Improved Lossless Intra Coding for H.264/MPEG-4 AVC

Yung-Lyul Lee, Ki-Hun Han, and Gary J. Sullivan, *Fellow, IEEE*

***Abstract*—A new lossless intra coding method based on sample-by-sample differential pulse code modulation (DPCM) is presented as an enhancement of the H.264/MPEG-4 AVC standard. The H.264/AVC design includes a multidirectional spatial prediction method to reduce spatial redundancy by using neighboring samples as a prediction for the samples in a block of data to be encoded. In the new lossless intra coding method, the spatial prediction is performed based on samplewise DPCM instead of in the block-based manner used in the current H.264/AVC standard, while the block structure is retained for the residual difference entropy coding process. We show that the new method, based on samplewise DPCM, does not have a major complexity penalty, despite its apparent pipeline dependencies. Experiments show that the new lossless intra coding method reduces the bit rate by approximately 12% in comparison with the lossless intra coding method previously included in the H.264/AVC standard. As a result, the new method is currently being adopted into the H.264/AVC standard in a new enhancement project.**

***Index Terms*—AVC, differential pulse code modulation (DPCM), H.264, intra coding, lossless image coding, lossless video coding, MPEG-4, spatial prediction.**

I. Introduction

THE H.264/MPEG-4 AVC [1]–[4], a recently developed video coding standard jointly developed as ITU-T Recommendation H.264 and ISO/IEC 14496-10 (MPEG-4 Part 10) Advanced Video Coding (AVC), is well known to provide efficient lossy coding of video content—enabling reduction of the bit rate by approximately 30%–70% when compared with previous video coding standards such as MPEG-4 Part 2 [5], H.263 [6], H.262/MPEG-2 Part 2 [7], etc., while providing the same or better image quality.

The lossless encoding capabilities of H.264/MPEG-4 AVC are less well known. The original version of the standard included a so-called *pulse-code modulation* (PCM) macroblock coding mode that allowed samples of selected macroblocks to be represented losslessly, although very inefficiently from a compression perspective (actually slightly expanding the quantity of data used to represent the PCM-encoded macroblocks). The second major version of the standard, which included the so-called *fidelity range extensions* (FRExt) [8], added design improvements for more efficient lossless encoding. In this paper, we present an improvement over the lossless coding feature found in the FRExt design. The improved method described herein was recently adopted for standardization in a new amendment to the H.264/MPEG-4 AVC standard [9].

In digital video coding, each picture of a video sequence is typically represented by three rectangular arrays of integer-valued samples. One of these arrays is typically called the *luma* component and represents the brightness of the content at each sample location in the picture. The other two arrays are typically called the *chroma* arrays, and represent the color-difference chromaticity at each sample location in the picture. When the three arrays all have the same dimensions, the sampling structure is referred to as "4:4:4". Often (especially in consumer-quality video applications), each chroma array has half the width and half the height of the luma array, resulting in a sampling structure called "4:2:0," and sometimes (especially for professional-quality interlaced-scan video) the chroma arrays have half the width but the same height as the luma array, resulting in a sampling structure called "4:2:2." Alternatively, rather than having the three arrays represent luma and chroma information, they may represent the intensity of red, green, and blue component signals to combine to represent the color and brightness at each location.

H.264/AVC has a *block-based* coding structure. In its design (as in many others since H.261 [10]), each picture is segmented into *macroblocks*, which consist of an array of 16×16 luma (or green component) samples[1] and two associated $M \times N$[2] arrays of chroma (or red and blue) samples, where M is 16 for 4:4:4 and is 8 for 4:2:2 and 4:2:0, and where N is 16 for 4:4:4 and 4:2:2 and is 8 for 4:2:0. Each macroblock is further decomposed into *blocks*, consisting of single-component sample arrays of various sizes from as large as 16×16 to as small as 4×4. By various mechanisms, each block is predicted in a manner indicated by the encoder, and then the residual difference block

[1]Following the convention used in the H.264/AVC standard, we use the term *sample*, rather than *pixel*, herein for reasons of clarity. In the graphics context, the term *pixel* typically refers collectively to all three color components for a particular location in the sampling grid, and all three components are typically found at every location (i.e., the sampling structure is ordinarily 4:4:4). In video and image coding, the concept of a pixel is typically not as well defined since other sampling structures may be in use such that the samples of different color components are not as tightly coupled together for interpretation of the image content, whereas a sample is generally clearly understood to be a scalar quantity for a single color component.

[2]Following the convention used in the H.264/AVC standard, we refer to the dimensions of a rectangular array (or *block*) as $M \times N$ when it contains M columns horizontally and N rows vertically. This convention may differ from that typically used in other contexts.

Manuscript received June 24, 2005; revised December 21, 2005. This work was supported in part by the Ubiquitous Autonomic Computing and Network Project, the Ministry of Science and Technology (MOST) 21st Century Frontier R&D Program, Korea. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Bruno Carpintieri.

Y.-L. Lee and K.-H. Han are with the Department of Computer Engineering, Sejong University, Seoul, Korea, 143-747 (e-mail: yllee@sejong.ac.kr; khhan@dms.sejong.ac.kr).

G. J. Sullivan is with the Microsoft Corporation, Redmond, WA 98052 USA (e-mail: garysull@microsoft.com).

Digital Object Identifier 10.1109/TIP.2006.877396

representing the difference between the predicted block and the actual source picture's content block is represented in a compact manner. When using the lossy coding features, the compression encoding process consists of application of a block linear transform followed by quantization and entropy coding of the block of transformed coefficients. When using the lossless coding capability, the encoding cannot be performed by quantization and, thus, consists only of entropy coding of the block of difference data.

There are two fundamental types of block-based prediction in H.264/AVC: *intra*picture prediction, which consists of using selectable position-dependent linear combinations of neighboring sample values to form a prediction block, and *inter*picture prediction, which consists of using one or two position-interpolated blocks of samples from previously coded pictures to form a prediction block. In either case, the goal of the prediction processing is to reduce the quantity of data needed to adequately represent the corresponding block of input picture samples. Typically, inter prediction provides better compression capability when other pictures have already been coded, but intra prediction is useful for various purposes including random access, ease of video sequence editing, coding of individual still pictures, and representation of significant changes of scene content.

Although H.264/AVC was designed primarily for lossy video coding, efficient lossless compression was included in the FRExt capability set as well. The desire in this case was primarily to enable the lossless encoding of individual isolated regions in a reasonably efficient manner—not to be a special-purpose encoding design targeting lossless compression as its primary goal. However, it was found that good lossless compression performance could be achieved while retaining most of the H.264/AVC design components without alteration (or with minor alterations). The resulting FRExt design, when compared to other well-known lossless coding schemes, seems particularly effective for lossless coding when using interpicture prediction. Its performance for lossless intra coding is comparatively less state-of-the-art.

Herein, we focus on improving the intra coding case, to improve its performance without a major complexity impact and without major changes for other parts of the design structure. This is partly motivated by the knowledge that the current FRExt design is not as effective at lossless intra compression as some others.

In the H.264/AVC intra prediction case, various block sizes are used, such as 16×16, 8×8 or 4×4 luma block prediction (with the block size and direction of prediction being selected by the encoder), and 16×16 (for 4:4:4 video), 16×8 (for 4:2:2 video), or 8×8 (for 4:2:0 video) chroma block prediction. The new spatial prediction process described herein is performed using sample-by-sample DPCM instead of block-based prediction. This results in better coding efficiency, since a sample immediately neighboring the sample to be predicted is typically a better predictor than a sample in a neighboring block that is several samples farther away. Although the use of such a sample-wise DPCM in the encoder may appear to break the blockwise pipeline processing structure of the decoding process, we retain the use of block-based processing for the residual difference coding, and show that the samplewise prediction is conceptually analogous to the application of a spatial residual transform as is ordinarily applied in the lossy video decoding process. The samplewise DPCM can, thus, be used with no major increase in computational complexity relative to the ordinary decoding process. By using the samplewise DPCM, a bit-rate reduction of approximately 12% is shown in the lossless intra coding compression performance. As a result, the new technique has recently been adopted as part of a new draft amendment of the H.264/AVC standard.

II. Spatial Prediction in H.264

H.264/AVC uses several spatial prediction block sizes including 16×16, 8×8, and 4×4 for luma and 16×16 (for 4:4:4 video), 16×8 (for 4:2:2 video), or 8×8 (for 4:2:0 video) for chroma. For each case, the encoder selects a directional spatial prediction mode which governs the creation of a prediction of the complete block of samples using the values of samples in neighboring blocks that have previously been decoded—specifically, the column of samples immediately to the left of the block to be predicted and the row of samples immediately above the block to be predicted.

Specifically, for 16×16 luma intra prediction and for chroma intra prediction, one of four prediction modes can be chosen—the horizontal, vertical, DC, and plane modes. Horizontal prediction consists of extrapolation towards the right from the values of the samples in the neighboring column, vertical prediction consists of extrapolation downward from the values of the samples in the neighboring row, DC prediction consists of generating a prediction block using the average of values in the neighboring column and row, and, finally, plane prediction consists of generating a prediction block using a three-parameter equation based on the mean and the horizontal and vertical slope exhibited in the neighboring row and column.

For 8×8 or 4×4 intra prediction, more prediction modes are available, using similar concepts—in these cases, there are nine different intra prediction modes that can be chosen, with conceptual prediction directions as illustrated in Fig. 1 (mode 2, not shown in the figure, is the "DC" averaging mode). For the luma component, in addition to providing the ability for the encoder to select among the various prediction directions, the encoder can also select which block size will be used for the prediction process—with smaller block sizes generally resulting in better prediction quality but using a greater number of bits to specify the specify the prediction due to the larger number of blocks for which to select the directional prediction mode.

These features of the H.264/AVC design allow the encoder to customize the prediction process to fit the particular local characteristics of the content of the video picture that is to be encoded, and to use either a relatively small or large quantity of data to govern the prediction process as necessary. However, a fundamental shortcoming of the existing design, for purposes of lossless coding, is the need to generate the entire block of prediction values all at once. In contrast, high-performance lossless image coding designs typically use very small prediction neighborhoods and perform the prediction and residual representation processes on an individual sample-by-sample basis in a tightly coupled fashion.

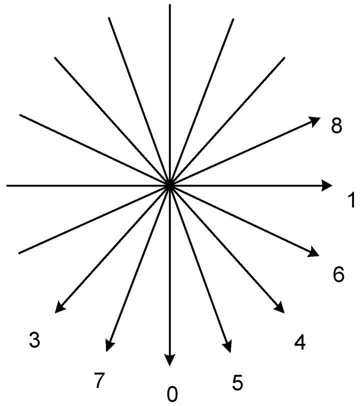


Fig. 1. Nine prediction modes for the intra 4 × 4 prediction in the H.264 standard.

l_0	l_1	l_2	l_3	l_4	l_5	l_6	l_7	l_8
q_0	p_0	p_1	p_2	p_3				
q_1	p_4	p_5	p_6	p_7				
q_2	p_8	p_9	p_{10}	p_{11}				
q_3	p_{12}	p_{13}	p_{14}	p_{15}				

Fig. 2. Boundary samples and inside samples for intra 4 × 4 prediction.

As an example, using the horizontal prediction method found in H.264/AVC for 4 × 4 luma blocks which is marked as mode 1 in Fig. 1, to predict a 4 × 4 block to be encoded as shown in Fig. 2, the prediction of every sample in the row of the block next to the neighboring column sample with value q_0, the prediction is the value q_0. Thus, if we denote the residual difference to be encoded for the encoding of the samples marked as p_0 through p_3, as r_0, through r_3, we see that

$$r_0 = p_0 - q_0 \quad (1)$$
$$r_1 = p_1 - q_0 \quad (2)$$
$$r_2 = p_2 - q_0 \quad (3)$$
$$r_3 = p_3 - q_0. \quad (4)$$

The residual differences r_0 through r_3, in (1)–(4) that are predicted from the block boundary samples are entropy coded as part of a residual 4 × 4 block and the entropy-coded data is sent from the encoder to the decoder. The decoder subsequently reconstructs the residual samples, creates the prediction block, and adds the residual block to reconstruct the final picture. The processing in the decoder preserves the pipeline processing design of the H.264/AVC coding structure, since it is implemented in a block-based processing manner. The vertical prediction is performed in a similar way to the horizontal prediction in the vertical direction. The mode 4 prediction direction shown in Fig. 1 is performed using the predictor, $(q_0+2\cdot l_0+l_1+2) \gg 2$, in the case where the p_0, p_5, p_{10}, and p_{15} samples are predicted, and all other remaining samples are predicted from three similarly chosen block boundary samples by considering the prediction direction in a similar way. As another example, the mode 5 prediction in Fig. 1 is performed by using the predictor, $(l_0 + l_1 + 1) \gg 1$, in the case where the p_0 and p_9 samples are predicted, and all other remaining samples are predicted from two or three block boundary samples by considering the prediction direction in a similar way. Here, we use the notation "$x >> y$" to denote an arithmetic right shift of a two's complement representation of an integer-valued quantity x by a number of bit positions y.

The other directional modes in Fig. 1 are performed in a similar ways to modes 1, 4, and 5, with alterations from the viewpoint of the prediction directions. The H.264/AVC lossless intra coding process is performed by using the spatial prediction method shown in Fig. 1 and entropy coding the difference block without applying the remaining elements of the design that are used in the lossy case—specifically, the 4 × 4 integer DCT-like transform and quantization. We particularly note that a 4 × 4 integer DCT-like transform is ordinarily a part of the H.264/AVC decoding process, but is skipped in the lossless coding cases.

III. Proposed Intra Lossless Coding Method Using DPCM and Residual Transform

In H.264/AVC, since the 4 × 4 integer DCT and quantization cause data loss due to rounding or shifting operations, the transform and quantization processes are skipped, in order to achieve lossless compression [11]. To perform intra lossless coding, H.264/AVC uses blockwise entropy coding to code the prediction error signals after the intra predictions. In order to improve the coding efficiency in intra lossless coding, we propose a method which employs both DPCM and its residual transform. By using the new method, the coding efficiency can be enhanced, as compared with block-based prediction.

A. Horizontal Prediction

As an alternative to the prediction found currently in H.264/AVC, as presented in (1)–(4), we suggest that when the horizontal prediction (mode 1), as shown in Fig. 1, is applied for the first row of a 4 × 4 block on the samples shown in Fig. 2, the residuals, r_0, r_1, r_2, and r_3, of each sample, p_0, p_1, p_2, and p_3, are calculated by sample-by-sample DPCM as follows:

$$r_0 = p_0 - q_0 \quad (5)$$
$$r_1 = p_1 - p_0 \quad (6)$$
$$r_2 = p_2 - p_1 \quad (7)$$
$$r_3 = p_3 - p_2. \quad (8)$$

The encoder sends r_0, r_1, r_2, and r_3 as part of a residual block, and the decoder can then decode the residuals as a block and then apply them for reconstruction as shown in (5)–(8). However, the blockwise processing structure of the decoder appears to be broken by this behavior, since samplewise DPCM, which is based on samplewise prediction, always requires each individual decoded sample to be provided sequentially prior to the prediction and reconstruction of the next sample value. As an example, in order to reconstruct sample p_3, sample p_2 should be reconstructed in advance using (7), which, in turn, requires sample p_1 to have been reconstructed from (6), which requires sample p_0 to have been reconstructed from (5). To solve this apparent problem of breaking the blockwise pipeline structure of the decoding process, we decompose the prediction structure into an alternative form based on the use of a residual transform in the decoder. From (5)–(8), we can derive the following:

$$p_0 = r_0 + q_0 \quad (9)$$

$$p_1 = r_0 + r_1 + q_0 \quad (10)$$

$$p_2 = r_0 + r_1 + r_2 + q_0 \quad (11)$$

$$p_3 = r_0 + r_1 + r_2 + r_3 + q_0. \quad (12)$$

Generalizing, we obtain the following relationship for the first row of the 4 × 4 block:

$$p_i = q_0 + \sum_{k=0}^{i} r_k, \quad 0 \le i \le 3. \quad (13)$$

This relationship can be expressed in matrix form as follows:

$$\begin{bmatrix} p_0 \\ p_1 \\ p_2 \\ p_3 \end{bmatrix} = q_0 \begin{bmatrix} 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} + \begin{bmatrix} 1 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ r_3 \end{bmatrix}. \quad (14)$$

The 4 × 4 matrix in (13) can be considered as a spatial residual transform that is slightly simpler than the spatial residual transform ordinarily used in the lossy coding process for H.264/AVC, thus illustrating that the sample reconstruction process when using samplewise DPCM prediction does not substantially increase computational complexity over the ordinary types of processing already found there.

B. Vertical and Other Prediction Modes

The vertical prediction can be performed in a similar way to the horizontal prediction.

In a very similar manner to the vertical or horizontal prediction, the residual transform using samplewise DPCM, which employs one sample as a predictor for each other sample, can be applied also to the mode 3 and mode 4 prediction modes for 4 × 4 blocks shown in Fig. 1, but not as easily to the DC mode or prediction modes 5, 6, 7, and 8, since H.264/AVC uses a combination of two or three block boundary samples as the sample predictor in the case of all the other directional predictions except for the DC, the horizontal and vertical predictions.

The new intra lossless coding method can be applied directly to any $M \times N$ block in the case where the horizontal (mode 1), vertical (mode 0), mode 3, and mode 4 predictions can be implemented with a one sample predictor.

IV. Experimental Results

In this paper, a lossless intra coding method based on samplewise DPCM has been presented in the context of the block-based H.264/AVC design. To verify the validity of this method, experiments are performed on several YUV 4:2:0 format test sequences with QCIF (Quarter Common Intermediate Format) and CIF resolutions, as shown in Table I, all of which have 300 frames and 30-Hz temporal frequencies. For the entropy coding, the *context-adaptive binary arithmetic coding* (CABAC) mode of H.264/AVC is used for the experiment.

TABLE I
Experimental Sequences and Coding Conditions

Image	Total frames (Hz)	Coding Options
News (QCIF)	300 (30Hz)	Rate-Distortion Optimization, CABAC, and only Intra frames are used
Container (QCIF)	300 (30Hz)	
Foreman (QCIF)	300 (30Hz)	
Silent (QCIF)	300 (30Hz)	
Paris (CIF)	300 (30Hz)	
Mobile (CIF)	300 (30Hz)	
Tempete (CIF)	260 (30Hz)	

In order to evaluate the performance of the modified method, the H.264/AVC lossless intra coding algorithm using the YUV 4:2:0 format is also implemented and tested. The performance of the new method is compared with that of the H.264/AVC intra lossless coding method using the YUV 4:2:0 format based on block prediction.

The new method is applied to 16 × 16 and 4 × 4 luma blocks and 8 × 8 chroma blocks, according to the 4:2:0 sampling structure. It is not applied to 8 × 8 luma block prediction, as that type of prediction uses combinations of samples rather than single sample values for its prediction operation. The first samplewise prediction method (marked method 1) in Table II uses the presented samplewise horizontal/vertical predictions and uses the other seven prediction modes of the H.264/AVC lossless intra coding without alteration for 4 × 4 luma blocks, and the samplewise horizontal/vertical predictions and DC and plane prediction modes of the H.264/AVC lossless intra coding method for the 16 × 16 luma block and 8 × 8 chroma block. Method 1 achieved an average compression ratio of 1.889:1 for various sequences, while the H.264/AVC intra lossless coding standard (H.264/AVC) in Table II achieved an average compression ratio of only 1.670:1. The second samplewise prediction method (method 2) in Table II uses samplewise horizontal (mode 1), vertical (mode 0), mode 3 and mode 4 predictions and the other five predictions of the H.264/AVC lossless intra coding method for the 4 × 4 luma block, and samplewise horizontal/vertical predictions, and the DC and plane predictions of the H.264/AVC lossless intra coding method for the 16 × 16 luma and 8 × 8 chroma blocks. Method 2 shows an average compression ratio of 1.899:1. For comparison to more state-of-the-art lossless coding techniques, the lossless coding efficiency of JPEG-LS (lossless) [12] and Motion JPEG2000 lossless [13] are compared with that of the modified H.264/AVC methods and with that of H.264/AVC lossless coding without alteration to show the efficiency of the samplewise prediction methods. For the Motion JPEG2000 results we used the "Jasper-1.701.0" software that can be downloaded from the JPEG site.

Based on these experimental results, we can state that samplewise prediction method 1 shows an improvement in the compression ratio compared with the block-based H.264/AVC lossless intra coding method and the Motion JPEG2000, while not substantially increasing the processing complexity in the decoder. Although samplewise prediction method 2 gives a

TABLE II
COMPARISON OF THE COMPRESSION RATIOS FOR H.264 INTRA LOSSLESS CODING, PROPOSED METHODS 1 AND 2, JPEG-LS (LOSSLESS), AND MOTION JPEG2000 LOSSLESS (M-JP2K)

Image	Original Image Size(Bits)	Method	Total Bits	Compr ession Ratio	Saving Bits (%)
News	91238400	H.264/AVC	49062832	1.859624	0
		Method1	42004648	2.172102	14.386
		Method 2	41901888	2.177429	14.595
		JPEG-LS	38493000	2.370260	21.543
		M-JP2K	44094160	2.069172	10.127
Cont ainer	91238400	H.264/AVC	47836576	1.907294	0
		Method1	42222416	2.160900	11.736
		Method 2	42194984	2.162304	11.793
		JPEG-LS	40503200	2.252622	15.330
		M-JP2K	44423256	2.053843	7.135
Fore man	91238400	H.264/AVC	50418312	1.809628	0
		Method1	46101624	1.979071	8.562
		Method 2	45233104	2.017071	10.284
		JPEG-LS	43903664	2.078150	12.921
		M-JP2K	48250840	1.890918	4.299
Silent	91238400	H.264/AVC	54273064	1.681099	0
		Method1	48020632	1.899983	11.520
		Method 2	47735224	1.911343	12.046
		JPEG-LS	44656200	2.043130	17.719
		M-JP2K	47552944	1.918670	12.382
Paris	364953600	H.264/AVC	224766912	1.623698	0
		Method1	194434228	1.877003	13.495
		Method 2	193983792	1.881361	13.696
		JPEG-LS	179265368	2.035829	20.244
		M-JP2K	196161712	1.860473	12.727
Mob ile	364953600	H.264/AVC	285423632	1.278638	0
		Method1	258162856	1.413657	9.551
		Method 2	257077240	1.419626	9.931
		JPEG-LS	231103384	1.579179	19.031
		M-JP2K	240223216	1.519227	15.836
Tem pete	316293120	H.264/AVC	205817192	1.536767	0
		Method1	183241664	1.726098	10.969
		Method 2	182992200	1.728451	11.090
		JPEG-LS	166747848	1.896835	18.983
		M-JP2K	175970768	1.797419	14.501
Ave rage		H.264/AVC		1.670964	0
		Method1		1.889831	11.460
		Method 2		1.899655	11.919
		JPEG-LS		2.036572	17.967
		M-JP2K		1.872817	11.001

slightly better results than method 1, the degree of that improvement is insignificant. Moreover, samplewise prediction method 1 can be easily applied to H.264/AVC lossless intra coding by modifying only the semantics and decoding process, without any syntax changes being required to the H.264/AVC standard. From the experiments of lossless intra coding, we, thus, can conclude that among the four prediction modes that can be most easily modified to use samplewise DPCM prediction, the horizontal and vertical prediction modes have primary influence over the coding efficiency (however, the relative importance of the different prediction modes may be different in lossy intra coding).

V. CONCLUSION

The modified intra lossless coding method using DPCM and its residual transform can be applied to the horizontal and vertical predictions of any $\mathrm{M} \times \mathrm{N}$ block samples. An overall improvement in lossless coding compression capability of approximately 12% has been shown in the experiment results, without a substantial increase in the complexity of the encoding or decoding processes. The modified methods can be considered as a useful new tool for future block-based intra lossless coding designs. Indeed, based on similar results to those shown in this paper, the samplewise DPCM [14], [15].

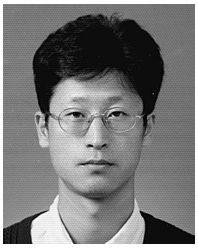
The intra coding method that is referred to as method 1 in this paper was recently adopted for standardization in an enhanced future version of the H.264/MPEG-4 AVC standard, in a package of capabilities currently known as the Advanced 4:4:4 profile [9]. Simulation results on RGB 4:4:4 format data can be found in [14] and [15].

ACKNOWLEDGMENT

The authors would like to thank Dr. S. Sun for his very helpful comments and experimental results verification concerning H.264/AVC lossless coding. They would also like to thank the reviewers for their through review and very helpful comments.

REFERENCES

[1] ITU-T Recommendation H.264 and ISO/IEC 14496-10, Advanced Video Coding for Generic Audiovisual Services, May 2003, (and subsequent amendment [8] and corrigenda).
[2] A. Luthra, G. J. Sullivan, and T. Wiegand, "Introduction to the special issue on the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 557–559, Jul. 2003.
[3] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[4] G. J. Sullivan and T. Wiegand, "Video compression—from concepts to the H.264/AVC standard," *Proc. IEEE*, no. 1, pp. 18–31, Jan. 2005.
[5] ISO/IEC 14496-2, Information Technology—Coding of Audio-Visual Objects Part 2: Visual, Dec. 1998, (and subsequent amendments and corrigenda).
[6] ITU-T Video Coding Experts Group (VCEG), Video Codec Test Model Near-Term, Version 10 (TMN10) Draft 1, Apr. 1998.
[7] ITU-T Recommendation H.262 and ISO/IEC 13818-2, Information Technology—Generic Coding of Moving Pictures and Associated Audio Information: Video, Jul. 1995, (and subsequent amendments and corrigenda).
[8] G. J. Sullivan, T. McMahon, T. Wiegand, and A. Luthra, Eds., Draft Text of H.264/AVC Fidelity Range Extensions Amendment to ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team document JVT-L047, Jul. 2004.
[9] H. Yu, Ed., Draft Text of H.264/AVC Advanced 4:4:4 Profile Amendment to ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team document JVT-Q209, Oct. 2005.
[10] ITU-T Recommendation H.261, Video Codec for Audiovisual Services at $p \times 64$ kbits, Nov. 1990, (with subsequent amendment).
[11] S. Sun, Intra Lossless Coding and QP Range Selection, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team document JVT-C023, May 2002.
[12] M. J. Weinberger, "The LOCO-I lossless image compression algorithm: principles and standardization into JPEG-LS," *IEEE Trans. Image Process.*, vol. 9, no. 8, pp. 1309–1324, Aug. 2000.
[13] *Motion JPEG2000 Final Committee Draft*, ISO/IEC JTC 1/SC29/WG 1, Document N2117, Mar. 2001.
[14] Y.-L. Lee, K.-H. Han, and S.-C. Lim, Lossless Intra Coding for Improved 4:4:4 Coding in H.264/MPEG-4 AVC, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16 Joint Video Team document JVT-P016, Jul. 2005.
[15] Y.-L. Lee and K.-H. Han, Complexity of the Proposed Lossless Intra, ISO/IEC JTC1/SC29/WG11 and ITU-T Q6/SG16, Document JVT-Q035r1, Oct. 2005.



Yung-Lyul Lee received the B.S. and M.S. degrees in electronic engineering from Sogang University, Seoul, Korea, in 1985 and 1987, respectively, and the Ph.D. degree in electrical and electronic engineering from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea, in 1999.

He has been an Associate Professor with the School of Computer engineering, Sejong University, Seoul, since 2001. He was a Principal Researcher at Samsung Electronics Co., Ltd., from 1987 to 2001. His current research interests include video compression, image processing, 3-D video coding, scalable video coding, and multimedia systems.

Ki-Hun Han received the B.S. and M.S. degrees in computer engineering from Sejong University, Seoul, Korea, in 2001 and 2003, respectively, where he is currently pursuing the Ph.D. degree.

His current research interests include video compression, image processing, multimedia systems, and future video coding technologies.

Gary J. Sullivan (S'83–M'91–SM'01–F'06) received the B.S. and M.Eng. degrees in electrical engineering from the University of Louisville, Louisville, KY, in 1982 and 1983, respectively, and the Ph.D. and Eng. degrees in electrical engineering from the University of California, Los Angeles, in 1991.

He is the ITU-T Rapporteur/Chairman of the ITU-T Video Coding Experts Group (VCEG), a Co-Chairman of the ISO/IEC Moving Picture Experts Group (MPEG) video coding subgroup, and a Co-Chairman of the Joint Video Team (JVT), which is a joint project between the VCEG and MPEG organizations. He has led ITU-T VCEG (ITU-T Q6/SG16) since 1996 and is also the ITU-T video liaison representative to MPEG. In MPEG (ISO/IEC JTC1/SC29/WG11), in addition to his current service as a Co-Chair of its video work, he also served as the Chairman of MPEG video work from March of 2001 to May of 2002. In the JVT, he was the JVT Chairman for the development of the next-generation H.264/MPEG-4 AVC video coding standard and for the development of its fidelity-range extensions (FRExt), and is now its Co-Chairman for the development of the scalable video coding (SVC) extensions. He holds the position of Video Architect in the Core Media Processing Team in the Windows Digital Media division of Microsoft Corporation. At Microsoft, he also designed and remains the Lead Engineer for the DirectX Video Acceleration API/DDI video decoding feature of the Microsoft Windows operating system platform. Prior to joining Microsoft in 1999, he was the Manager of Communications Core Research at PictureTel Corporation, the quondam world leader in videoconferencing communication. He was previously a Howard Hughes Fellow and Member of the Technical Staff in the Advanced Systems Division of Hughes Aircraft Corporation and was a terrain-following radar system software engineer for Texas Instruments. His research interests and areas of publication include image and video compression, rate-distortion optimization, motion representation, scalar and vector quantization, and error- and packet-loss-resilient video coding.

Dr. Sullivan received the Technical Achievement award from the International Committee on Technology Standards (INCITS) in 2005 for his work on H.264/MPEG-4 AVC and other video standardization topics.