BLOCK-BASED ADAPTIVE BIT ALLOCATION FOR REFENCE MEMORY REDUCTION

Seanae Park, Jung hak Nam*, Donggy Sim*, Young Hun Joo**, Yong Serk Kim**, Hyun Mun Kim***

* Department of computer engineering
Kwangwoon Univ., Seoul, Korea
** Samsung Electronics
E-mail: dgsim@kw.ac.kr

ABSTRACT

In this paper, we propose an effective memory reduction algorithm to reduce the amount of reference frame buffer and memory bandwidth in video encoder and decoder. In general video codecs, decoded previous frames should be stored and referred to reduce temporal redundancy. Recently, reference frames are recompressed for memory efficiency and bandwidth reduction between a main processor and external memory. However, these algorithms could hurt coding efficiency. Several algorithms have been proposed to reduce the amount of reference memory with minimum quality degradation. They still suffer from quality degradation with fixed-bit allocation. In this paper, we propose an adaptive block-based min-max quantization that considers local characteristics of image. In the proposed algorithm, basic process unit is 8x8 for memory alignment and apply an adaptive quantization to each 4x4 block for minimizing quality degradation. We found that the proposed algorithm could improve approximately 37.5% in coding efficiency, compared with an existing memory reduction algorithm, at the same memory reduction rate.

Keywords: memory reduction, bandwidth reduction, recompression video codec, H.264/AVC

1. INTRODUCTION

Recently, handheld consumer electronic devices such as camera phones, digital still cameras, digital camcorders, and personal media players have been widely deployed. Operating time with a small battery is a key factor in success of these consumer devices. Because of the power limitation, they have a light processor and a minimum amount of memory. Based on these requirements in consumer business side, video standardizations have been done to develop low-complexity and high-efficiency decoder. However, encoder complexity was not carefully considered in standardization. These days, two-way video communication and low-complexity encoder applications are widely commercialized. With current video standards, it is not easy to achieve the best coding efficiency with limited power.

In general, computational complexity and the number of memory operation are proportional to power consumption [1][2]. This leads us to develop a video codec to minimize computational complexity and memory load for low-power applications. H.264/AVC supports multiple reference frames, multiple block modes, and de-blocking filter for coding efficiency. These methods can increase coding efficiency but also requires high computational complexity and a large number of memory operations. These days, handheld devices also support high-resolution video, so that memory operation and bandwidth become critical issues. To cope with these problems, several algorithms have been proposed to reduce an amount of reference memory, resulting that required bandwidth could be reduced between a main processor and external memory. However, they employ a fixed-length compression for all the blocks. However, natural images have different characteristics in each local area. While blocks with lower dynamic range could be coded without degradation, blocks with higher dynamic range could be severely degraded. We need to consider the block-by-block characteristics in compressing the reference memory to reduce quality degradation with minimizing the amount of reference memory.

In this paper, we propose a memory reduction algorithm to reduce an amount of memory and bandwidth for low-power and low-complexity codec. A recompression codec for memory bandwidth has been studied and it compresses reference frames before storing them into DPB and they should be reconstructed before motion estimation/compensation. This recompression algorithm must have low computational complexity and also low quality degradation. In this paper, we propose a recompression algorithm that is designed to minimize quality-degradation by considering local dynamic ranges.

In Section 2, several conventional recompression algorithms will be introduced. The proposed block-based adaptive min-max quantization will be presented in Section 3. Experimental results and conclusion will be given in Section 4 and 5, respectively.

2. CONVENTIONAL VIDEO RECOMPREATION ALGORITHM

To reduce the amount of memory and bandwidth, several algorithms have been proposed. DPCM-based memory reduction algorithm was presented by using pixel-wise prediction [3]. This approach is simple but severe quality degradation could occur at high dynamic ranges. A fixed-bit min-max quantizer was proposed to be applied to down-sampled decoded reference frame [4]. On the other hand, a memory reduction method was presented by using the fixed-bit min-max quantizer without down-sampling operation [5].

2.1 DPCM-based reference frame compression

DPCM-based memory reduction was proposed for reference memory recompression [3]. Figure 1 shows the block diagram of the memory reduction decoder.

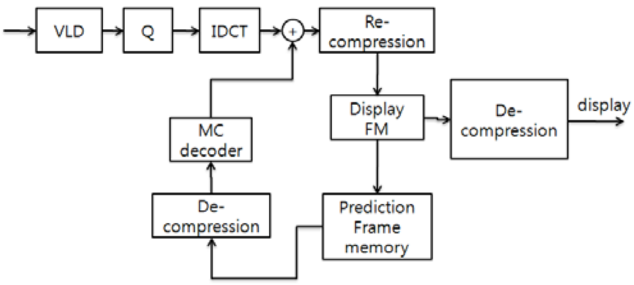

Fig. 1: Decoder block diagram of DPCM-based memory reduction

DPCM is applied to one column in each 8x8 block. Then, the DPCM values are quantized and stored in the reference memory, as shown in Fig. 1. Because the DPCM has low-complexity, this memory reduction codec could be implemented with minimal modification. However, when a difference value between consecutive pixels is large, the quantization error could not be negligible. In this case, the error could propagate into next pixels. This DPCM and quantization can be represented by

$$Y_{n,m} = \begin{cases} X_{n,m} & (m=0) \\ Y_{n,m-1} + Q[X_{n,m} - Y_{n,m-1}] & (m \neq 0) \end{cases} \quad \text{n, m} = 0,1,\cdots,7 \tag{1}$$

where, X is input image and Y is output image to be stored in the reduced reference memory. n and m are row and column index in image domain, respectively. In this method, 4-bit non-linear quantizer is used. When we make use of 4-bit depth at each pixel, we can achieve 43% memory saving with this approach.

2.2 Reference frame recompression algorithm with fixed-bit min-max quantizer for down-sampled reference frames

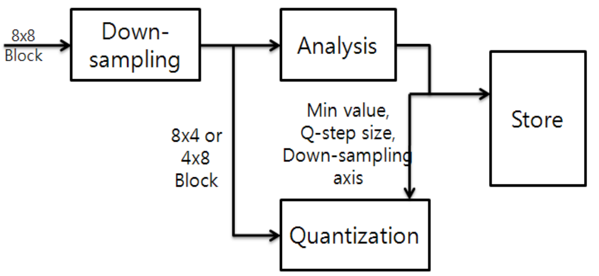

Fig. 2: Block diagram of memory reduction with min-max quantizer and down-sampling

Min-max quantizer was employed to reduce the amount of memory and bandwidth [4]. It is based on down-sampling and min-max quantization for high compression ratio and low-computational complexity. Figure 2 is the block diagram of the fixed-bit min-max quantizer with down-sampling [3].

In this algorithm, an image is divided into 8x8 blocks and each block is down-sampled in horizontal or vertical directions by a factor of two. Then, minimum and maximum values are identified in the down-sampled block. We can quantize this block in 4-bit length. Figure 3 shows bitstream structure for syntax elements for memory reduction. This algorithm makes use of two down-sampling modes and an indicate bit is required. In addition, 8-bit is needed to represent the minimum value and 7-bit is needed for quantization step size. 128 (=8x4x4 or 4x8x4) bits are required for all the pixels in 8x8 block. In total, 144 bits are required for each 8x8 block.

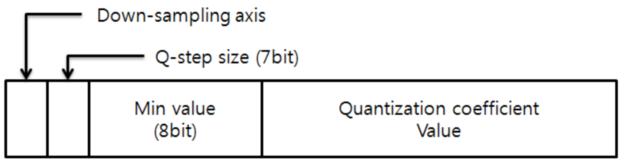

Fig. 3: Bitstream structure in reduced reference memory

2.3 Memory reduction with fixed-bit min-max quantizer

Figure 3 shows the decoder block diagram of a video codec for reducing memory size and memory bandwidth [5]. This conventional algorithm has two key steps. The first is fixed-length compression (FLC) to compress reference frames in order to maintain random access of any block in memory. The second step is to carry out reference frame compression in the core video coding loop to prevent drift error between encoder and decoder because quantization errors is produced during FLC in the residual domain after motion compensation.

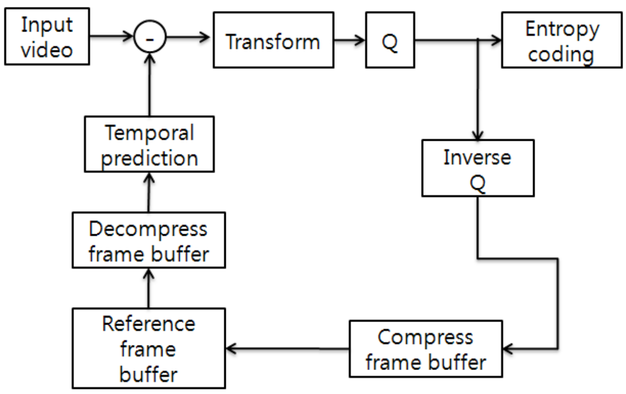

Fig. 4: Block diagram of memory reduction codec with a min-max quantizer

In this memory reduction algorithm, the fixed-length compression technique can be regarded as a block scalar quantization scheme. For each 4x4 pixel block, the minimum and maximum pixel values are obtained and stored. Then, all the pixels in the 4x4 block are uniformly quantized between the minimum and maximum pixel values and the quantized values are stored. This memory reduction algorithm is similar to [3], but it do not use down-sampling step and its basic process unit size is 4x4.

3. PROPOSED BLOCK –BASED ADAPTIVE BIT ALLOCATION OF MIN-MAX QUANTIZER FOR MEMORY REDUCTION

Video coding standards utilize inter frame prediction for coding efficiency and image quality. Reconstructed frames should be stored for motion estimation and compensation in encoder and decoder. We need to load these frames to estimate motion vectors between the current frame and reconstructed previous frames. As image size is larger, the bandwidth between a main processor and external memory becomes larger. It could be a bottleneck in improving encoding speed and reducing power consumption. To alleviate these problems, the reconstructed frames can be recompressed and stored in the decoding buffer. Then, the compressed data in the decoding buffer is decoded and the decoded reconstructed blocks are used for motion estimation and compensation. During the recompression, additional degradation occurs and the degradation should be minimized with minimum computational complexity.

Table 1: Bit ratio to be required for lossless recompression at 4x4 blocks

Sequence	QP	Bits required for lossless-recompression						Total
		0.5bit	1bit	2bit	3bit	4bit	5 bit	
News_100 frame.qcif	22	7.65%	8.57%	9.69%	12.16%	11.88%	11.94%	61.89%
	27	6.64%	9.09%	12.66%	11.41%	11.22%	11.14%	62.16%
	32	9.54%	8.77%	11.67%	12.58%	10.15%	10.74%	63.43%
	37	12.36%	5.95%	10.36%	12.62%	13.56%	10.55%	65.41%
Foreman_ 100frame. qcif	22	1.97%	1.79%	5.96%	13.56%	17.09%	18.04%	58.40%
	27	1.89%	3.54%	8.85%	12.46%	15.48%	17.06%	59.29%
	32	2.05%	4.36%	9.62%	12.72%	15.31%	17.01%	61.07%
	37	1.88%	3.82%	9.53%	14.15%	15.64%	18.27%	63.28%
Football_p 704*480_2 60frame.y uv	22	0.16%	0.64%	2.70%	12.19%	28.14%	27.34%	71.17%
	27	0.88%	2.58%	6.22%	13.44%	25.96%	23.82%	72.90%
	32	3.88%	7.03%	12.05%	14.69%	19.03%	18.82%	75.51%
	37	10.09%	10.45%	17.21%	15.33%	12.85%	12.68%	78.61%
Mobile_p7 04*480_19 0freame.y uv	22	1.28%	0.29%	5.87%	7.86%	9.85%	12.19%	37.34%
	27	1.70%	6.55%	6.85%	8.24%	8.78%	11.30%	43.43%
	32	1.86%	4.15%	7.17%	8.86%	9.16%	11.35%	42.56%
	37	1.89%	3.81%	7.72%	9.75%	10.28%	11.89%	45.33%

In the proposed algorithm, a reference frame is divided into 8x8 blocks and optimal bits are allocated to each 4x4 block in 8x8 blocks. Table 1 shows bits ratio to be required for lossless recompression for every 4x4 blocks. As shown the table, it is possible to recompress 40~80% 4x4 block with less than 8 bits. If we know the optimal bits for each 4x4 block and we can improve coding efficiency by allocating remain bits to other blocks. We propose an adaptive bit allocation algorithm for 4x4 blocks in 8x8 block.

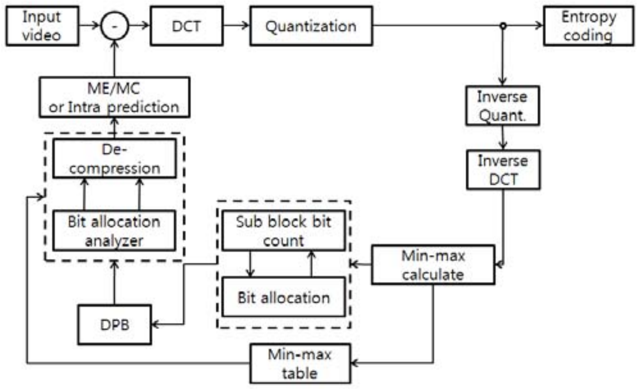

Fig. 5: Encoder block diagram of proposed block based adaptive bit-length allocation algorithm by using min-max quantizer

Figure 5 shows the block diagram of the proposed encoder. At first, input image is separated into 8x8 blocks. One 8x8 block is again divided into four 4x4 blocks. And then, we compute minimum and maximum values in each 4x4 block. These minimum and maximum values are considered as dynamic ranges of each 4x4 block. Next, we can compute bits for lossless recompression with the difference between minimum and maximum values. In the proposed algorithm, each 4x4 block can have variable bits, but total allocated bits for four 4x4 blocks should be fixed. If total allocated bits are not fixed, bite-aligned memory access could not be performed.

For example, if we want to perform five-bit depth recompression, 320 (=8x8x5) bits are assigned into 8x8 block, but each 4x4 block can use adaptive bits. It is desired that the number of bits required for 8x8 block is smaller than 320 bits or same. In this case, we can recompress without any loss. If not, we need to allocate optimal bits for each 4x4 block by removing 1 bit in turn from the first to last 4x4 block until allocated bits is the same to 320 bits. The proposed bit-allocation algorithm is presented as follows.

```
// calculate bits required for each 4x4 block
for(i=0;i<4;i++) bitcount_B[i] = log2(max[i]-min[i])
sum_bitcount = bitcount_B[0]+ bitcount_B[1]+
               bitcount_B[2]+ bitcount_B[3];
i=0;
if (sum_bitcount <= M*4) lossless coding
  else {
       do { bitcount_B[i]--;   i++}
       while (sum_bitcount > M*4)
}
```

The proposed algorithm employs the min-max quantizer.

For each 4x4 block, the minimum and maximum pixel values are calculated and stored. Then, all the pixels in the 4x4 block are uniformly quantized between the minimum and maximum pixel values and stored. Figure 6 shows bit allocation structure for each 8x8 block.

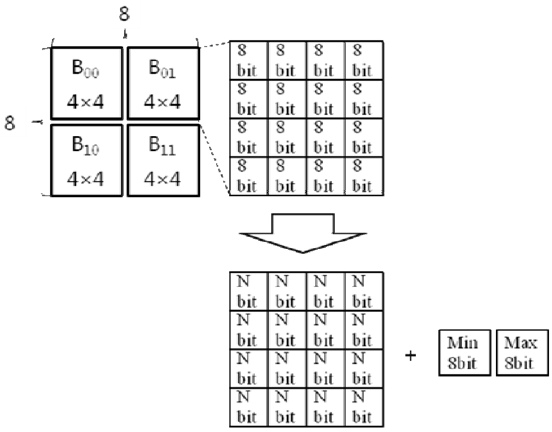

Fig. 6: Bit allocation structure for 8x8 block

In Fig. 6, ‘N bit’ represents allocated bits for 4x4 blocks. The sub-block bit counter in Fig. 7 is to compute required bits for 4x4 blocks. The pseudo code of the block was presented before. Quantization is conducted in the bit-allocation block based on computed required bits.

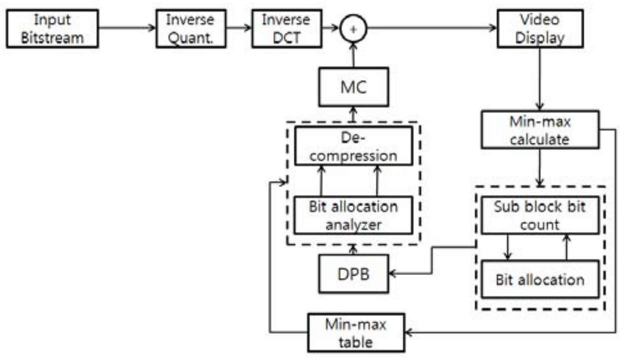

Fig. 7: Decoder block diagram of the proposed block-based adaptive bit allocation algorithm with a min-max quantizer

4. EXPRIMENTAL RESULTS

In this paper, we show the effectiveness of the proposed algorithm with comparative study. We implemented the proposed algorithm on JM 12.4 which is H.264/AVC reference software released by the JVT. All the experiments were conducted on a Pentium® D CPU 3.0 GHz (2.0G RAM). We used eight test sequences. Table 2 shows test sequences and their characteristics. In this experiment, bit depth is five and the GOP structure is IPPP. The other test configuration is based on the baseline profile in VCEG test condition [6].

Table 3 shows the PSNR and bitrate of the proposed algorithm and the conventional memory reduction algorithm with fixed-bit min-max quantizer [5]. In general, the proposed algorithm yields a good performance, compared with the conventional algorithm. In our experimentation, we found that the proposed algorithm can achieve average 1.7% BD-bitrate gain and average 0.03dB BD-PSNR gain. Also, we found that degradation for large size sequences is lower than that of small size sequences for two algorithms. The conventional algorithm also shows a good performance in large size videos. Large size sequences have higher probability of flat region than small size. As shown Table 1, pixel ratio to be coded without any loss for sequence “Football_p704*480” is higher than that for other sequences. Based on this observation, we achieve good coding efficiency for the sequence.

Table 2: Test sequences

Sequence name	Size	Frame rate	# frame
Foreman_300frames.cif	352 x 288	30	300
MissAmerica_300frames.cif	352 x 288	30	300
Silent_300frames.cif	352 x 288	30	300
Suzie_300farmes.cif	352 x 288	30	300
Mobile_300frames.cif	352 x 288	30	300
Football_p704*480_260frame.yuv	704 x 480	30	260
Mobile_p704*480_190freame.yuv	704 x 480	30	190
Suzi_p704*480_260freame.yuv	704 x 480	30	260

Table 3: Performance comparison of the proposed and fixed-bit min-max algorithms [5]

	QP	H.264/AVC		Fixed min-max Q[4]		Average		Propsed method		Average	
		bit-rate	PSNR	bit-rate	PSNR	BD PSNR	BD bitrate	bit-rate	PSNR	BD PSNR	BD bitrate
Foreman_300frames.cif	22	1181.8	40.5	1262	40	-0.47	12.782	1237.8	40.2	-0.35	9.1927
	27	505.35	37	535.6	36.7			528.93	36.8		
	32	210.3	33.7	220	33.4			218.01	33.5		
	37	103.24	30.8	106.2	30.6			105.74	30.6		
MissAmerica_300frames.cif	22	469.4	45	476.7	44.8	-0.14	4.656	473.71	44.9	-0.11	3.7093
	27	177.24	42.2	179.4	42.1			178.81	42.1		
	32	67.23	39.2	67.77	39.1			67.87	39.1		
	37	26.17	36.3	26.64	36.3			26.54	36.2		
Silent_300frames.cif	22	552	40.3	578	40	-0.25	5.8737	571.65	40	-0.19	4.3188
	27	267.46	36.6	274.9	36.4			272.86	36.5		
	32	127.91	33.3	130.3	33.2			129.41	33.2		
	37	63.84	30.5	64.34	30.5			64.32	30.5		
Suzie_300farmes.cif	22	777.04	42.7	789.3	42.5	-0.12	3.4814	784.3	42.6	-0.09	2.4921
	27	357.25	39.5	361.6	39.4			359.56	39.4		
	32	142.44	36.4	144.1	36.4			144.55	36.4		
	37	63.56	33.9	64.34	33.8			63.91	33.8		
Mobile_300frames.cif	22	3673.7	39.1	3859	37.7	-0.76	15.96	3823.2	38	-0.63	12.949
	27	1946.6	34.9	2047	34.2			2027.7	34.3		
	32	842.54	30.5	885.8	30.1			879.69	30.2		
	37	326.44	26.5	342.9	26.3			340.27	26.3		
Football_p704*480_260frame.yuv	22	9195.5	40	9316	39.6	-0.18	3.81%	9294.6	39.7	-0.13	2.71%
	27	5049.9	36.7	5092	36.6			5084.4	36.6		
	32	2617.8	33.4	2629	33.3			2627	33.3		
	37	1357.6	30.6	1361	30.5			1360.4	30.5		
Mobile_p704*480_190freame.yuv	22	19954	39	20202	38	-0.54	11.23%	20158	38.1	-0.4	7.85%
	27	11124	35	11305	34.5			11276	34.6		
	32	5297.5	31	5416	30.7			5396.8	30.7		
	37	2152.4	27.3	2208	27.1			2202.9	27.1		
Suzi_p704*480_260freame.yuv	22	4328.1	41.1	4387	41	-0.14	6.17%	4362.1	41	-0.1	4.47%
	27	1190.8	38.3	1224	38.2			1214.5	38.2		
	32	363.88	35.7	373.5	35.7			371.46	35.7		
	37	158.92	33.4	161.2	33.4			160.45	33.4		
Aveage						-0.14	6.1698			-0.1	4.4702

5. CONCLUSION

In this paper, we propose an effective memory reduction algorithm to reduce the amount of reference frame buffer and memory bandwidth in video encoder and decoder. The proposed algorithm is based on an adaptive block-based min-max quantization algorithm that considers the local characteristic of image. In our experimental results, we found that the proposed algorithm can obtain around 1.7% BD-bitrate gain and 0.03dB BD-PSNR gain, compared with the conventional fixed-bit min-max algorithm with 37.5% memory saving.

ACKNOLWEDGEMENT

This research was conducted with the grand from the Samsung Electronics in 2008.

6. REFERENCES

[1] Sungho Seo, Yoonsik Choe, Yonggoo Kim, and Yungho Choi, "A Complexity Measurement Using a Processor Simulator," ITU Video Coding Experts Group (VCEG) 35th Meeting, July 2008.

[2] M. Budagavi and M. Zhou, "Requirements for next generation video coding standards", ITU-T SG16 meeting, Geneva, Apr. 2006

[3] Hideo Ohira and Fumitoshi karube, "A Memory reduction approach for MPEG decoding system," IEICE Trans. Fundamentals, Vol. E82-A, no. 8, pp. 1588-1591, Aug. 1999.

[4] H.J Lee and O. J. Kwan "Method and apparatus for video frame recompression combining down-sampling and max-min quantizing mode," issued on Mar. 2007.

[5] M. Budagavi and M. Zhou, "Video coding using compressed reference frames," Video Coding Experts Group (VCEG) 31st Meeting, Jan. 2007.

[6] TK Tan, G. Sullivan, T. Wedi, "Recommended Simulation Conditions for Coding Efficiency Experiments", Video Coding Experts Group (VCEG) 34th VCEG Meeting, Jan. 2008.

[7] G. Bjøntegaard, "Calculation of Average PSNR Differences between RD curves", Video Coding Experts Group (VCEG) 21st Meeting, Austin, TX, April 2001.