THREE-DIMENSIONAL POSITION AND AMPLITUDE VLC CODING IN H.264/AVC

Junlin Li and Ghassan AlRegib

Georgia Institute of Technology
{lijunlin, gregib}@ece.gatech.edu

Dihong Tian, Pi Sheng Chang and Wen H. Chen

Cisco Systems, Inc.
{dtian, pschang, wenc}@cisco.com

ABSTRACT

Hybrid variable length coding (HVLC) was recently proposed as a novel entropy coding scheme for block-based image and video compression, which divides each transform block into low frequency (LF) region and high frequency (HF) region and codes them differently. To take advantage of the clustered nature of the nonzero coefficients in the LF region, a two-dimensional position and one-dimensional amplitude coding scheme (2DP1DA) was also proposed, which codes the 2D position information, i.e., run of consecutive zero-valued coefficients and run of consecutive nonzero coefficients, jointly by a 2D code table. To further improve the coding efficiency, joint position and amplitude coding is desirable. In this paper, we propose a three-dimensional position and amplitude coding scheme (3DPA) to jointly code the position and amplitude information for each nonzero cluster by a 3D code table. The experimental results show that HVLC with 3DPA for coding LF region achieves about 3.2%$\sim$3.7% bit rate reduction for a wide range of quantization parameters (QP) compared with CAVLC in H.264. Moreover, the 3DPA scheme is a compact way to realize the joint position and amplitude coding and it is very suitable for context-adaptive HVLC design.

Index Terms— Video coding, entropy coding, VLC, H.264.

1. INTRODUCTION

Block-based hybrid video coding approach has been widely adopted by the existing video coding standards, such as MPEG-2/4, H.263, and H.264/AVC [1], where motion-compensation prediction is used to exploit the temporal redundancy, spatial prediction and/or transform coding of the prediction residual are used to exploit the spatial redundancy, and entropy coding is used to exploit the statistic redundancy of the quantized transform coefficients. Variable length coding (VLC), due to its efficiency and simplicity, is widely deployed for entropy coding.

In VLC coding, the sequence of quantized transform coefficients are first represented by a sequence of coding symbols and then each symbol is coded by a unique variable length codeword. There are different ways to construct the coding symbols to efficiently represent the quantized transform coefficients. A conventional representation is (run, level) pair, where“run” indicates the number of zeros preceding a nonzero coefficient and “level” indicates the magnitude of the nonzero coefficient. In H.263, the “run” and “level” are jointly coded by a two-dimensional code table, referred to as RL-VLC. In H.264/AVC, a more sophisticated context-adaptive VLC, referred to as CAVLC, is used, where “run” and “level” are separately coded by a code table adaptively chosen from multiple code tables. Compared with RL-VLC, CAVLC is more efficient due to the introduction of multiple VLC tables with context adaptive table switch. Multiple tables based RL-VLC is also studied in [2, 3], where context-adaptive table switch is used in [2], while a position-dependent table switch is used in [3]. Another run-level based adaptive coding was also proposed in [4] to encode the “run” and “level” separately using adaptive binary arithmetic coding. Run-level based coding schemes, such as RL-VLC in H.263 and CAVLC in H.264, are efficient to code scattered nonzero coefficients, however, they are inefficient in coding clustered nonzero coefficients, due to the fact that n separate codes are required to represent n consecutive nonzero coefficients, each of which has a run equal to zero[1].

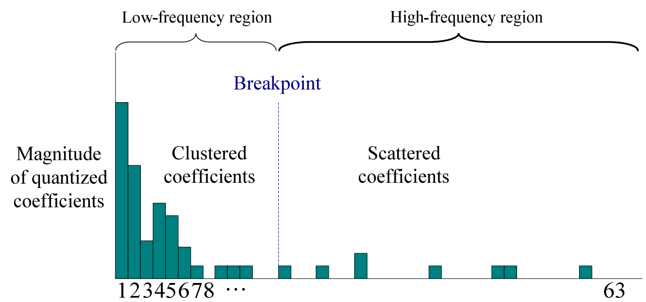

Fig. 1. Coefficient scan of an 8×8 block along a pre-defined path, e.g., zigzag.

To overcome the inefficiency of the conventional run-level coding scheme in coding consecutive nonzero transform coefficients, hybrid variable length coding (HVLC) [5] was recently proposed. In HVLC, a breakpoint, which is a coefficient index along the coefficient scan path, is first defined as shown in Fig. 1. The coefficients below and above the breakpoint are considered as low-frequency (LF) and high-frequency (HF) coefficients, respectively, where the nonzero coefficients are statistically more clustered in LF region while more scattered in HF region. In HVLC, the conventional RL-VLC or an equivalent scheme is used for coding the HF coefficients, while a new coding scheme is desired for coding the LF coefficients to exploit the clustered nature of nonzero coefficients in LF region.

For coding LF region in HVLC, two-dimensional position and one dimensional amplitude coding (2DP1DA) scheme [5] was proposed, which codes the run of consecutive zero coefficients and run of consecutive nonzero coefficients as a pair by a two-dimensional VLC table, and codes the amplitude of each nonzero coefficient independently by a one-dimensional VLC table. In [6], a joint position and amplitude coding scheme (JPAC) was proposed, where the 2D position information and the amplitudes of the last few, up to a pre-determined number of M, trailing coefficients in the nonzero cluster are jointly coded by a multiple-dimensional code table. As

[1] In H.264, the zero-runs at the start of the transform coefficient array need not be encoded because they can be inferred from the already coded parameters in the block.

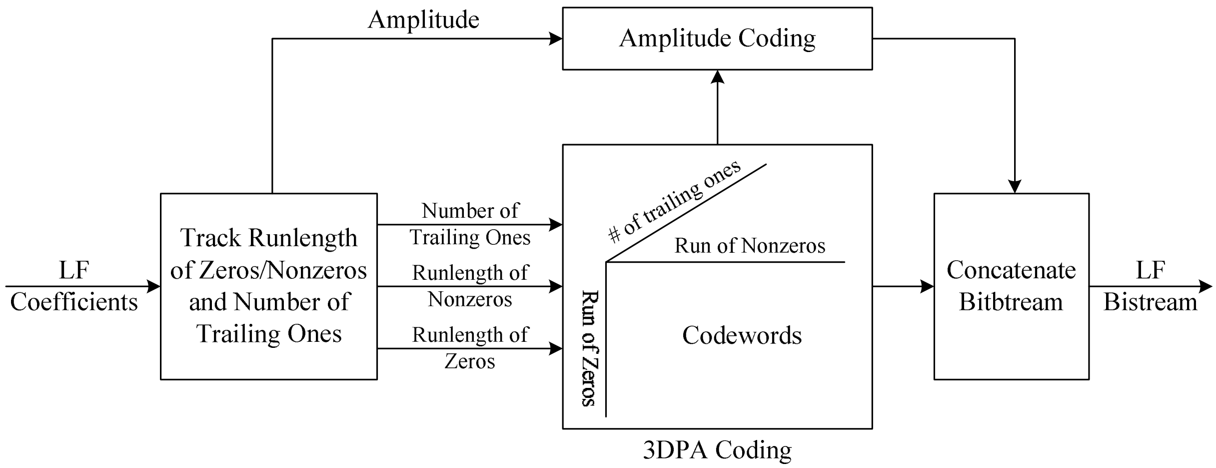

Fig. 2. The block diagram of the proposed 3DPA coding for LF coefficients.

shown in [5, 6], HVLC with 2DP1DA for coding LF region is superior to both RL-VLC in H.263 and CAVLC in H.264, and HVLC with JPAC for coding LF region further considerably improves the coding efficiency by joint position and amplitude coding with manageable code table size. Note that, in JPAC scheme, the number of the jointly coded coefficients (M) must be pre-determined to generate the code table. In this paper, we propose a three-dimensional position and amplitude coding scheme (3DPA) by jointly coding the 2D position information and the number of trailing ones at the end of the nonzero cluster with a three-dimensional code table. Compared with JPAC, 3DPA is a more compact way to realize the joint position and amplitude coding, thus it is more suitable for context-adaptive HVLC design.

The subsequent sections are organized as follows. In Section 2, we describe the proposed 3DPA scheme in detail. In Section 3, we present the experimental results. Finally, the conclusion is given in Section 4.

2. THREE-DIMENSIONAL POSITION AND AMPLITUDE CODING (3DPA) FOR LF REGION

There are essentially two types of information that need to be represented in coding a sequence of quantized transform coefficients: the positions of the nonzero coefficients and their corresponding amplitudes. In RL-VLC scheme, the position and amplitude of each nonzero coefficient are jointly coded as a (run, level) pair, but different nonzero coefficients are coded independently. On the contrary, in 2DP1DA scheme[5], the positions of consecutive nonzero coefficients are jointly coded as a two-dimensional position code (2DP), i.e., the run of zeros that precede the nonzero cluster and the run of nonzero coefficients within the cluster, while the amplitudes of the nonzero coefficients are coded separately from the position information. To further improve the coding efficiency, it is desirable to code the positions and amplitudes of the consecutive nonzero coefficients together. However, the complexity, such as code table size, will grow exponentially, thus it is necessary to have a compact design for joint position and amplitude coding.

As shown in Fig. 1, the nonzero coefficients are statistically more clustered in LF region, while they are scattered in HF region, which motivates the HVLC design. It is also observed that the amplitudes of the trailing coefficients in each nonzero cluster in LF region are often sequences of +/-1. Based on these two observations, we design a three-dimensional position and amplitude coding scheme, referred to as 3DPA, which codes the 2D position information and the number of the trailing ones of each nonzero cluster jointly with a 3D code table. Fig. 2 shows the block diagram of 3DPA scheme for coding LF region in HVLC, which includes two major components: 3DPA coding and additional one-dimensional amplitude coding.

1. *3DPA coding*: denote the run of the zero-valued coefficients that precede the nonzero cluster as R_z, denote the run of the consecutive nonzero-valued coefficients as R_n, and denote the number of trailing coefficients with amplitude +/-1 in the nonzero cluster as T_1. To efficiently code the 2D position information and the amplitudes of the trailing ones in the nonzero cluster, 3DPA scheme codes R_z, R_n and T_1 as a triple by a three-dimensional code table. Since a run of nonzero-valued coefficients implies that the following coefficient is a zero-valued coefficient (otherwise it would have been counted into the nonzero cluster), it can be skipped in the coding process to save bits, that is to say, each run of zero-valued coefficients can be reduced by 1 before it is coded, with the exception of the first run at the beginning of a block. Therefore, the code symbol of 3DPA has the following form:

$$\begin{aligned}&(R_z, R_n, T_1), \text{ for the first symbol in a block,}\\&(R_z - 1, R_n, T_1), \text{ otherwise.}\end{aligned}$$

   Table 1 shows all the coding symbols of 3DPA. To further improve the coding efficiency, a binary information, denoted as L_n, can be incorporated into the 3DPA symbols for LF region and RL symbols for HF region to indicate whether this is the last nonzero coefficient in the block, then the extended code symbol of 3DPA has the following form:

$$\begin{aligned}&(R_z, R_n, T_1, L_n), \text{ for the first symbol in a block,}\\&(R_z - 1, R_n, T_1, L_n), \text{ otherwise,}\end{aligned}$$

   where $L_n = 1$ means that it is the last nonzero coefficient in the block, while $L_n = 0$ means that there are still some remaining nonzero coefficients in the block.

2. *Amplitude coding*: since the trailing T_1 nonzero coefficients have amplitude +/-1 and have been jointly coded by the 3DPA coding as above, only the other $R_n - T_1$ nonzero coefficients

at the beginning of the nonzero cluster need to be encoded additionally. The amplitudes of all the remaining nonzero coefficients except the one immediately preceding the trailing ones are coded directly by a 1D VLC, while the amplitude of the coefficient immediately preceding the trailing ones is reduced by 1 and codec by a 1D VLC to shorten the codeword since it must be greater than 1.

To illustrate the proposed HVLC coding scheme using 3DPA for LF region and RL-VLC for HF region, consider an example shown in Table 2, where all the remaining coefficients in the sequences are assumed to be zeros, and a constant breakpoint $N = 14$ is used. In the coding stream, C_{3D} denotes the 3DPA codeword, C_A denotes the one-dimensional amplitude codeword, and C_{RL} denotes the codeword of RL-VLC. Note that the sign of each nonzero coefficient is coded separately with 1 bit and omitted in Table 2.

3. EXPERIMENTAL RESULTS

In this section, we report the preliminary test results of HVLC using 3DPA for coding LF region and RL-VLC for coding HF region. We incorporate HVLC into the H.264 codec by replacing the CAVLC entropy coding scheme in H.264 with HVLC scheme. Since the proposed HVLC algorithm is targeted at coding high resolution video sequences with 8×8 transform block size, H.264/AVC FRExt with fixed 8×8 transform is used in the simulation.

The test database includes ten 4CIF standard test sequences. Each sequence has 210 frames with a frame rate of 25 frames per second (fps). The GOP length is 15 frames, where one is INTRA coded and the others are INTER coded. For HVLC with 3DPA coding scheme, the code tables are constructed based on the measured statistics of symbols and are generated separately for INTRA and INTER modes. A constant breakpoint (N) is used for a given quantization parameter (QP). In particular, we use three pairs of (QP, N) parameters, i.e., (QP=5, N=27), (QP=25, N=20), and (QP=37, N=14), as test examples for small, medium, and large QPs respectively, and their corresponding breakpoints are determined by gathering the statistics from the training sequences to appropriately separate the clustered region and scattered region. For comparison purpose, HVLC with 2DP1DA for coding LF region and CAVLC in H.264/AVC are also tested. All the experiments are conducted using H.264/AVC FRExt with fixed 8×8 transform. Note that the complexity of HVLC with 3DPA algorithm and CAVLC in H.264 is much less than the complexity of CABAC in H.264 [7], therefore, CABAC is not included in the comparison.

Table 3, 4 and 5 show the bit-rate results for coding the test sequences with QP = 5, 25, and 37, respectively. For each sequence, we provide the respective bit-rate results for INTRA- and INTER-coded blocks and the overall bit-rate results. The total results for all the test sequences are also given. All the bit-rate results are in kbits/sec. A positive percentage in the tables indicates the bit-rate reduction achieved by HVLC, while a negative percentage indicates that the bit rate is increased by HVLC. From the results, we can see that compared with CAVLC in H.264, HVLC with 3DPA for coding LF region can reduce the bit rate considerably for a wide range of QP, i.e., 3.16% for QP=5, 3.64% for QP=25 and 3.45% for QP=37, while HVLC with 2DP1DA for coding LF region can reduce the bit rate averagely by 2.90% for QP=5, 2.00% for QP=25 and only 0.52% for QP=37, i.e., a larger gain for small QP, while a marginal gain for large QP. It can be easily understood as follows: for small QP, a big gain is achieved by solo 2D position coding while a small additional gain is achieved by joint amplitude with position coding since there are many large nonzero clusters and most of the nonzero coefficients tend to be large; on the contrary, for large QP, a small gain is achieved by solo 2D position coding while a big additional gain is achieved by joint amplitude with position coding since the nonzero clusters tend to be small and most of the nonzero coefficients also tend to be small.

Furthermore, compared with HVLC with JPAC for coding LF region [6], HVLC with 3DPA for coding LF region achieves similar coding efficiency with similar code table size. However, 3DPA scheme is a more compact way to jointly code the 2D position information and amplitude information such that it is very suitable for context-adaptive HVLC design and it serves as the framework for our current research on context-adaptive HVLC algorithm.

4. CONCLUSIONS AND FUTURE WORK

3DPA takes advantage of the clustered nature of the nonzero coefficients in the LF region and the fact that the trailing coefficients in each nonzero cluster are often sequences of +/-1 by jointly coding the 2D position information and the number of trailing ones in the nonzero cluster. HVLC with 3DPA for coding LF region achieves about 3.2%$\sim$3.7% bit rate reduction for a wide range of quantization parameters.

In HVLC with 2DP1DA in [5], JPAC in [6] or 3DPA in this paper for coding LF region, a fixed breakpoint for a given QP is used, which is not necessarily optimal. Also empirical studies show that it is difficult to find a common "optimal" breakpoint for a given coding parameter since the optimal breakpoint varies with different sequences, even different blocks in the same sequence. One way to solve this problem is to include the optimal breakpoint for each individual block in the bitstream, which introduces a considerable overhead. Another way is to design a context-adaptive mechanism to determine the "optimal" breakpoint for each block automatically without any extra cost, which is our current research direction.

5. REFERENCES

[1] ISO/IEC 14496-10, *Draft of Version 4 of H.264/AVC,* 2005.

[2] Q. Wang, D.-B. Zhao, and W. Gao, "Context-based 2D-VLC entropy coder in AVS video coding standard," *J. Comput. Sci. Technol.*, vol. 21, no. 3, pp. 315–322, 2006.

[3] G. Lakhani, "Optimal huffman coding for dct blocks," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 14, no. 4, pp. 522–527, Apr. 2004.

[4] C. Tu, J. Liang, and T. Tran, "Adaptive runlength coding," *IEEE Signal Processing Letters*, vol. 10, no. 3, pp. 61–64, Mar. 2003.

[5] D. Tian, W. H. Chen, P. S. Chang, G. AlRegib, and R. Mersereau, "Hybrid variable length coding for image and video compression," in *Proc. of the Intl. Conf. on Acoustics, Speech, and Signal Processing*, Honolulu, HI, Apr. 15-20 2007, vol. 1, pp. 1133–1136.

[6] J. Li, G. AlRegib, D. Tian, P. S. Chang, and W. H. Chen, "Joint position and amplitude coding in hybrid variable length coding for video compression," in *Proc. of the Intl. Conf. on Acoustics, Speech, and Signal Processing*, Las Vegas, NV, Mar. 30 - Apr. 4 2008, to appear.

[7] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard," *IEEE Trans. Circuits and Systems for Video Tehnology*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

Table 1. Symbols of 3DPA

zero run	nonzero run with the number of trailing ones												
	1		2			3				...	n			...
0	(0,1,0)	(0,1,1)	(0,2,0)	(0,2,1)	(0,2,2)	(0,3,0)	(0,3,1)	(0,3,2)	(0,3,3)	...	(0,n,0)	...	(0,n,n)	...
1	(1,1,0)	(1,1,1)	(1,2,0)	(1,2,1)	(1,2,2)	(1,3,0)	(1,3,1)	(1,3,2)	(1,3,3)	...	(1,n,0)	...	(1,n,n)	...
2	(2,1,0)	(2,1,1)	(2,2,0)	(2,2,1)	(2,2,2)	(2,3,0)	(2,3,1)	(2,3,2)	(2,3,3)	...	(2,n,0)	...	(2,n,n)	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
z	(z,1,0)	(z,1,1)	(z,2,0)	(z,2,1)	(z,2,2)	(z,3,0)	(z,3,1)	(z,3,2)	(z,3,3)	...	(z,n,0)	...	(z,n,n)	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Table 2. An Example of HVLC with 3DPA for LF region and RL-VLC for HF region

Index:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	$\cdots$
Coefficient:	9	-5	3	-2	1	0	0	2	1	1	0	0	0	-1	1	0	0	0	1	0	$\cdots$
Coding Stream:	$C_{3D}(0,5,1,0)$ $C_A(9)$ $C_A(5)$ $C_A(3)$ $C_A(1)$ $C_{3D}(1,3,2,0)$ $C_A(1)$ $C_{3D}(2,2,2,0)$ $C_{RL}(2,1,1)$																				

Table 3. Bit-rate results for QP = 5 and N = 27 (kb/s)

Sequence	INTRA			INTER			Overall		
	H.264[a]	2DP1DA	3DPA	H.264	2DP1DA	3DPA	H.264	2DP1DA	3DPA
Crew	3688.02	0.28%	0.46%	47765.26	0.35%	0.63%	51453.28	0.34%	0.62%
Harbour	4319.22	2.92%	3.08%	51629.02	1.15%	1.20%	55948.23	1.29%	1.34%
Soccer	3667.38	2.50%	2.83%	43490.45	1.31%	1.76%	47157.83	1.40%	1.85%
Barcelona	7095.85	-0.41%	-0.26%	75807.66	3.78%	3.89%	82903.51	3.42%	3.53%
Fries	3820.64	0.88%	1.02%	47980.86	0.40%	0.89%	51801.5	0.43%	0.90%
Mobile	6499.55	0.39%	0.78%	64043.72	3.03%	3.37%	70543.27	2.78%	3.13%
Music	5318.01	1.11%	1.31%	62068.96	3.14%	3.19%	67386.97	2.98%	3.04%
Race	5145.37	3.59%	3.93%	63949.22	3.49%	3.80%	69094.59	3.49%	3.81%
Rower	4726.3	9.47%	9.93%	56193.65	5.27%	5.73%	60919.96	5.59%	6.05%
Rugby	5114.3	7.12%	7.40%	64950.39	5.25%	5.43%	70064.69	5.39%	5.58%
Total	49394.64	2.66%	2.92%	577879.19	2.92%	3.18%	627273.83	2.90%	3.16%

Table 4. Bit-rate results for QP = 25 and N = 20 (kb/s)

Sequence	INTRA			INTER			Overall		
	H.264	2DP1DA	3DPA	H.264	2DP1DA	3DPA	H.264	2DP1DA	3DPA
Crew	573.12	-0.23%	1.89%	4444	-2.26%	1.17%	5017.12	-2.03%	1.25%
Harbour	1047.71	5.16%	6.06%	7407.56	-1.32%	2.52%	8455.27	-0.52%	2.96%
Soccer	782.68	1.76%	2.87%	4745.95	-1.04%	2.30%	5528.63	-0.64%	2.38%
Barcelona	2336.33	-1.47%	-1.09%	13491.89	1.40%	3.42%	15828.22	0.98%	2.76%
Fries	539.73	0.52%	2.53%	4247.23	-1.60%	1.29%	4786.96	-1.36%	1.43%
Mobile	2081.04	-1.91%	-1.40%	10822.53	1.43%	3.38%	12903.57	0.89%	2.61%
Music	1253.53	-2.55%	-1.71%	8293.97	-0.10%	1.47%	9547.49	-0.42%	1.06%
Race	1349.74	3.98%	4.32%	13559.64	4.53%	4.88%	14909.37	4.48%	4.83%
Rower	1138.95	10.66%	11.21%	9767.34	5.68%	6.49%	10906.29	6.20%	6.99%
Rugby	1293.31	8.08%	8.76%	13304.42	4.63%	5.59%	14597.73	4.94%	5.87%
Total	12396.14	1.96%	2.69%	90084.53	2.00%	3.77%	102480.65	2.00%	3.64%

Table 5. Bit-rate results for QP = 37 and N = 14 (kb/s)

Sequence	INTRA			INTER			Overall		
	H.264	2DP1DA	3DPA	H.264	2DP1DA	3DPA	H.264	2DP1DA	3DPA
Crew	145.63	0.06%	2.90%	754.14	-0.11%	1.84%	899.77	-0.08%	2.01%
Harbour	332.74	2.80%	6.22%	1073.7	-3.23%	2.13%	1406.44	-1.80%	3.10%
Soccer	188.96	-0.74%	2.44%	790.36	-0.63%	2.30%	979.32	-0.65%	2.33%
Barcelona	734.38	-0.65%	0.78%	1645.03	-1.01%	2.08%	2379.42	-0.90%	1.68%
Fries	151.49	1.87%	4.48%	906.1	0.14%	2.48%	1057.59	0.39%	2.77%
Mobile	717.25	-1.28%	-0.47%	1736.41	0.07%	3.57%	2453.67	-0.32%	2.39%
Music	400.12	-1.90%	-0.99%	1630.86	-0.92%	2.21%	2030.99	-1.11%	1.58%
Race	410.72	1.73%	3.62%	2867.58	0.22%	3.77%	3278.3	0.41%	3.75%
Rower	347.09	4.43%	7.49%	2191.23	1.08%	4.42%	2538.31	1.53%	4.84%
Rugby	390.92	5.19%	7.33%	3123.91	3.64%	6.06%	3514.83	3.81%	6.21%
Total	3819.3	0.84%	2.73%	16719.32	0.44%	3.61%	20538.64	0.52%	3.45%

[a] In Tables 3, 4 and 5, the results of H.264 are generated using H.264/AVC FRExt with CAVLC and fixed 8×8 transform.