2011 18th IEEE International Conference on Image Processing

REDUCED-COMPLEXITY ENTROPY CODING OF TRANSFORM COEFFICIENT LEVELS USING TRUNCATED GOLOMB-RICE CODES IN VIDEO COMPRESSION

Tung Nguyen[1], *Detlev Marpe*[1], *Heiko Schwarz*[1], *and Thomas Wiegand*[1,2]
[tung.nguyen | detlev.marpe | heiko.schwarz | thomas.wiegand]@hhi.fraunhofer.de

[1] Fraunhofer Institute for Telecommunications
Heinrich Hertz Institute
Image Processing Department
Einsteinufer 37, D-10587 Berlin, Germany

[2] Image Communication Chair
School of EE & CS
Berlin Institute of Technology
Einsteinufer 17, D-10587 Berlin, Germany

ABSTRACT

In hybrid video coding, the difference between the intra or inter prediction signal and the original signal is transmitted using block-based transform coding. The state-of-the-art in coding the quantized transform coefficients is the approach specified in H.264/AVC for context-adaptive binary arithmetic coding. It has, however, been shown that the number of binary symbols that have to be arithmetically coded for the transform coefficients can become very large, making the concept less attractive for high rate applications. To overcome this issue, we propose a combination of simple variable-length codes and context-adaptive binary coding, which yields the same coding efficiency as the H.264/AVC transform coefficient coding at a lower complexity level and which has been adopted into the HEVC test model (HM).

Index Terms**—** Transform coding, video compression

1. INTRODUCTION

All video coding standards of practical importance (such as the state-of-the-art standard H.264/AVC [1]) and also the test model for HEVC [2], which is a new standardization activity of ITU-T VCEG and ISO/IEC MPEG, specify block-based transform coding. The difference between the intra or inter prediction signal and the original signal is transformed using a separable block transform, the transform coefficients are quantized, and the resulting transform coefficient levels are entropy coded. The state-of-the-art for coding the transform coefficient levels is the approach specified for the context-adaptive binary arithmetic coding (CABAC) [3] in H.264/AVC, which was also adopted into the HEVC test model including some improvements for larger block sizes [4]. For reducing the complexity of the entropy coding for HEVC, it is also considered to replace the core arithmetic coding engine of CABAC with PIPE [5], which represents a combination of probability interval partitioning and simple codes that map a variables number of binary symbols (bin) to variable-length codewords. The context modeling and the mapping of transform coefficient levels to binary symbols for PIPE is the same as for CABAC. It consists of two steps. In the first step, a so-called significance map is coded, which specifies the positions of transform coefficient levels not equal to 0. After coding the significance map, the absolute transform coefficient levels are coded using adaptive context modeling, while the sign information is coded in the low-complexity bypass mode.

In the worst-case scenario, 26 bins per pixel (for 4:2:0 chroma sampling) have to be coded using the context-adaptive arithmetic coding engine (or the equivalent in the case of PIPE), where 21 of these bins are related to the coding of the absolute transform coefficient levels. In order to reduce the complexity of the entropy coding it is desirable to decrease the number of bins that are coded in a context-adaptive way. This is particularly important for hardware implementations with low power consumption and limited clock cycles. This goal can be achieved without any loss in coding efficiency by coding a part of the binary representation for absolute transform coefficients using truncated Golomb-Rice (Rice) [6] codes.

The paper is organized as follows. Sec. 2 briefly reviews the CABAC transform coefficient coding. The proposed modifications are described in Sec. 3. In Sec. 4 and Sec 5, the modified concept is analyzed with respect to its coding efficiency and complexity.

2. CABAC TRANSFORM COEFFICIENT CODING

In CABAC or PIPE, the absolute values of the transform coefficient levels are binarized using a concatenation of a truncated unary (TU) code and a 0th order Exp-Golomb (EG0) code as illustrated in Fig. 1. The bin 0 is already coded as part of the significance map and is not considered as part of the binarization. The bins 1 to 14 are coded using a TU code (“TU prefix” in Fig. 1). Starting with bin 15, all remaining bins are coded using an EG0 code (“EG0 suffix” in Fig. 1). It means, first the minimum of the absolute

978-1-4577-1303-3/11/$26.00 ©2011 IEEE

transform coefficient level minus 1 and 14 is binarized using a TU code. If the absolute transform coefficient level is greater than 14, the TU prefix part is followed by an EG0 code for the absolute transform coefficient level minus 15.

abs_level	Bin string																			
	TU prefix														EG0 suffix					
1	0																			
2	1	0																		
3	1	1	0																	
4	1	1	1	0																
5	1	1	1	1	0															
...	...	...	...	...	...	...														
...	...	...	...	...	...	...	...	...	...	...	...	...								
13	1	1	1	1	1	1	1	1	1	1	1	1	0							
14	1	1	1	1	1	1	1	1	1	1	1	1	1	0						
15	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0					
16	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0			
17	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1			
18	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	0	
19	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	0	1	
20	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	0	1	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
bin	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	...

Fig. 1: Binarization of transform coefficient levels in H.264/AVC

For the first bin of the bin string, a context model from a set of 5 context models is selected based on already coded bins inside the transform block. For the remaining bins of the TU prefix part, a context model from another set of 5 context models is selected. Here, the same context model selection mechanism and the same set of 5 context models are used for the bins 2 to 14. The bins of the EG0 part are coded with a fixed non-adaptive context model (with a probability of 0.5 for both binary values) using the bypass mode of CABAC or PIPE.

This binarization scheme is further illustrated in Fig. 2, where “PIPE or CABAC” denotes the TU prefix part and “EG0” denotes the EG0 part. The range of the transform coefficient levels is divided into two intervals. If an absolute transform coefficient level falls inside the first interval, it is coded using a TU binarization; otherwise a sequence of 14 bins equal to 1 is coded followed by an EG0 code for the absolute transform coefficient level minus 15.

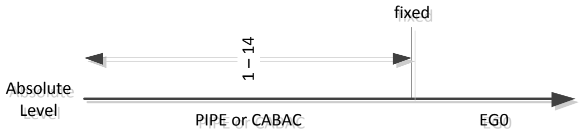

Fig. 2: Decomposition of the transform coefficient range in H.264/AVC

In the worst case scenario, all 14 bins for the TU prefix part of an absolute transform coefficient level have to be coded or parsed in a context-adaptive way, while the remaining bins for the absolute transform coefficient levels are bypassed. It should be noted that for PIPE, the bypass consists only of a direct insertion of the EG0 code into the bitstream, while for CABAC, a range update is needed for each bin. In total, up to 26 bins have to be coded using adaptive contexts.

3. PROPOSED MODIFICATIONS

For reducing the complexity of the CABAC/PIPE transform coding, we reduced the number of context-adaptively coded bins for coding the absolute transform coefficient levels. This is achieved by introducing a variable-length code, as illustrated Fig. 3, for which the resulting bins are coded in the bypass mode. First, the minimum of the absolute transform coefficient level minus 1 and 3 is binarized using a TU code. This binarization is basically the same as the binarization for the TU prefix part described above, only the maximum value is reduced from 14 to 3. Hence, at most 2 bins for the absolute transform coefficient levels are coded using adaptive context models. The context modeling for these two bins is not modified relative to H.264/AVC.

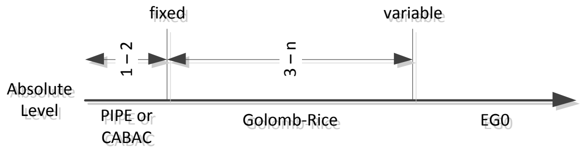

Fig. 3: Proposed decomposition of the transform coefficient range

If absolute transform coefficient levels greater than 2 are coded (i.e., if the first two bins are equal to ‘11’), a second entropy code is used to code the remaining part of the absolute transform coefficient level. Therefore, the value of the absolute transform coefficient level minus 3, up to a specific limit, is coded using a truncated Rice code, where all bins of the Rice codewords are coded in the low-complexity bypass mode. The limit is variable and depends on the Rice parameter, which is derived using backward adaptation as will be described below. If the absolute transform coefficient level minus 3 is larger than the maximum value that can be coded with the truncated Rice code, 3 and the maximum value supported by the selected Rice code are subtracted from the absolute transform coefficient level. The resulting value is then coded with the EG0 code in bypass mode, similar to the original H.264/AVC concept.

The difference to the transform coding in H.264/AVC and the HM is that at maximum 2 bins (instead of 14 bins) per absolute transform coefficient level are coded using adaptive context models, which results in a complexity reduction and, in particular, in a reduction of the worst-case complexity. It should be noted that in combination with PIPE, all except the first 2 bins can be directly inserted into the bitstream, while for regular CABAC a low-complexity range update is still required.

3.1 Truncated Rice codes

Rice codes are a subset of Golomb codes. The Golomb codes are constructed as follows. Given a particular Golomb parameter k_G, the value n to be coded is decomposed into a prefix part p and a remainder part r,

$$p = \left\lfloor \frac{n}{k_G} \right\rfloor, \qquad r = n - p \cdot k_G .$$

The remainder is coded using a truncated binary representation, while the prefix is coded using a unary code, where multiple values can have same prefix. Note that the resulting value $p+1$ represents the number of bins of the prefix part. Rice codes are characterized by a parameter k. The relationship between this Rice parameter and the parameter k_G of the corresponding Golomb code is given by

$$k_G = 2^k .$$

Note that the Golomb parameter k_G for Rice codes is always a multiple of 2 and, hence, the remainder part represents fixed-length codewords with k bits. As examples, the first 8 codewords of the Rice codes with k equal to 1 and 2 are shown in Table 1.

Table 1: Examples for Rice codes

value	k=1		k=2	
	prefix	rem.	prefix	rem.
0	0	0	0	00
1	0	1	0	01
2	10	0	0	10
3	10	1	0	11
4	110	0	10	00
5	110	1	10	01
6	1110	0	10	10
7	1110	1	10	11

Golomb codes and hence also Rice codes are designed for infinite alphabets. By using Rice codes for finite alphabets as in this paper, the codeword set can be truncated in order to remove the redundancy in the binary representation. In the proposed method, the number of values for the Rice codes is selected in a way that only the prefix is truncated (similarly to the truncated unary code), while no remainder truncation is required. The benefit from avoiding remainder truncation is that no further shift operations are required for reconstructing the remainder value. The Rice parameter specifies the number of bins in the remainder part. In addition, the maximum of the Rice parameter is chosen to be equal to 3.

3.2 Selection of Rice parameters

As mentioned above, the Rice parameter is derived using backward-adaptive modeling. For each transform block and the first scan position that contains a significant coefficient greater than 2, the absolute transform coefficient level is coded using the Rice code with $k = 0$. Depending on the previously coded absolute transform coefficient level greater than 2, the parameter k is either increased or kept constant. Its maximum value is 3 as mentioned before. The parameter k is never decreased inside a transform coefficient block. The parameter selection can be summarized by the following formula, where k_{t+1} denotes the parameter for the next coding stage, k_t the current parameter, $value_t$ denotes the current value of absolute transform coefficient level minus 3, and $n_{k\,t}$ denotes the threshold, which depends on the current parameter k_t, with $n_0 = 12, n_1 = 10, n_2 = 9$:

$$k_{t+1} = \begin{cases} 0 & value_t \in [0,1] \wedge k_t < 1 \\ 1 & value_t \in [2,3] \wedge k_t < 2 \\ 2 & value_t \in [4, n_{k_t}] \wedge k_t < 3 \\ 3 & value_t > n_{k_t} \wedge k_t < 4 \\ k_t & otherwise \end{cases}$$

The maximum possible value which can be coded with the Rice code depends on the Rice parameter k. This relationship is shown in Table 2.

Table 2: Maximum value coded with specific Rice parameter

Rice parameter	Max. value
0	7
1	20
2	42
3	70

4. EXPERIMENTAL RESULTS

For demonstrating the negligible impact of the proposed modification on the coding efficiency of the CABAC (or PIPE) transform coding, we implemented it into the second HEVC test model. Simulations comparing the coding efficiency of the proposed method with that of the original CABAC approach have been run for the set of HEVC test sequences and three different configurations: intra-only coding, IPPP, and hierarchical B pictures with a GOP size of 8 pictures. For each sequence, 4 quantization parameters have been tested according to the HEVC test conditions and the average bit rate saving has been calculated using the Bjøntegaard metric [7]. The average bit rate savings are summarized in Table 3 where negative values stand for bitrate savings. As can be seen, the coding efficiency is virtually not influenced.

Table 3: Average bit rate savings of the proposed method relative to the original CABAC transform coefficient coding

Sequence	Resolution	Intra	GOP 8	IPPP
Traffic	2560x1600	-0.07 %	-0.01 %	
PeopleOnStreet	2560x1600	-0.17 %	-0.12 %	
NebutaFestival	2560x1600	-0.14 %	0.25 %	
SteamLocomotive	2560x1600	0.08 %	0.01 %	
Kimono	1920x1080	-0.15 %	-0.07 %	-0.04 %
ParkScene	1920x1080	-0.01 %	0.01 %	0.06 %
Cactus	1920x1080	-0.04 %	-0.03 %	0.06 %
BasketballDrive	1920x1080	-0.01 %	-0.05 %	-0.04 %
BQTerrace	1920x1080	-0.05 %	-0.01 %	0.00 %
BasketballDrill	832x480	-0.01 %	-0.02 %	-0.01 %
BQMall	832x480	-0.07 %	-0.09 %	-0.02 %
PartyScene	832x480	-0.01 %	0.01 %	0.06 %
RaceHorses	832x480	0.07 %	-0.03 %	-0.01 %
BasketballPass	416x240	-0.04 %	-0.08 %	-0.03 %
BQSquare	416x240	-0.18 %	0.01 %	0.02 %
BlowingBubbles	416x240	0.02 %	0.02 %	-0.09 %
RaceHorses	416x240	-0.06 %	0.10 %	-0.11 %
Vidyo1	1280x720	-0.07 %		-0.60 %
Vidyo3	1280x720	-0.09 %		0.03 %
Vidyo4	1280x720	-0.06 %		-0.04 %
Average		**-0.05 %**	**-0.01 %**	**-0.05 %**

5. COMPLEXITY ANALYSIS

For evaluating the impact on the encoder and decoder complexity, we analyzed the number of bins that have to be coded or parsed by the regular coding engine of PIPE/CABAC (i.e., without counting bins that are coded in bypass mode) in the worst case scenario. At maximum 2 bins are spent per luma sample for describing the side information as the quadtree structure, the motion vector difference, the motion vector index, prediction mode etc. For the original transform coding in CABAC, up to 26 regularly coded bins can be required per pixel including the side information. The remaining bins are coded in bypass mode. With the proposed method, a reduction of the maximum number of regularly coded bins to 8 bins per pixel (2 bins for the side information and 6 bins for transform coefficient information) is achieved. We encoded the test set with the lowest possible quantization parameter (i.e., the quantization parameter (QP) is set equal to 0) for checking the worst case scenario on real sequences. Therefore, we counted the number of total and bypassed bins and calculated the number of bins per pixel. Table 4 lists the number of regularly coded bins per pixel and Table 5 shows the relative number of bypassed bins. It can be seen that the number of bins that are coded with adaptive context models is significantly decreased using the modified transform coding. The relative number of bins coded in bypass mode is increased by approximately a factor of 2. It should be noted that the maximum number of regularly coded bins per pixel cannot become greater than 8.

Table 4: Average number of regularly coded bins per pixel (QP = 0)

bin/pixel	Anchor	Proposed
Min	1.52 bpp	1.21 bpp
Max	8.13 bpp	3.64 bpp
Average	4.29 bpp	2.84 bpp

Table 5: Relative number of bins coded in bypass mode (QP = 0)

% EP bins	Anchor	Proposed
Intra	20.20%	45.98%
Random	17.91%	34.90%
LowDelay	17.42%	35.68%
Average	18.52%	39.02%

6. CONCLUSTION

A reduced-complexity method for coding the absolute transform coefficient levels using the CABAC framework has been described. The presented method provides the same coding efficiency as the original CABAC transform coefficient coding, but the associated complexity, in particular the worst-case complexity, is reduced. This is achieved by introducing a Rice code in the binarization for which all bins are coded in the low-complexity bypass mode.

REFERENCES

[1] ITU-T and ISO/IEC, "Advanced Video Coding for Generic Audiovisual Services", ITU-T Rec. H.264 and ISO/IEC 14496-10, version 13, 2010.

[2] JCT-VC, "Working Draft 1 of High-Efficiency Video Coding", Doc. JCTVC-C403, Oct. 2010.

[3] D. Marpe, H. Schwarz, and T. Wiegand, "Context-Based Adaptive Binary Arithmetic Coding in H.264/AVC Video Compression Standard", IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 620-636, July 2003.

[4] T. Nguyen, H. Schwarz, H. Kirchhoffer, D. Marpe, and T. Wiegand, "Improved Context Modeling for Coding Quantized Transform Coefficients in Video Compression", Picture Coding Symposium (PCS 2010), Nagoya, Japan, Dec 7-10, 2010.

[5] D. Marpe, H. Schwarz, and T. Wiegand, "Probability Interval Partitioning Entropy Codes", Proc. Picture Coding Symposium (PCS 2010), Nagoya, Japan, Dec 7-10, 2010.

[6] Golomb S., "Run-length encodings", IEEE Transactions on Information Theory, Vol. 12, No. 3, pp. 399-401, Jul. 1966.

[7] G. Bjøntegaard, "Calculation of average PSNR differences between RD curves", ITU-T SG16/Q6, Doc. VCEG-M33, Apr. 2001.

[8] ITU-T Q6/16 and ISO/IEC JCT1/SC29/WG11, "Joint Call for Proposals on Video Compression Technology", Doc. VCEG-AM90, WG11 N11113, Jan. 2010.