A New Reference Frame Recompression Algorithm and Its VLSI Architecture for UHDTV Video Codec

Li Guo, Dajiang Zhou, *Member, IEEE*, and Satoshi Goto, *Life Fellow, IEEE*

***Abstract*—Video encoders and decoders for HEVC-like compression standards require huge external memory bandwidth, which occupies a significant portion of the codec power consumption. To reduce the memory bandwidth, this paper presents a new lossless reference frame recompression algorithm along with a high-throughput hardware architecture. Firstly, hybrid spatial-domain prediction is proposed to combine the merits of DPCM scanning and averaging. The prediction is then enhanced with multiple modes to accommodate various image characteristics. Finally, efficient residual regrouping based on semi-fixed-length (SFL) coding is used to improve the compression performance. Compared to no compression, the proposed scheme can reduce data traffic by an average of 57.6% with no image quality degradation. The average compression ratio is 2.49, an improvement of at least 12.2–13.2%, relative to the state-of-the-art algorithms. By applying a reordered two-step architecture and the two optimizations, residual reuse and simplified coding mode decision, the hardware cost is similar to that of previous reference frame recompression architectures. The computational complexity increase caused by multi-mode prediction affects the HW cost slightly. This work can be implemented with 45.1 k gates for the compressor and 34.5 k gates for the decompressor at 300 MHz, enough to support a 3840×2160@60fps video encoder and decoder.**

***Index Terms*—Embedded compression, H.264/AVC, HEVC, lossless reference frame recompression, multi-mode DPCM and averaging prediction.**

I. Introduction

In VIDEO codec systems that include encoders and decoders for HEVC, H.264/AVC, MPEG-2, and so on, usually a large external DRAM is required to buffer mass data such as reference frames. As a result of this huge bandwidth requirement, the power consumed by memory access occupies a significant part of the system power [1]. Therefore, techniques to overcome this large memory bandwidth problem play a vitally necessary role. Many works have been done from various points of view, including reusing the overcalled reference frame data on various levels [2]–[5] and improving the DRAM access efficiency by optimized memory controller architectures [6][7].

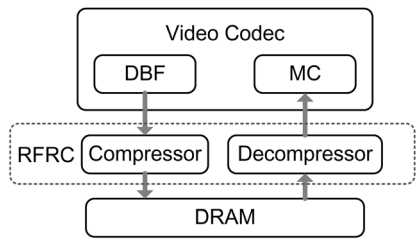


Fig. 1. Reference frame recompression, where DBF denotes "de-blocking filter" and MC denotes "motion compensation."

Embedded compression (EC) is another effective solution to the DRAM bandwidth problem. It has been widely discussed in previous work and demonstrated in several video decoder chips [8][9]. To reduce the memory data traffic, EC works as an additional layer between the codec core and the DRAM controller, that compresses the frames before storing them into DRAM and decompresses the data fetched back, as is shown in Fig. 1. Block-based and line-based ECs are distinguished by their basic processing unit. Block-based ECs [10]–[16] compress original pixel data with the information of any pixels within the same $N \times N$ block. However, the line-based ECs in [17]–[20] are mainly proposed for displaying frames line by line. Only the pixels in the same line can be used for the current pixel compression, thus the compression performance of a block-based EC is usually better than line-based. An EC for compressing the reference frames is also known as *reference frame recompression (RFRC)*. Most RFRCs are block-based ECs for better compression performance.

Existing RFRC schemes can also be roughly divided into two categories, lossy and lossless RFRC. The first category, lossy RFRC, is based on a fixed compression ratio (CR). By fixing the CR, random access of the frame data can be easily supported while ensuring bandwidth reduction. However, fixing the CR inevitably results in image quality degradation [21]–[23], while the error propagation caused by the loss in quality of the reference frames can become a more severe problem. Moreover, some blocks with a higher potential for compression can only be compressed at the designated relatively low CR, leading to the degradation of compression efficiency.

Manuscript received October 29, 2013; revised April 27, 2014; accepted August 04, 2014. Date of publication August 21, 2014; date of current version November 13, 2014. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Klara Nahrstedt.

L. Guo was with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan. She is now with the Department of Electronic Engineering, Shanghai Jiao Tong University, Shanghai 200240, China (e-mail: guoli0104@gmail.com).

D. Zhou and S. Goto are with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu 808-0135, Japan (e-mail: zhou@fuji.waseda.jp; goto@waseda.jp).

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.2014.2350256

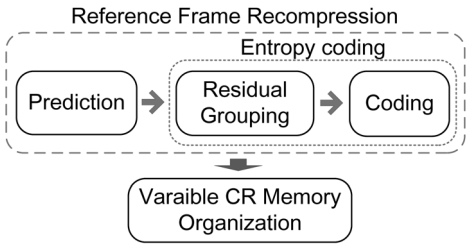


Fig. 2. Typical processing flow of lossless reference frame recompression.

The other category is lossless RFRC, which performs lossless compression and decompression. Therefore, it does not influence coding efficiency or image quality of the video encoder or decoder. Lossless RFRC schemes are based on a variable compression ratio. As a result, special memory organization is required to store and fetch the sizes of the compressed data partitions [10]. The data compression of lossless RFRC is usually composed of prediction and entropy coding, as shown in Fig. 2.

For the prediction stage, two types of algorithms are most frequently used. One type is spatial domain prediction, such as DPCM scanning [8], hierarchical minimum and difference (HMD) [11] and hierarchical average and copy (HAC) prediction [12]. The other transforms the data into the frequency domain with a discrete wavelet transform (DWT) [13], modified Hadamard transform (MHT) [17], or similar technique.

For the entropy coding stage, variable length coding (VLC) is widely used, such methods include adjusted binary coding (ABC) [14], Golomb Rice (GR) coding [15], adaptive GR coding [17] and Huffman coding [16]. Considering the difficulties in implementing high-throughput VLC hardware, the latest RFRC schemes such as semi-fixed length (SFL) [8] coding and significant bit truncation (SBT) [12] employ two-step entropy coding: first separating the residuals into small groups, and then performing fixed length coding according to the local residual feature of each group.

While several hardware implementations of lossless embedded compression schemes have been presented in [8]–[24], their throughputs are not enough to handle Ultra-High Definition TV (UHDTV) video sequences in real time. Some of them combine the compressor and decompressor together to reduce hardware cost, meaning that compression and decompression cannot occur at the same time.

This paper presents a new lossless reference frame recompression algorithm with a high-throughput hardware architecture. Firstly, to combine the merits of DPCM scanning and averaging, hybrid spatial-domain prediction is proposed. This prediction is then enhanced by multiple modes to accommodate various image characteristics. Finally, an efficient residual regrouping method based on SFL coding is used to further improve the compression performance.

The proposed hardware architecture is composed of a compressor and a decompressor. To reduce the increase in complexity caused by multi-mode prediction, a reordered two-step architecture is used for compressor, including an advance prediction mode decision step and coding step. By bringing forward the prediction mode decision, the complex coding step only needs to process one prediction mode. Moreover, by applying two optimizations, residual reuse and a simplified coding mode decision, the hardware cost of the compressor is further reduced. For clearly understanding, the abbreviations frequently used in this paper are summarized in Table I.

TABLE I
SUMMARY OF ABBREVIATIONS

Abbreviation	Full Name
CM/CMD	Coding Mode/Coding Mode Decision
CR	Compression Ratio
DPCM	Differential Pulse-code Modulation
DRR	Data Reduction Ratio
EC	Embedded Compression
HAC	Hierarchical Average and Copy
MDA	Multi-mode DPCM and Averaging
PM	Prediction Mode
RFRC	Reference Frame Recompression
RGM	Re-Grouping Mode
SBT	Significant Bit Truncation
SFL	Semi-Fixed Length Coding
SPIHT	Set Partitioning In Hierarchical Trees
TPUA	Throughput Per Unit Area
UHDTV	Ultra-High Definition TV (4kx2k)

Compared to previous reference frame recompression works, the contributions of our work are:

1) *A new reference frame recompression with high CR*: A new lossless reference frame recompression algorithm based on multi-mode DPCM & averaging prediction and a residual regrouping method is proposed, improving CR by at least 12.2–13.2% over previous methods.
2) *Efficient hardware architecture*: By applying a reordered two-step architecture and the two optimizations of residual reuse and simplified coding mode decision, the proposed algorithm can be implemented with low-cost hardware. The hardware cost of the proposed architecture is similar to that of previous lower-CR RFRC works.
3) *High-throughput hardware for a UHDTV encoder*: The throughput of the proposed scheme is up to 10.7 samples/cycle for the compressor and 21.3 samples/cycle for the decompressor. This high throughput is enough to support both a real-time 2160p UHDTV video encoder and decoder under 4:2:0 sampling at 300 MHz, a rate that has not been achieved until now.

The rest of this paper is organized as follows. The proposed multi-mode DPCM and averaging (MDA) prediction scheme is explained in Section II. The pipelined hardware architecture of the proposed MDA&SFL RFRC algorithm is described in Section III. Section IV presents the experimental results of the compression efficiency and hardware implementation for the proposed method compared to several previous works. The conclusions are drawn in Section V.

II. PROPOSED ALGORITHM

A. Multi-Mode DPCM and Averaging Prediction

The spatial correlation between neighboring pixels in natural sequences has been discussed in [12] where it is shown that the spatial correlation in the horizontal direction is stronger than in the vertical and that the averaging prediction using two neighboring pixels performs better than a prediction based on the value of one neighboring pixel. If most of the predictions in one block are based on the average prediction, some pixels are inevitably predicted by 2- or 4-pixel distance pixels, which

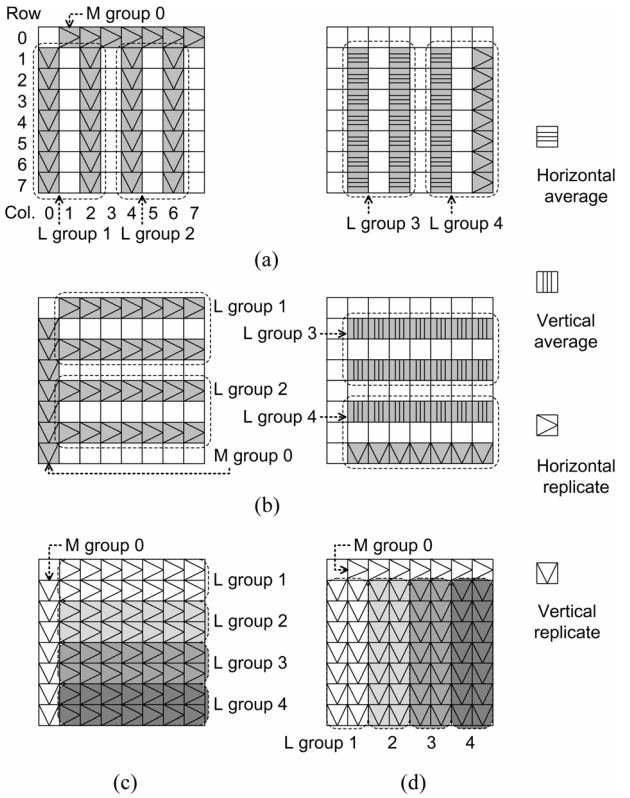


Fig. 3. Proposed MDA prediction modes and group division. (a)–(d) Prediction modes and minimum grouping division (one M-size, group 0, and four L-size, groups 1–4).

may lower the compression efficiency. The spatial correlation is weaker for prediction using longer distance pixels, as shown in [25]. Therefore, there is a tradeoff between a better prediction technique and shorter prediction distance.

As shown in Fig. 3(a), PMode 0 is a basic mode of the proposed prediction scheme. The predictions of pixels in row 0 are calculated by horizontal DPCM scanning. The current pixel is predicted by the value of the left neighboring pixel. Then 28 pixels in even columns are predicted by DPCM scanning in the vertical direction. Horizontal predictions are used for pixels in odd columns. In columns 1, 3, and 5, pixels are predicted by the average of the left and right neighboring pixels in the horizontal direction. In the final column 7, seven pixels are predicted by replicating the value of the left neighboring pixel. The remaining single top-left pixel is represented by its original 8-bit value. Therefore, the proposed PMode 0 uses only adjacent pixels for prediction, avoiding the problem of prediction by pixels at 2- or 4-pixel distances as in HAC prediction [12].

In natural sequences, the features of blocks are variable; images include gradually-changing blocks, smooth blocks, and so on. Therefore, multi-mode predictions are proposed to accommodate variable image characteristics and can improve the compression performance greatly. PMode 1 is the transpose of PMode 0, as shown in Fig. 3(b). Vertical averaging and horizontal DPCM scanning are utilized in prediction PMode 1. There are 43 different predicted pixels between PMode 0 and PMode 1. However, the directions of averaging and DPCM scanning prediction are quadrature in PMode 0 or PMode 1. Hence, these two modes are not so efficient for gradually-changing blocks. Thus, simple horizontal and vertical DPCM scanning modes (PMode 2 and PMode 3) are added, as shown in Figs. 3(c) and (d).

TABLE II
PROBABILITY THAT THE FOUR MDA PREDICTION MODES WILL BE THE SELECTED PREDICTION MODE

Qp	PMode 0	PMode 1	PMode 2	PMode 3
22	29.34%	26.88%	22.80%	20.97%
27	29.68%	27.23%	22.13%	20.96%
32	30.83%	28.15%	21.10%	19.91%
37	32.12%	29.34%	19.62%	18.93%
Average	30.49%	27.90%	21.41%	20.19%

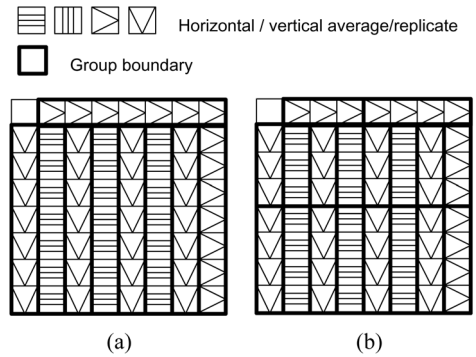


Fig. 4. Basic grouping modes. (a) Middle-size group. (b) Small-size group.

The probability that each of these four prediction modes will be chosen as the selected mode was calculated. The experimental results are shown in Table II. The probabilities given are the average of all frames in 18 test sequences. The detailed experimental conditions are discussed in Section IV. According to the experimental results, it is obvious that the addition of every prediction mode is reasonable. Hence, these four prediction modes are combined as Multi-mode DPCM and Averaging (MDA) prediction.

B. Residual Regrouping for MDA

1) Basic Residual Grouping: Residuals need to be grouped before fixed-length coding with a method such as significant bits truncation or semi-fixed length coding. An efficient residual grouping scheme is the foundation of good compression performance. To meet the features of SFL and SBT coding, we combine the residuals with similar distributions into one group. Hence the residuals predicted by the same prediction technique are usually grouped together. One 8×8 prediction block can be divided into nine groups, as shown in Fig. 4(a), according to the same prediction technique of averaging or DPCM scanning. In order to make the group sizes equal to each other, there are seven pixels in every group. Such a 7-pixel group is called middle-size group (M). However, if only one or two residuals in a middle group are distinctly large, two small-size groups (S)consisting of three or four pixels are more efficient than one middle group. A grouping example with only small groups is presented in Fig. 4(b). On the contrary, for some smooth blocks, the coding modes (CM) of SFL coding in one block are similar. Hence, a large-size group (L) is introduced to combine two middle groups together, as shown in Figs. 3(a)–(d), and the additional bits of representing the CM can be saved. The above three types of groups (S, M, and L) form the basic residual groups.

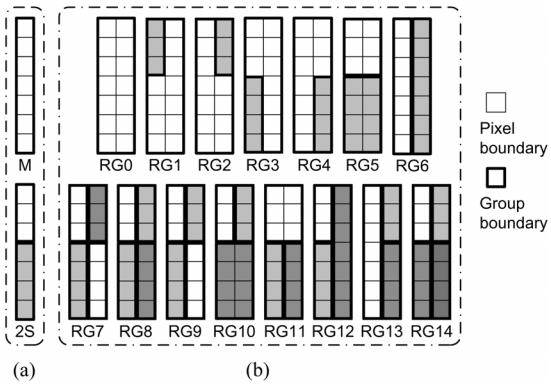


Fig. 5. Regrouping scheme. (a) Two regrouping modes for M grou p 0 and (b) 15 regrouping modes for L groups 1–4. The same gray value indicates the same CM.

TABLE III
FLAGS OF RESIDUAL REGROUPING MODES FOR LARGE-SIZE GROUPS

RGMode	RGM0	RGM1	RGM2	RGM3	RGM4
Flag	000	0010	0011	0100	0101
RGMode	RGM5	RGM6	RGM7	RGM8	RGM9
Flag	0110	0111	1000	1001	1010
RGMode	RGM10	RGM11	RGM12	RGM13	RGM14
Flag	1011	1100	1101	1110	1111

2) Residual Regrouping: For small groups, fewer bits are needed to represent residuals compared to middle or large groups. However, the overhead bits needed for CM are twice those needed for a middle group, and 3.6 times more than needed for a large group. Therefore, a scheme that regroups four small groups within one large group is proposed to reduce the overhead bits needed by small groups. In general, the coding modes of four adjacent small groups within one large group are usually similar or even the same, hence small groups with the same coding mode can be regrouped together.

As Figs. 3(a)–(d) show, one 8×8 block can be divided into four large groups and one middle group. For middle group 0, two small groups can be regrouped if their coding modes are the same, as Fig. 5(a) shows. An additional one-bit flag is enough to indicate regrouping modes (RGM). For the large groups 1–4, there are 15 regrouping modes, shown in Fig. 5(b). A three-bit flag is required for RG0, while four bits are required for other regrouping modes, as shown in Table III.

C. Semi-Fixed Length Coding

For the entropy coding part, semi-fixed length coding proposed in [8] is used. Modified maximum (mm) is first defined as the maximum absolute value of residuals inside one group. When the absolute values of minimum and maximum residuals are equal, mm is added by 1. Then a coding mode (M) is decided according to semi-fixed-length code table shown in Fig. 6. Based on M, residuals are coded (to be D). The represented range for M between one and six is $[-2^{M-1}+1, 2^{M-1}]$ or $[-2^{M-1}, 2^{M-1}-1]$. Therefore, if any of the residuals equal to -2^{M-1} or 2^{M-1}, a trailing bit (T) is added to denote the sign of the specific residual(s).

Semi-fixed-length Code Table

mm		0	1	2	3~4	5~8	9~16	17~32	>=33
M(mode)		0	1	2	3	4	5	6	7
D (Coding)	0		0	00	000	0000	00000	000000	Original 8-bit value
	±1		1	S1	SS1	SSS1	SSSS1	SSSSS1	
	±2			10	S10	SS10	SSS10	SSSS10	
	±3				S$\bar{S}$1	SS$\bar{S}$1	SSS$\bar{S}$1	SSSS$\bar{S}$1	
	±4				100	S100	SS100	SSS100	
	±5					S$\bar{S}$01	SS$\bar{S}$01	SSS$\bar{S}$01	
	±6					S$\bar{S}$10	SS$\bar{S}$10	SSS$\bar{S}$10	
	±7					S$\bar{S}\bar{S}$1	SS$\bar{S}\bar{S}$1	SSS$\bar{S}\bar{S}$1	
	±8					1000	S1000	SS1000	
	...								
	±16						10000	S10000	
	...								
	±32							100000	

"S" is the sign bit of residuals, and "$\bar{S}$" is logic negation of S.
XX: additional trailing bit T is used to indicate the sign bit.

Fig. 6. Semi-fixed-length codes.

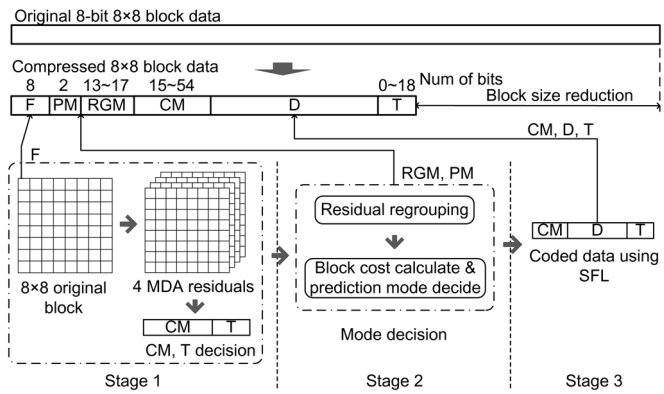


Fig. 7. Overall processing flow of proposed MDA and SFL RFRC.

D. Overall Processing Flow

Fig. 7 shows the overall processing flow of the proposed RFRC scheme. Reference frames are divided into 8×8 blocks. The processing flow can be briefly divided into three stages: residual calculation, prediction mode decision, and coding. In stage 1, four MDA residual blocks are calculated by subtracting the predicted value of the MDA from the original pixel value. The top-left pixel retains its original 8-bit value (F). Then the coding mode (CM) and trailing bit (T) for small size groups are calculated. In mode prediction decision stage 2, after small groups with the same CM are regrouped, the bit costs of all modes are evaluated. The one with the minimum cost is chosen to be the prediction mode (PM). In stage 3, all the residuals calculated by the PM and grouped by RGM are coded with a semi-fixed-length code. After SFL coding, all data, including F, PM, RGM, CM, D, and T, are merged into a bit stream.

E. Reference Frame Recompression for the Chroma Block

In previous work, the reference frame recompression algorithm for the chroma block is usually the same as for the luma block. Most methods do not detail the RFRC of the chroma block. However, the features of the luma and chroma blocks are quite different. In most cases, the chroma block is smoother than the luma block. Therefore, based on this feature, even more representing bits can be saved for the chroma block.

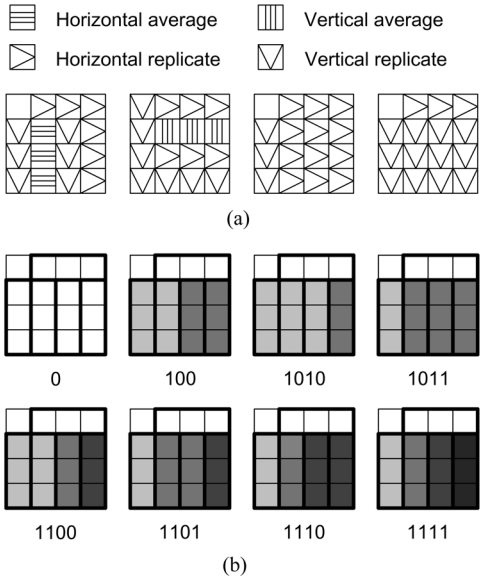


Fig. 8. Prediction and residual regrouping method for the chroma block: (a) four prediction modes and (b) eight residual regrouping modes.

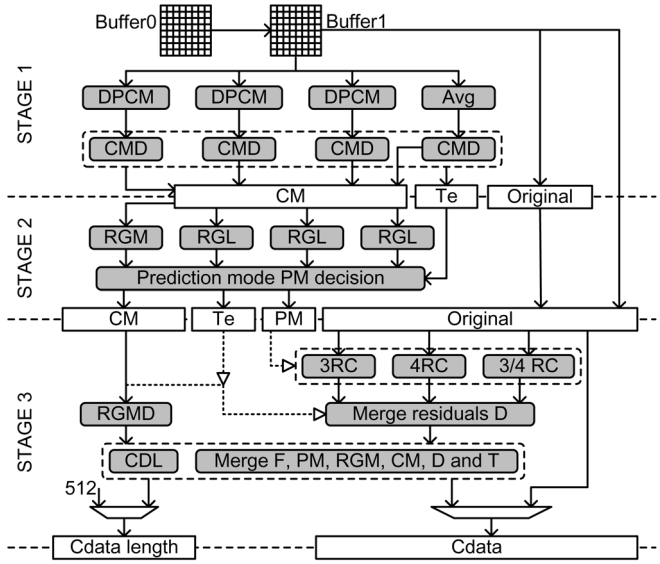


Fig. 9. Three-stage pipeline architecture of MDA and SFL compressor, where Avg denotes "averaging prediction," CMD denotes "coding mode decision," RGM/RGL denotes "M/L size group regrouping," 3/4RC denotes "3 or 4 residual calculation," RGMD denotes "regrouping mode decision," and CDL denotes "compressed data length calculation".

For the prediction stage, the chroma block uses the same multi-mode DPCM and averaging prediction method, as shown in Fig. 8(a), which is just the smaller size of the luma block.

For the residual grouping before SFL coding, we group the residuals predicted by the same technique first. There are five groups in each 4×4 chroma block and in each group there are three residuals, which is the basic group size. In order to further reduce the overhead bits of representing coding mode, we combine the basic groups with the same CM within these five groups, shown as Fig. 8(b). The groups with the same CM are marked with the same color. Since the chroma block is usually smooth, some complicated but useless regrouping modes are also removed. The flags of these eight regrouping modes are shown in Fig. 8(b). One to four bits are required for each regrouping mode.

After the residual regrouping, which is different from the method for luma block, the residuals within one group are represented by same-length bits using semi-fixed-length coding.

III. Hardware Implementation

As an additional layer between the codec core and DRAM controller, reference frame recompression has two separate processes: to compress the reference frame data after the de-blocking filter and to decompress them before motion compensation (see Fig. 1). RFRC compressor and decompressor work in parallel with the core part of the encoder/decoder. So basically RFRC will not decrease encoding/decoding speed as long as the RFRC compressor and decompressor deliver enough throughput to process the data traffic between video encoder/decoder and the memory interface.

Although some parts of the compressor and decompressor can be reused to reduce the hardware cost, this may lead to extra latency and low overall throughput due to the frequent conversion between the compressing and decompressing processes. Hence, in order to realize high resolution and real-time video processing, the proposed hardware consists of a compressor and a decompressor, that is to say, there are no parts shared between these two processes.

A. MDA Compressor

In order to reduce the complexity caused by multi-mode prediction, a reordered two-step hardware architecture is designed that includes advance prediction mode decision and coding. In the prediction mode decision part, four prediction modes are evaluated. However, to do this evaluation, only the number of coded bits is needed. This can be obtained from a relatively low-complexity calculation by checking the maximum and minimum values of the residuals in a group. After the prediction mode has been decided, the highly complex coding step only needs to process one selected prediction mode. The three-stage pipelined architecture is shown in Fig. 9. The first two stages are designed for advance prediction mode decision, while the original pixels are coded in stage 3.

In stage 1, all residuals for the four prediction modes are calculated before the coding mode decision. By reusing some residuals with the same prediction technique and value, the number of residuals that need to be computed reduces from 252 to 168. Moreover, a simplified coding mode decision algorithm can further decrease the computational complexity, where only a few logic gates are used instead of many relatively complex comparators. The details of the residual reuse and simplified coding mode decision algorithms are shown below.

Residual reuse: Since all pixels in one middle (M) group have the same prediction technique, we consider reusing residuals among the M groups in different prediction modes. The M groups division of prediction modes 0 and 3 are shown in Fig. 10(a), while group division for prediction modes 1 and 2 are shown in Fig. 10(b). There are a total of 36 M groups for these four prediction modes. It is obvious that all the prediction techniques for group0 and group1 are the same DPCM scanning, hence the residual calculation of six M groups can be saved. In

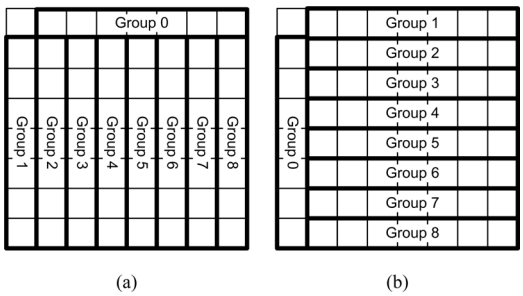


Fig. 10. M groups division method (a) for PMode0 and PMode3 and (b) for PMode1 and PMode2.

addition, the predicted values of group3, group5, and group7 can also be reused between PMode0 and PMode3, and the same goes for PMode1 and PMode2. Thus, only 24 M groups' residuals need to be computed. By calculating the residuals based on residual reuse, the computational complexity can be reduced by 33%.

Simplified coding mode decision (SCMD): The coding mode is decided based on the range of residuals within one small group. It is generally calculated by comparing the maximum and minimum values in one group with 13 thresholds that are the boundary values of the range of representation. To determine the CM, at least thirteen 9-bit comparators are required to compare with the thresholds. In addition, an extra four or six 9-bit comparators are needed for calculating the maximum and minimum residual values in one S group (three or four residuals per S group). To ensure high throughput, eight similar coding mode decision (CMD) units are required. More than 140 9-bit comparators are required for the CMD part alone.

To reduce the computational complexity and decrease the hardware cost greatly, a simplified scheme to decide the coding mode is proposed. For SFL coding, there are two ranges of representation $[-2^{M-1}+1, 2^{M-1}]$ or $[-2^{M-1}, 2^{M-1}-1]$ for each coding mode (from 1 to 6). That is to say, if all residuals or negative residuals in one group are within the range $[-2^{M-1}, 2^{M-1}-1]$, then the CM is equal to M. Hence, we can consider the CM of residuals and negative residuals separately, and the smaller one determines the CM. The complement is used to present residuals in this hardware design. The features of represented residuals in each CM are shown in Table IV. According to the characteristic of presented residuals, CM can be decided by checking the number of identical leading bits (N_{ILB}) that corresponds to the number of leading 1 s when the sign bit is 1 or the number of leading 0 s when the sign bit is 0. Complex comparators are not needed because the CM can be determined by simply checking N_{ILB}. Synthesis results show that by using the simplified CMD saves 70% of the hardware cost compared to a general comparison method.

The positive and negative residuals in one S group are input to the simplified coding mode decision part separately. If these two coding modes are different, the smaller one is selected as the CM, and a trailing bit (T) should be added in the SFL coding (the trailing bit enable $\mathrm{Te} = 1$).

In stage 2 of the MDA compressor, the same CMs of S groups in M group0 or L group1–4 (as shown in Figs. 3(a)–(d)) can be combined to further compress the representing bits of the CM.

TABLE IV
FEATURES OF PRESENTED RESIDUALS IN EACH CODING MODE

CM	0	1	2	3
Range	0	[-1, 0]	[-2, 1]	[-4, 3]
Residual	000000000	000000000 111111111	00000000x 11111111x	0000000xx 1111111xx
$N_{ILB}^{1)}$	9	9	8	7
CM	4	5	6	7
Range	[-8, 7]	[-16, 15]	[-32, 31]	[-255, 255]
Residual	000000xxx 111111xxx	00000xxxx 11111xxxx	0000xxxxx 1111xxxxx	Others
$N_{ILB}^{1)}$	6	5	4	1∼3

1) $N_{ILB}^{1)}$: the number of identical leading bits, starting from the sign bit.
2) x: the bit can be 0 or 1.

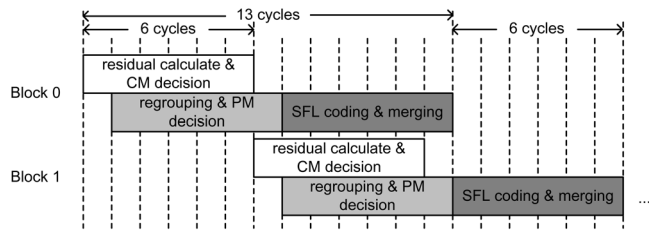


Fig. 11. Pipeline stage of MDA and SFL compressor.

According to the coding modes of all S groups, the residual regrouping modes (RGM) are selected and the number of flag bits to indicate RGM fixed. After the coding mode, trailing bit enable, and regrouping mode are decided, the bit cost of four prediction modes can be evaluated. The mode with the minimum bit cost is chosen as the prediction mode (PM).

In stage 3 of coding, based on the selected prediction mode, residuals are calculated again. For different PMs, most operations can be reused. Then the coded CM-bit residuals (if CM=7, residuals are represented by the original 8 bits) in each group are merged (to be D) together. For one S group, if the trailing bit enable is set, that is to say, at least one of the residuals is equal to -2^{M-1} or 2^{M-1}, a trailing bit T will be added to denote the sign of the specific residual(s). After the regrouping mode is decided, F, PM, RGM, CM, D, and T are merged to form the compressed representation of the 8×8 block. However, if the compressed data length is longer than the original 512 bits, the block is presented by its original 8-bit/sample representation.

Although multi-mode prediction is used in MDA RFRC, the increase in hardware cost is low, less than proportional to the number of prediction modes due to the reordered two-step architecture and the two optimizations, residual reuse and simplified coding mode decision. Only six cycles are required for each 8×8 block compression, as shown in Fig. 11. Based on this three-stage pipeline architecture for the compressor, a throughput of 10.7 samples/cycle is achieved.

B. MDA Decompressor

For the MDA RFRC decompressor, the coded block should be separated into F, PM, RGM, CM, D, and T before splitting and decoding into independent residuals. In each coded block, the bit lengths of F and PM are fixed. The length of RGM can be found by checking 000. Then the length of CM is fixed from the

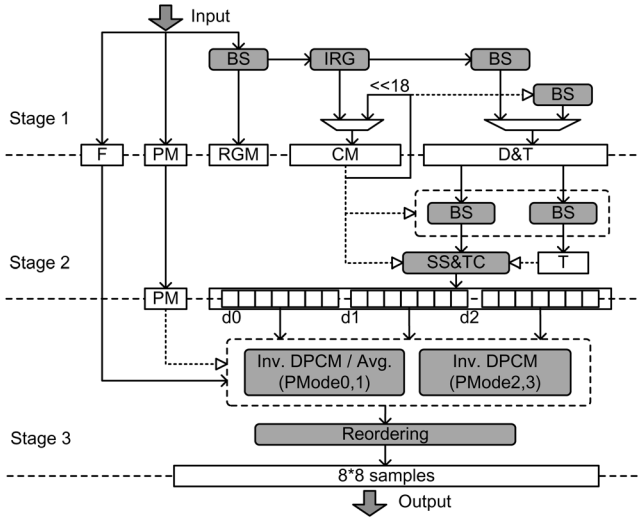


Fig. 12. Three-stage pipeline architecture of MDA and SFL decompressor, where BS denotes "barrel shifter," IRG denotes "inverse regrouping," and SS&TC denotes "sub-block splitting and trailing bit compensation."

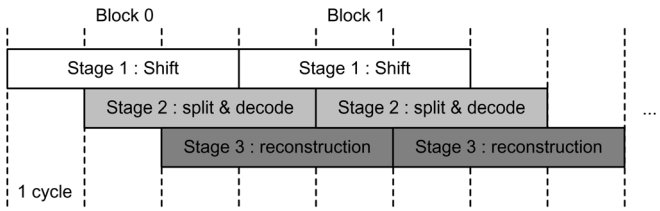


Fig. 13. Pipeline stage of MDA and SFL RFRC decompressor.

given RGM, while the length of D for each group is determined by the fixed CM. The length of T can be derived by subtracting the compressed block length from the sum of F, PM, RGM, CM, and D. Therefore, the order to separate the coded block is: F, PM, RGM, CM, D, and finally T.

The three-stage pipeline architecture of the MDA decompressor is shown in Fig. 12. In stage 1, the input data is shifted to five registers, including F (DPCM start point), PM (prediction mode), RGM (regrouping mode), CM (SFL coding mode), and D&T (coded residuals and trailing bit). The coding mode (CM) for each S group is decoded according to the regrouping mode (RGM). In stage 2, after the coded residuals are separated into M groups by barrel shifters (BS), they are further split and decoded to independent residuals. Meanwhile, trailing bits T are separated from D&T and used to determine the signs of the residuals after decoding. Finally, in stage 3, samples are reconstructed by inverse DPCM scanning or averaging.

In the MDA decompressor, only one prediction mode needs to be processed and most of the hardware can be reused for the different prediction modes. Hence, the computational complexity of the MDA decompressor does not increase much compared to previous work [8]. In this proposed MDA decompressor architecture, only three cycles are required for decoding one 8×8 block, as shown in Fig. 13. The throughput is up to 21.3 samples/cycle.

C. Performance Analysis

Based on the designed architecture, the throughput of MDA compressor reaches 3.2 GSamples/s at 300 MHz, while MDA decompressor throughput is up to 6.4 GSamples/s. The de-signed RFRC compressor and decompressor can easily support both a $3840 \times 2160@60\text{fps}$ encoder and decoder, which have not been achieved by previous works.

For the RFRC compressor, the reference frames from encoder/decoder core are written into external memory only once. Therefore, the throughput requirement for an RFRC compressor to support a $3840 \times 2160@60\text{fps}$ codec is at least 747 Msamples/s under 4:2:0 sampling. This throughput requirement can easily be met by designed RFRC compressor architecture with a throughput of 3.2 Gsamples/s at 300 MHz.

For the RFRC decompressor, the evaluation of cache profiling has been presented in previous work [26][27], where the equivalent frames loaded from external memory when doing motion estimation/disparity estimation on a reference frame for encoder was investigated. From their experimental results, the bandwidth requirement for reference frame reading is, on average, about six times that of the frame size. Hence, with a throughput of 6.4 Gsamples/s (i.e., 4.3 Gpixels/s under 4:2:0 sampling), the proposed RFRC decompressor architecture is able to support a $3840 \times 2160@60\text{fps}$ video encoder at 300 MHz. Since the required bandwidth for a video decoder is much less than an encoder, the 4 K UHDTV video decoder can be easily supported.

Although the throughput and compression efficiency of designed RFRC architecture is improved, the important factors of both video encoder and decoder won't be influenced, especially the encoding/decoding speed. While the proposed RFRC does introduce additional complexity on the basis of core encoder/decoder, the complexity increase is reflected as the hardware cost increase of RFRC compressor and decompressor components (see Table IX), instead of the degradation of encoding/decoding speed. In addition, the proposed MDA&SFL RFRC based on image grouping/division is restricted in 8×8 blocks, so it doesn't influence parallelization of the video encoder/decoder. In the following, the parallelization techniques are explained from the lower and higher classes.

The first class of parallelization is on lower levels such as the pixel and sample levels. However, such parallelization is only needed internally in the core part of the encoder/decoder. When frame data are written into or read from the external memory, they are still transmitted block by block, MB by MB, or CTB by CTB. Therefore the division inside 8×8 blocks in RFRC compressing and decompressing does not complicate the internal parallelism of the core encoder/decoder.

The second class of parallelization is on higher levels such as the wavefront, slice, tile and frame levels. Their basic idea is to divide a frame or a sequence into multiple partitions and use multiple replicas of identical hardware to process these partitions simultaneously. In such a case, the basic unit of processing is always an integer multiple of an 8x8 block. Therefore the division inside 8x8 blocks does not influence parallelization either.

IV. Experimental Results

To evaluate the efficiency of proposed MDA algorithm, the algorithm is integrated with reference software HM. All reconstructed frames for 18 test sequences in five classes are coded in this experiment. The configuration of low delay coding main

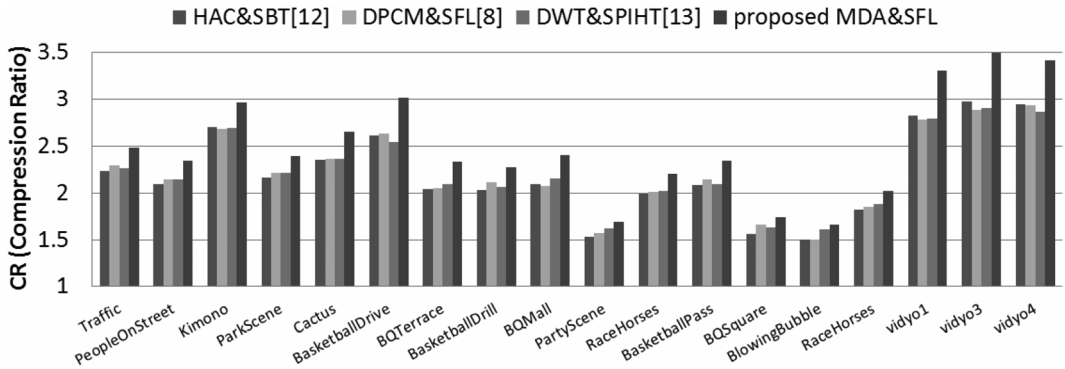


Fig. 14. Compression ratio comparison of the proposed and previous methods.

mode is used. Frame sequence type is IPPP. Quantization parameters (Qp) are 22, 27, 32, and 37.

Together with the proposed algorithm, the performance of three previous works are also simulated for comparison, which are HAC prediction with significant bit truncation (SBT) coding [12], DPCM scanning with SFL coding [8], and DWT-based set partitioning in hierarchical trees (SPIHT) [13]. Since compression performance is usually better for larger block sizes, the basic block sizes of all RFRC algorithms are set to 8×8 for fair comparison.

Given the motion information of encoded video, the required memory bandwidth for frame reading and writing is proportional to the data size of the compressed reference frame. Data reduction ratio (DRR) therefore indicates the achievable ratio of memory bandwidth reduction for frame data access, which is the percent of reduced data size compared to the original data size [see Eq. (1)]. The compression ratio (CR) is the original data size divided by the compressed data size [see Eq. (2)], which can actually be derived from DRR.

$$\text{DRR} = \left(1 - \frac{\text{Compressed data size}}{\text{Original data size}}\right) \times 100\% \quad (1)$$

$$\text{CR} = \frac{\text{Original data size}}{\text{Compressed data size}} \quad (2)$$

Since only RFRC schemes for the luma block are clearly described in previous work, we compare CR results for four Qp and 18 test sequences with all frames for the luma blocks among the proposed and previous methods [8]–[13] (see Fig. 14). The proposed MDA&SFL RFRC outperforms other methods in every case. The CR of the proposed method is further improved by 8.9–19.5%, compared to previous work.

The average CR and DRR of 18 test sequences for luma blocks is shown in Table V. The proposed RFRC scheme performs lossless compression and decompression, which makes it transparent to the core part of the video encoder/decoder, so the encoded image quality and compression ratio will not be influenced. Hence, the average CR of MDA&SFL can achieve as much as 2.49 for the luma block alone with no quality degradation and no bitrate increment. Compared to the previous HAC&SBT algorithm, the proposed scheme can further improve CR by 13.2% on average.

The average CR and DRR of the proposed method under 4:2:0 sampling are shown in Table VI. In addition, a detailed DRR comparison of 18 test sequences between HAC&SBT and the proposed MDA&SFL is presented in Table VII for luma only. In addition, Table VIII shows the CR and DRR comparisons for $4\text{K} \times 2\text{K}$ sequences.

TABLE V
AVERAGE DRR AND CR COMPARISON

Qp	HAC&SBT [12]		DPCM&SFL [8]		DWT&SPIHT [13]		**MDA&SFL Proposed**	
	DRR %	CR	DRR %	CR	DRR %	CR	DRR %	CR
22	48.87	2.06	49.78	2.09	50.15	2.09	**54.22**	**2.33**
27	51.69	2.17	52.51	2.20	52.76	2.20	**57.16**	**2.47**
32	53.65	2.25	54.24	2.27	54.43	2.27	**58.88**	**2.55**
37	55.22	2.31	55.49	2.31	55.76	2.31	**60.12**	**2.60**
Avg	52.36	2.20	53.01	2.22	53.28	2.22	**57.60**	**2.49**

TABLE VI
AVERAGE DRR AND CR OF PROPOSED MDA AND SFL RFRC OF LUMA ONLY FOR A 4:2:0 SAMPLING BLOCK

Qp	DRR %				CR			
	22	27	32	37	22	27	32	37
Luma	54.22	57.16	58.88	60.12	2.33	2.47	2.55	2.60
4:2:0	59.02	61.41	62.94	64.17	2.58	2.73	2.82	2.89

The detailed hardware implementation results of the proposed MDA&SFL architecture and simulation environment are shown in Table IX. In previous work [13], since DWT is used to convert the image into various sub-frequency bands before the SPIHT is adopted as the entropy coding bit-plane by bit-plane, the processing speed is limited by its complex transform and SPIHT coding.

Compared with HAC&SBT [12], which is not able to compress and decompress at the same time, the separate design of the compressor and decompressor is able to fit the characteristic of video codec better. Since the architecture of HAC&SBT cannot compress and decompress reference frames at the same time, the efficiency of TPUA cannot be compared directly. However, no matter whether this architecture is taken as compressor or decompressor, the TPUA of the proposed architecture will always be better than the previous HAC&SBT.

The previous work DPCM&SFL [8] consists of a compressor and a decompressor. Although the complexity of this proposed method is increased by its multi-mode prediction, by applying a reordered two-step architecture and two optimizations, residual reuse and simplified coding mode decision, the efficiency of TPUA is similar to previous work [8]. With an acceptable increase in gate cost, the proposed RFRC architecture is able to

TABLE VII
DATA REDUCTION RATIO COMPARISON OF HAC AND SBT [12] AND PROPOSED MDA AND SFL (ALL 18 HEVC TEST SEQUENCES ARE CODED. BLOCK SIZE IS 8×8 AND ONLY LUMA BLOCKS ARE COMPRESSED.)

Class	Test sequences	DRR % Qp = 22			DRR % Qp = 27			DRR % Qp = 32			DRR % Qp = 37		
		HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**
A	Traffic	53.15	54.93	**57.98**	54.81	56.31	**59.54**	56.03	57.18	**60.49**	56.35	56.98	**60.41**
	PeopleOnStreet	49.52	50.49	**54.70**	51.89	53.13	**57.22**	53.06	54.39	**58.24**	53.99	54.94	**58.90**
B	Kimono	62.27	62.00	**65.59**	63.33	63.03	**66.56**	63.51	63.02	**66.61**	63.36	62.75	**66.37**
	ParkScene	49.92	51.55	**54.64**	52.97	54.41	**57.74**	55.07	56.14	**59.50**	56.14	56.75	**60.06**
	Cactus	53.43	54.20	**58.21**	57.24	57.53	**62.28**	58.59	58.51	**63.42**	60.09	59.71	**64.66**
	BasketballDrive	58.51	59.28	**63.79**	61.79	62.16	**67.20**	62.74	62.83	**67.80**	63.50	63.26	**68.21**
	BQTerrace	43.52	43.58	**49.22**	50.72	51.03	**57.35**	53.44	53.51	**59.57**	54.97	55.17	**60.47**
C	BasketballDrill	43.73	46.36	**49.34**	48.92	51.17	**54.42**	53.38	55.16	**58.48**	55.29	56.66	**60.01**
	BQMall	50.34	50.39	**56.90**	51.86	51.72	**58.43**	52.74	52.35	**58.95**	53.23	52.58	**58.90**
	PartyScene	29.44	30.90	**35.26**	33.28	34.94	**39.46**	36.29	37.81	**42.54**	39.56	40.61	**45.60**
	RaceHorses	47.01	47.65	**51.84**	48.77	49.47	**53.77**	49.97	50.67	**54.92**	52.46	52.93	**57.29**
D	BasketballPass	48.49	50.64	**53.93**	50.95	52.64	**56.41**	52.83	53.95	**58.03**	55.18	55.65	**60.09**
	BQSquare	28.13	31.86	**34.54**	32.89	36.90	**39.81**	38.37	42.24	**45.22**	42.10	45.63	**48.74**
	BlowingBubble	27.40	27.85	**33.72**	30.63	30.98	**37.29**	34.73	34.70	**41.25**	39.66	38.99	**45.67**
	RaceHorses	40.33	41.03	**45.70**	42.62	43.40	**48.03**	45.88	46.69	**51.18**	50.77	51.25	**55.71**
E	vidyo1	63.85	63.35	**69.65**	64.92	64.41	**70.35**	65.11	64.65	**70.01**	64.90	64.04	**69.28**
	vidyo3	65.86	65.20	**71.30**	66.74	65.83	**72.13**	66.93	65.73	**72.10**	66.01	64.76	**71.04**
	vidyo4	64.83	64.80	**69.68**	66.17	66.08	**70.91**	66.97	66.73	**71.48**	66.47	66.24	**70.84**
Average		48.87	49.78	**54.22**	51.69	52.51	**57.16**	53.65	54.24	**58.88**	55.22	55.49	**60.12**

TABLE VIII
DATA REDUCTION RATIO COMPARISON OF HAC AND SBT [12] AND PROPOSED MDA AND SFL (ALL 18 HEVC TEST SEQUENCES ARE CODED. BLOCK SIZE IS 8×8 AND ONLY LUMA BLOCKS ARE COMPRESSED.)

Test sequences	DRR % Qp = 22			DRR % Qp = 27			DRR % Qp = 32			DRR % Qp = 37		
	HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**	HAC SBT	DPCM SFL	**MDA SFL**
CrowdRun	47.65	47.15	**51.19**	53.46	53.24	**57.08**	54.88	54.68	**58.51**	56.23	55.66	**59.73**
DucksTakeOff	35.26	36.19	**38.63**	53.96	55.48	**57.59**	57.26	58.88	**60.67**	59.09	60.31	**62.31**
InToTree	46.12	46.47	**49.16**	62.37	63.00	**65.68**	68.24	68.60	**71.32**	71.14	70.82	**73.81**
ParkJoy	50.44	50.61	**53.60**	53.64	53.71	**56.84**	54.95	54.79	**58.07**	56.09	55.59	**59.11**

TABLE IX
HARDWARE IMPLEMENTATION RESULT COMPARISON

	HAC&SBT [12]	DWT& SPIHT [13]	DPCM&SFL [8]		**MDA&SFL**	
	Comp. or Decomp.[1)]	Comp. or Decomp.[1)]	Comp.	Decomp.	**Comp.**	**Decomp.**
CR (only Luma)	2.20	2.22	2.22		**2.49**	
DRR (only Luma)	52.36%	53.28%	53.01%		**57.60%**	
Unit	16×8	16×16	8×4		**8×8**	
CMOS tech. (nm)	180	180	90		**90**	
Max. freq. (MHz)	180	10	300		**300**	
Throughput(Gsamples/s)	0.9 as comp./2.6 as decomp.	0.005	1.8	4.8	**3.2**	**6.4**
Throughput(samples/cycle)	5.1 as comp./14.2 as decomp.	0.45	6	16	**10.7**	**21.3**
Gate count (k)	36.1	26.9	18.40	26.41	**45.13**	**34.45**
TPUA[2)] [12] (10^{-5}samples/cycle/gate)	$*$ [3)]	1.7	32.6	60.6	**23.7**	**61.8**

[1)] comp. or decomp.: compressor and decompressor can't be used at the same time.

[2)] TPUA [12]: (throughput/gate count), is the evaluation criterion to consider both hardware cost and throughput.

[3)] $*$: can't compare with the others' TPUA directly.

achieve higher throughput than previous work [8]–[13]. Since the throughput of proposed architecture is enough to process the data traffic between the video encoder/decoder and the memory, both $3840 \times 2160@60\text{fps}$ video encoder and decoder can be supported without compression speed decrease.

V. CONCLUSION

This paper proposes a novel lossless reference frame recompression algorithm to reduce external memory bandwidth. Multi-mode spatial-domain prediction is proposed to combine the merits of DPCM scanning and averaging. Moreover, an efficient residual grouping method further improves the compression efficiency. Experimental results show that compared to no compression, the DRR of the proposed RFRC is approximately 57.6% on average without quality degradation. By applying several optimizations, the designed architecture is able to support both a $3840 \times 2160@60\text{fps}$ video encoder and decoder at 300 MHz with an acceptable hardware cost. Meanwhile, the proposed lossless RFRC can be easily extended to lossy RFRC, such as in the previous work [21] that changes the SFL coding. In future, we will focus on an efficient scheme to extend this proposal into lossy reference frame recompression.

REFERENCES

[1] M. Budagavi and M. Zhou, "Video coding using compressed reference frames," in *Proc. ICASSP*, 2008, pp. 1165–1168.

[2] T.-C. Chen, C.-Y. Tsai, Y.-W. Huang, and L.-G. Chen, "Single reference frame multiple current macroblocks scheme for multiple reference frame motion estimation in H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 17, no. 2, pp. 242–247, Feb. 2007.

[3] C.-Y. Chen, S.-Y. Chien, Y.-W. Huang, T.-C. Chen, T.-C. Wang, and L.-G. Chen, "Analysis and architecture design of variable block-size motion estimation for H.264/AVC," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 53, no. 3, pp. 578–593, Mar. 2006.

[4] Y.-K. Lin, C.-C. Lin, T.-Y. Kuo, and T.-S. Chang, "A hardware-efficient H.264/AVC motion-estimation design for high-definition video," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, no. 6, pp. 1526–1535, Jul. 2008.

[5] X. Chen, P. Liu, J. Zhu, D. Zhou, and S. Goto, "Block pipelining cache for motion compensation in high definition H.264/AVC video decoder," in *Proc. ISCAS*, 2009, pp. 1069–1072.

[6] P. Chao and Y.-L. Lin, "Reference frame access optimization for ultra high resolution H.264/AVC decoding," in *Proc. ICME*, 2008, pp. 1441–1444.

[7] J. Zhu, P. Liu, and D. Zhou, "An SDRAM controller optimized for high definition video coding application," in *Proc. ISCAS*, 2008, pp. 3518–3521.

[8] D. Zhou, J. Zhou, X. He, J. Zhu, J. Kong, P. Liu, and S. Goto, "A 530 Mpixels/s $4096 \times 2160@60\mathrm{fps}$ H.264/AVC high profile video decoder chip," *IEEE J. Solid-State Circuits*, vol. 6, no. 4, pp. 777–788, Apr. 2011.

[9] D. Zhou, J. Zhou, J. Zhu, P. Liu, and S. Goto, "A 2 Gpixel/s H.264/AVC HP/MVC video decoder chip for super hi-vision and 3DTV/FTV applications," in *Proc. ISSCC*, 2012, pp. 224–226.

[10] 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.

[11] S. Lee, M.-K. Chung, and C.-M. Kyung, "Low latency variable length coding scheme for frame memory recompression," in *Proc. ICME*, 2010, pp. 232–237.

[12] 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.

[13] C.-C. Cheng, P.-C. Tseng, and L.-G. Chen, "Multimode embedded compression codec engine for power-aware video coding system," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 2, pp. 141–150, Feb. 2009.

[14] Y. Lee and T. Tsai, "An efficient embedded compression algorithm using adjusted binary code method," in *Proc. ISCAS*, 2008, pp. 2586–2589.

[15] Y.-Y. Lee, Y.-H. Lee, and T.-H. Tsai, "An efficient lossless embedded compression engine using compacted-FELICS algorithm," in *Proc. SOCC*, 2008, pp. 233–236.

[16] T. Song and T. Shimamoto, "Reference frame data compression method for H.264/AVC," *IEICE Electron. Exp.*, vol. 4, no. 3, pp. 121–126, Feb. 2007.

[17] T. Yng, B. 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.

[18] Y. Li, W. Wang, and G. Zhang, "Hybrid pixel encoding: An effective display frame compression algorithm for HD video decoder," in *Proc. ICCSE*, 2012, pp. 303–309.

[19] H.-T. Yang, J.-W. Chen, H.-C. Kuo, and Y.-L. Lin, "An effective dictionary-based display frame compressor," in *Proc. ESTIMedia*, 2009, pp. 28–34.

[20] 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.

[21] L. Song, D. Zhou, X. Jin, and S. Goto, "A constant rate bandwidth reduction architecture with adaptive compression mode decision for video decoding," in *Proc. EUSIPCO*, 2010, pp. 2017–2021.

[22] Y. V. Ivanov and D. Moloney, "Reference frame compression using embedded reconstruction patterns for H.264/AVC decoder," in *Proc. ICDT*, 2008, pp. 168–173.

[23] Y.-H. Lee, Y.-C. Chen, and T.-H. Tsai, "A bandwidth-efficient embedded compression algorithm using two-level rate control scheme for video coding system," in *Proc. ISCAS*, 2010, pp. 1149–1152.

[24] Y.-H. Lee, Y.-Y. Lee, H.-Z. Lin, and T.-H. Tsai, "A high-speed lossless embedded compression codec for high-end LCD applications," in *Proc. A-SSCC*, 2008, pp. 185–188.

[25] L. Guo, D. Zhou, and S. Goto, "Lossless embedded compression using multi-mode DPCM & averaging prediction for HEVC-like video codec," in *Proc. EUSIPCO*, 2013, pp. 1–5.

[26] L.-F. Ding, W.-Y. Chen, P.-K. Tsung, T.-D. Chuang, P.-H. Hsiao, Y.-H. Chen, H.-K. Chiu, S.-Y. Chien, and L.-G. Chen, "A 212 Mpixels/s $4096 \times 2160\mathrm{p}$ multiview video encoder chip for 3D/Quad full HDTV applications," *IEEE J. Solid-State Circuits*, vol. 45, no. 1, pp. 46–58, Jan. 2010.

[27] J. Zhou, D. Zhou, G. He, and S. Goto, "A 1.59 Gpixel/s motion estimation processor with -211-to-211 search range for UHDTV video encoder," in *Proc. VLSIC*, 2013, pp. C286–C287.

[28] Xiph.org Foundation, "Xiph.org Video Test Media $4\mathrm{k} \times 2\mathrm{k}$ test sequences," [Online]. Available: http://media.xiph.org/video/derf/

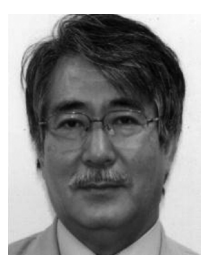
Li Guo received the B.E. degree from Shanghai Jiao Tong University, Shanghai, China, in 2012, and the M.E. degree in engineering from both Waseda University, Kitakyushu, Japan, and Shanghai Jiao Tong University, Shanghai, China, in 2013, through a double-degree program. She is currently working toward the M.E. degree at Shanghai Jiao Tong University, Shanghai, China.

Her research interests are in algorithm and VLSI architectures for video coding.

Dajiang Zhou (S'08–M'10) received B.E. and M.E. degrees from Shanghai Jiao Tong University, Shanghai, China, and the Ph.D. degree in engineering from Waseda University, Kitakyushu, Japan, in 2010.

He is currently an Assistant Professor with the Graduate School of Information, Production and Systems, Waseda University, Kitakyushu, Japan. His research interests are algorithms and implementations for multimedia and communications signal processing, especially in low-power high-performance VLSI architectures for video codecs, including H.265/HEVC and H.264/AVC.

Dr. Zhou received a number of awards including the Best Student Paper Award of VLSI Circuits Symposium 2010, the International Low Power Design Contest Award of ACM ISLPED 2010, the 2013 Kenjiro Takayanagi Young Researcher Award, and the Chinese Government Award for Excellent Students Abroad of 2010. He was a recipient of the research fellowship from the Japan Society for the Promotion of Science from 2009 through 2011. His work on an 8 K UHDTV video decoder VLSI chip was granted the 2012 Semiconductor of the Year Award from Japan.



Satoshi Goto (S'69–M'77–SM'84–F'86–LF'11) received the B.E. and M.E. degrees in electronics and communication engineering from Waseda University, Kitakyushu, Japan, in 1968 and 1970, respectively, and the Dr. of engineering from Waseda University, Kitakyushu, Japan, in 1978.

He joined NEC Laboratories in 1970 where he worked with LSI Design, Multimedia Systems and Software as GM and Vice President. Since 2002, he has been a Professor with the Graduate School of Information, Production and Systems of Waseda University, Kitakyushu, Japan. He is a Visiting Professor with Shanghai Jiao Tang University, Shanghai, China, and Tsinghua University of China, Beijing, China.

Dr. Goto served as GC of ICCAD, ASPDAC, VLSI-SOC, ASICON, and ISOCC, and is a Member of the Science Council of Japan. He was a Board Member of the IEEE Circuits and Systems Society. He is an IEICE Fellow.