous and current frames is plotted on the graph. From these results, the average difference of $Cost_{av}$ between consecutive frames is found to be approximately 1.5% and the maximum difference between consecutive frames is found to be less than 10%, which shows the similarity between consecutive frames.

Next, it should be noted that coefficients in (7) are obtained from the data with a rate between 3/16 and 9/16. This is problematic when $Cost_i$ is very low, which results in the CR value

being much lower than 3/16 in (16). Because such blocks also use the CR of 3/16, this may cause the rate explosion (i.e., a significant increase in the overall rate). To compensate for this effect, for blocks with very low $Cost_i$ values, which ensures a CR lower than 3/16 in (16), $\log_2(1+Cost_i)$ value is slightly modified in the process of calculating S. Precisely, when (16) provides a CR below 3/16, the minimum CR (i.e., 3/16) is being used and the $\log_2(1+Cost_i)$ value for calculating S is modified to $\log_2(1+Cost_i')$, which satisfies the following equation.

$$\frac{3}{16} = CR_{Fixed} + \frac{a}{b}(\log_2(1+Cost_i') - S). \qquad (18)$$

Then, this modified $\log_2(1+Cost_i')$ is used to estimate S for the next frame. This technique allows an overestimation of S so that each coding block uses a lower CR, in turn preventing rate explosion.

However, even after using the techniques to compensate for the rate explosion, the condition in (5) may not be satisfied. To make sure that the rate constraint (5) is always satisfied, the encoder checks the number of available bits for the remaining blocks of the frame. If there are a sufficient number of available bits, the encoder assigns the highest CR (i.e., 9/16) to all remaining blocks. Precisely, if the following condition holds, the encoder assigns 9/16 to all the remaining blocks of the frame.

$$N \times CR_{Fixed} - \sum_{i=0}^{N-RN-1} CR_i = RN \times \frac{9}{16} \qquad (19)$$

where RN denotes the number of remaining coding blocks. On the other hand, if the number of available bits is low, the encoder assigns the lowest CR (i.e., 3/16) to all remaining blocks. Precisely, if the following condition holds, the encoder assigns 3/16 to all the remaining blocks of the frame.

$$N \times CR_{Fixed} - \sum_{i=0}^{N-RN-1} CR_i = RN \times \frac{3}{16}. \qquad (20)$$

By using the conditions in (19) and (20), the proposed method can always satisfy the condition in (5). It should be noted that the number of coding blocks whose CRs are determined by (19) and (20) is very small due to the similarity between consecutive frames as shown in Fig. 8.

V. Experimental Results

A. Performance Estimation of the Proposed System

In this subsection, the video quality achieved using the proposed adaptive CR scheme is presented and compared to the quality with conventional DWT and SPIHT algorithms for 1-D [11] and 2-D [13]. In addition, the hardware implementation results of the proposed scheme are also presented and compared with these previous schemes. For fair comparison, each scheme is implemented and simulated.

Table III shows a comparison of the PSNR values for the methods that use the three DWT and SPIHT algorithms based on the various CR values. Each value is the average of the results obtained by performing software simulations with nine test sequences, as shown in Table I. The number of frames in each sequence is 100. The values, represented by the bold font in this table, represent the best results in each list. The experimental results indicate that the overall video quality is significantly enhanced by the proposed scheme as compared to the previous work [11] on which the proposed scheme is based, regardless of the target CR. The maximum difference in PSNR between the proposed scheme and the conventional 1-D scheme [11] is approximately 2.23 dB at the target CR of 5/16. Furthermore, compared to the previous 2-D DWT and SPIHT scheme [13], which operates with an 8×8 processing unit, the proposed scheme achieves an even better video quality at low CRs from 4/16 to 6/16. It should be noted that, in general, 2-D compression schemes are more efficient than 1-D compression schemes because 2-D schemes take advantage of data correlations in both horizontal and vertical directions by using more hardware resources [11]. Nevertheless, at relatively low CRs, the proposed scheme enables 1-D schemes to achieve better performance with much less hardware resources than 2-D schemes.

TABLE III
Comparison of the PSNR According to Various Compression Ratios

PSNR (dB)	Compression Ratio				
	4/16	5/16	6/16	7/16	8/16
Proposed	**40.91**	**44.72**	**47.95**	51.39	55.14
1-D [11]	39.15	42.49	45.75	49.63	54.04
2-D [13]	40.14	43.65	47.39	**51.98**	**57.84**

To prove the effectiveness of the proposed method more directly, Table IV shows the average amount of difference in MSE (ΔMSE_{av}) per block in the above experiment. For example, when the proposed scheme with CR 4/16 is being used, the table tells that 25.9% of blocks have high $Cost_{DWT}$ so that they are passively compressed with the high CR, and the amount of MSE decrease is 4,632,740 (on average). It should be noted that the PSNR increases as the MSE decreases. On the other hand, 40.9% of blocks have low $Cost_{DWT}$ so that they are aggressively compressed with the low CR, and therefore the amount of MSE increase is 888,692 (on average). Since the blocks of middle $Cost_{DWT}$ have CR 4/16, MSE remains the same. It is clear that for all CRs from 4/16 to 8/16, the amount of MSE decrease in high $Cost_{DWT}$ blocks is much greater than the amount of MSE increase in low $Cost_{DWT}$ blocks. Thus, the above results experimentally prove that the overall MSE reduced by having the adaptive CRs, and the PSNR is remarkably improved.

Furthermore, in order to evaluate the efficiency of the proposed schemes by a comparison of the subjective video quality, Fig. 9(a), (b), (c), and (d) show still images of the reconstructed videos in the "Aspen" "Blue_sky", and "Controlled_burn" sequences for the uncompressed (as ground-truth), conventional 1-D, conventional 2-D scheme, and proposed method, respectively. The CR of 4/16 is used for each DWT and SPIHT algorithm. In order to increase the visibility, a part of the reconstructed video sequences is selected, including both the background part with a relatively low $Cost_{DWT}$ blocks and the object part with a relatively high $Cost_{DWT}$ blocks, and

TABLE IV
COMPARISON OF THE MSE AND PROPORTION ACCORDING TO VARIOUS COMPRESSION RATIOS AND $Cost_{DWT}$ VALUES

CR	4/16		5/16		6/16		7/16		8/16	
	ΔMSE_{av}	Proportion	ΔMSE_{av}	Proportion	ΔMSE_{av}	Proportion	ΔMSE_{av}	Proportion	ΔMSE_{av}	Proportion
High $Cost$	-4,632,740	25.9%	-2,668,938	30.2%	-1,205,924	31.7%	-499,255	33.4%	-149,751	38.9%
Middle $Cost$	0	33.5%	0	29.8%	0	28.6%	0	27.5%	0	23.4%
Low $Cost$	888,692	40.6%	409,608	40.0%	240,586	39.7%	134,054	39.1%	57,946	37.7%

Fig. 9. Still images of the reconstructed frames in the "Aspen," "Blue_sky," and "Controlled_burn" sequences at CR 4/16. (a) Uncompressed. (b) Conventional 1-D. (c) Conventional 2-D. (d) Proposed.

enlarged twice before capturing. In high $Cost_{DWT}$ blocks (i.e., objects), only Fig. 9(b) shows clear vertical lines between the coding blocks at the low CR of 4/16 due to differences in the values of pixels at the boundary of the coding block. However, in Fig. 9(b) by the proposed scheme, the video quality degradation is invisible compared to uncompressed images in Fig. 9(a) or 2-D reconstructed images in Fig. 9(c) even in high $Cost_{DWT}$ blocks because the proposed scheme passively compresses high $Cost_{DWT}$ blocks with the high CR. On the other hand, in low $Cost_{DWT}$ blocks (i.e., backgrounds), such line effect cannot be found in both Fig. 9(b) and (d). This is because even if the simple background part is compressed more aggressively in the propose scheme, the loss does not increase so much and thus, the effect on the video quality is negligible compared to uncompressed images in Fig. 9(a). This subjective video quality result proves the effectiveness of the proposed method with Tables III and IV.

Next, for the various DWT and SPIHT algorithms, the hardware implementation results for logic circuits are obtained with a post-layout simulation using the netlist synthesized by a Synopsys Design Compiler targeting a 0.13 μ m process technology and presented in Table V. Since the processing unit and operating frequency of each system are different, the normalized throughput and normalized logic size are presented in the

TABLE V
HARDWARE IMPLEMENTATION COMPARISON OF THE PROPOSED AND PREVIOUS SCHEMES

	Proposed		1-D [11]		2-D [13]	
Processing Unit	1×64	1×64	1×64	1×64	1×4	1×4
Frequency (MHz)	115	139	115	139	150	110
Throughput (Gbits/sec)	5.66	6.35	5.66	6.35	2.34	1.72
Norm. Throughput (Mbits/(s×MHz))	49.22	45.68	49.22	45.68	15.6	15.64
Logic (no.)	18699	27546	16621	25824	22996	21901
Normalized Logic (no.×sec×Hz/bits)	380	603	338	565	1474	1404
Int. Mem. (Kbits)	0	0	0	0	128.3	126.7

TABLE VI
COMPARISON OF THE BDPSNR ACCORDING TO VARIOUS FMC SCHEMES

BDPSNR (dB)	H.264	HEVC
Proposed FMC	**-0.316**	**-0.302**
1-D FMC [11]	-0.63	-0.559
2-D FMC [13]	-0.481	-0.401

TABLE VII
AMOUNT OF THE POWER GAINS ACCORDING TO VARIOUS FMC SCHEMES

Systems	Internal Gain (mW)	External Gain (mW)	Total Gain (mW)
Proposed 1-D FMC	-10.65	44.08	33.43
2-D FMC [13]	-29.42	44.08	14.66

sixth and eight rows, respectively. Comparing the results for the proposed scheme with those for the previous 1-D scheme [11], we can see that the proposed scheme can be implemented easily with only a slight increase in the logic size for the adaptive CR scheme while maintaining the throughput. Thus, the proposed scheme achieves much better video quality while maintaining a similar level of hardware resource requirement.

On the other hand, comparing the results for the proposed scheme with those for the previous 2-D scheme [13], we can see that the proposed scheme achieves much higher normalized throughput with much smaller normalized logic size. Furthermore, it should be noted that both 1-D schemes (i.e., the proposed system and [11]) do not require additional internal memory for the encoding and decoding process, whereas the internal memory is essential for the 2-D scheme to generate 2-D inputs and the size of this internal memory increases as the width of the input frame increases. In short, the proposed scheme achieves a video quality similar to the previous 2-D DWT and SPIHT scheme [13] with a much lower hardware resource requirement.

B. Performance Estimation as Frame Memory Compression

As described in the previous section, since the compression ratio of ECs such as the DWT and SPIHT module is not as outstanding as video coding standards, ECs have been used as the FMC for video coding standards [28], [29], LCD overdriving techniques [8], and deep neural networks [30] rather than being used for a single video compression method. In this section, the effectiveness of the proposed scheme which improves the performance of the DWT and SPIHT module is evaluated on the FMC for video coding standards as an actual application.

In H.264 video compression standard and HEVC standard, the reconstructed frame is stored in the external memory because it is used as the reference frame for the motion estimation in the next frame. In order to reduce the power consumed by the external memory, the proposed hardware platform includes the FMC encoder between the main video compression module and the external memory. When the reference frame is loaded from the external memory, the FMC decoder recovers the frame data and transfers the recovered data to the main video compression module. As shown in Table V, the proposed FMC encoder and decoder are implemented with very high throughput in order to facilitate real-time application when used with the main compression module.

In order to demonstrate the changes in video quality achieved by applying the various FMCs, Table VI presents the Bjontegaard Delta PSNR (BDPSNR) [31] of the H.264 and HEVC standards with the various FMCs. Each system is tested with the nine video sequences in Table I and the average BDPSNRs are presented. The first column denotes the type of FMC used. All FMCs operate with the CR of 6/16. The results show that the proposed FMC exhibits the smallest BDPSNR degradation regardless of the video coding standard. Compared to the conventional 1-D FMC [11], the proposed scheme compensates the maximum BDPSNR degradation up to approximately 50%. When the proposed scheme is used for FMC purposes, the quality enhancement by the proposed scheme compared to the previous schemes is smaller than the results listed in Table III because video coding standard also cause loss by quantization. Hence, the quality degradation by the FMC overlaps with the degradation by the video coding standard. Nevertheless, compared to the conventional 1-D FMC [11], the proposed scheme can significantly compensate for the video quality degradation caused by the FMC.

Table VII provides the estimate of power consumption for H.264 standards with the two FMCs: the proposed 1-D FMC and the previous 2-D FMC [13]. It should be noted that the H.264 standards with the proposed FMC as well as the previous FMC presented in [11] result in a similar power reduction effect because the hardware resources and operation manner are almost the same. The amount of power gain is calculated by dividing the internal[1] and external[2] power gains. Video coding standards with the FMC modules consume more internal power (i.e., negative power gain as shown in the second column) due to the requirement of additional modules for the FMC, but their external power consumption is much low (i.e., positive power gain as shown in the third column). As a result, both systems

[1]The internal power consumption is obtained by post-layout simulation with the same simulation environments for Table V.

[2]The number of memory accesses and the necessary memory size for the external power consumption are simulated by ModelSim, a register transistor logic simulation environment. In the simulation, the power consumption by LPDDR2 SDRAM(2 Gb, × 32 width, burst 4) is measured using a power calculator provided by Micron [32].

with the FMCs can achieve the positive amount of total power gains as shown in the fourth column. However, due to the difference in the internal power gain, the proposed FMC achieves more than twice the total power gain compared to the 2-D FMC [13].

VI. Conclusion

An increase of video resolution in mobile multimedia devices has led to the wider use of the FMC. The combination of DWT and SPIHT algorithms is suitable for the FMC due to its high compression efficiency with low computational complexity. For increasing the efficiency of the DWT and SPIHT algorithms, the proposed study makes three significant contributions. First, the proposed scheme remarkably improves the video quality by internally applying the adaptive CRs of the SPIHT algorithm to various coding blocks based on DWT coefficients while maintaining the final bit-stream size. Second, the mathematical model and optimization schemes are proposed in order to determine the optimum rate allocation. Experimental results show that a maximum PSNR enhancement of up to 2.23 dB can be achieved with a negligible increase in computations and resources when compared to the results from the conventional DWT and SPIHT algorithms. Third, for implementing an efficient video recording system, the proposed module is implemented as hardware and integrated with H.264/HEVC video coding standards as the FMC use. This integrated system is also implemented in hardware and verified the operation on the FPGA board. Thanks to the use of the proposed FMC, the power consumption in video coding standards has been significantly reduced while maintaining a similar video quality. It is possible to apply the proposed EC modules to the video processing module as well as the video recording module because all video systems store their inputs in the external memory. The proposed system is expected to contribute greatly to the generalization of the high-performance FMC.

References

[1] M. Casares and S. Velipasalar, "Adaptive methodologies for energy- efficient object detection and tracking with battery-powered embedded smart cameras," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 10, pp. 1438–1452, Oct. 2011.

[2] S. Park, I. Hong, J. Park, and H. Yoo, "An energy-efficient embedded deep neural network processor for high speed visual attention in mobile vision recognition SoC," *IEEE J. Solid State Circuits*, vol. 51, no. 10, pp. 2380–2388, Oct. 2016.

[3] X. Bao, D. Zhou, P. Liu, and S. Goto, "An advanced hierarchical motion estimation scheme with lossless frame recompression and early-level termination for beyond high-definition video coding," *IEEE Trans. Multimedia*, vol. 14, no. 2, pp. 237–249, Apr. 2012.

[4] L. Guo, D. Zhou, and S. Goto, "A new reference frame recompression algorithm and its VLSI architecture for UHD TV video codec," *IEEE Trans. Multimedia*, vol. 16, no. 8, pp. 2323–2332, Dec. 2014.

[5] H.-C. Kuo and Y.-L. Lin, "A hybrid algorithm for effective lossless compression of video display frames," *IEEE Trans. Multimedia*, vol. 14, no. 3, pp. 500–509, Jun. 2012.

[6] Y. Jin, Y. Lee, and H.-J. Lee, "A new frame memory compression algorithm with DPCM and VLC in a 4×4 block," *EURASIP J. Adv. Signal Process.*, vol. 2009, no. 629285, pp. 1–18, 2009.

[7] T. Y. Lee, "A new frame-recompression algorithm and its hardware design for MPEG-2 video decoders," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 6, pp. 529–534, Jun. 2003.

[8] S. Kim, D. Lee, J.-S. Kim, and H.-J. Lee, "A block truncation coding algorithm and hardware implementation targeting 1/12 compression for LCD overdrive," *J. Display Technol.*, vol. 12, no. 4, pp. 376–389, Apr. 2016.

[9] H. Kim, C. E. Rhee, J.-S. Kim, S. Kim, and H.-J. Lee, "Power-aware design with various low-power algorithms for an H.264/AVC encoder," in *Proc. IEEE Int. Symp. Circuits Syst.*, May 2011, pp. 571–574.

[10] H. Kim, C. E. Rhee, and H.-J. Lee, "An effective combination of power scaling for H.264/AVC compression," *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 23, no. 11, pp. 2685–2689, Nov. 2015.

[11] S. Kim, D. Lee, J.-S. Kim, and H.-J. Lee, "A high-throughput hardware design of a one-dimensional SPIHT algorithm," *IEEE Trans. Multimedia*, vol. 18, no. 3, pp. 392–404, Mar. 2016.

[12] T. Yng, B.-G. Lee, and H. Yoo, "A low complexity and lossless frame memory compression for display devices," *IEEE Trans. Consum. Electron*, vol. 54, no. 3, pp. 1453–1458, Aug. 2008.

[13] Y. Jin and H.-J Lee, "A block-based pass-parallel SPIHT algorithm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 7, pp. 1064–1075, Jul. 2012.

[14] G. K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consum. Electron.*, vol. 38, no. 1, pp. xviii–xxxiv, Feb. 1992.

[15] C. Christopoulos, A. Skodras, and T. Ebrahimi, "The JPEG2000 still image coding system: An overview," *IEEE Trans. Consum. Electron.*, vol. 46, no. 4, pp. 1103–1127, Nov. 2000.

[16] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Trans. Commun.*, vol. COMM-27, no. 9, pp. 1335–1342, Sep. 1979.

[17] J. Kim and C.-M. Kyung, "A lossless embedded compression using significant bit truncation for HD video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 6, pp. 848–860, Jun. 2010.

[18] Khalid Saywood, *Introduction to Data Compression*. San Mateo, CA, USA: Morgan Kaufmann, 2005, pp. 4–5, 497.

[19] A. Said and W. Pearlman, "A new, fast, and efficient image codec based on set partitioning in hierarchical trees," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, no. 3, pp. 243–250, Jun. 1996.

[20] P.-Y. Chen, "VLSI implementation for one dimensional multilevel lifting-based wavelet transform," *IEEE Trans. Comput.*, vol. 53, no. 4, pp. 386–398, Apr. 2004.

[21] H. Kim, C. E. Rhee, and H. J. Lee, "A low-power video recording system with multiple operation modes for H.264 and light-weight compression," *IEEE Trans. Multimedia*, vol. 18, no. 4, pp. 603–613, Apr. 2016.

[22] M. J. Shensa, "The discrete wavelet transform: wedding the a trous and Mallat algorithm," *IEEE Trans. Signal Process.*, vol. 40, no. 10, pp. 2464–2482, Oct. 1992.

[23] J. M. Shapiro, D. S. R. Center, and N. J. Princeton, "Embedded image coding using zerotrees of wavelet coefficients," *IEEE Signal Process.*, vol. 41, no. 12, pp. 3445–3462, Dec. 1993.

[24] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Process.*, vol. 9, no. 7, pp. 1158–1170, Jul. 2000.

[25] W. A. Pearlman, A. Islam, N. Nagaraj, and A. Said, "Efficient, low-complexity image coding with a set-partitioning embedded block coder," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 11, pp. 1219–1235, Nov. 2004.

[26] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, Apr. 2004.

[27] Video Test Media. [Online]. Available: https://media.xiph.org/video/derf/. Accessed on: Sep. 19, 2017.

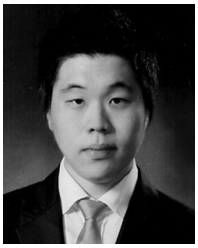
[28] A. D. Gupte, B. Amrutur, M. M. Mehendale, A. V. Rao, and M. Budagavi, "Memory bandwidth and power reduction using lossy reference frame compression in video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 2, pp. 225–230, Jan. 2011.

[29] H. Kim and H.-J. Lee, "A low-power surveillance video coding system with early background subtraction and adaptive frame memory compression," *IEEE Trans. Consumer Electron.*, vol. 63, no. 4, pp. 359–367, Nov. 2017.

[30] J. Wu, C. Leng, Y. Wang, Q. Hu, and J. Cheng, "Quantized Convolutional Neural Networks for Mobile Devices," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2016, pp. 4820–4828.

[31] G. Bjontegaard, "Calculation of average PSNR differences between RD curves," Doc. VCEG-M33, ITU-T Q6/16, Austin, TX, USA, Apr. 2001.

[32] Micron LPDDR2 Power Calculator. [Online]. Available: http://www.micron.com/~/media/documents/products/power-calculator/tn4201_lpddr2_system_power_calculator.xlsx. Accessed on: Sep. 19, 2017.



Hyun Kim received the B.S., M.S. and Ph.D. degrees in electrical engineering and computer science from Seoul National University, Seoul, Korea, in 2009, 2011 and 2015, respectively. From 2015 to 2018, he was with the BK21 Creative Research Engineer Development for IT, Seoul National University, Seoul, Korea, as a Research Professor. In 2018, he joined the Department of Electrical and Information Engineering, Seoul National University of Science and Technology, Seoul, where he is currently working as an Assistant Professor. His research interests are in the areas of algorithm, computer architecture, and SoC design for low-complexity multimedia applications.

Albert No received the B.Sc. degree in both electrical engineering and mathematics from Seoul National University, Seoul, South Korea, in 2009, and the M.Sc. and Ph.D. degrees in electrical engineering from Stanford University, Stanford, CA, USA, in 2012 and 2015, respectively. From 2015 to 2017, he was a Data Scientist with Roche. In 2017, he joined Hongik University, Seoul, South Korea, where he is currently an Assistant Professor of electronic and electrical engineering. His research interests include the relation between information and estimation theory, lossy compression, and bioinformatics.

Hyuk-Jae Lee received the B.S. and M.S. degrees in electronics engineering from Seoul National University, Seoul, South Korea, in 1987 and 1989, respectively, and the Ph.D. degree in electrical and computer engineering from Purdue University, West Lafayette, IN, USA, in 1996. From 1998 to 2001, he was with the Server and Workstation Chipset Division, Intel Corporation, Hillsboro, OR, USA as a Senior Component Design Engineer. From 1996 to 1998, he was on the faculty of the Department of Computer Science, Louisiana Tech University, Ruston, LA, USA. In 2001, he joined the School of Electrical Engineering and Computer Science, Seoul National University, where he is currently a Professor. He is the founder of Mamurian Design, Inc., a fabless SoC design house for multimedia applications. His research interests include computer architecture and SoC design for multimedia applications.