



# Efficient Transform Coefficient Coding in HEVC

Claudia C. Oprea<sup>(✉)</sup>, Radu O. Preda, Ionut Pirnog,  
and Robert Al. Dobre

Telecommunication Department, University Politehnica of Bucharest,  
Iuliu Maniu 1-3, 061071 Bucharest, Romania  
{crisrina, radu, ionut}@comm.pub.ro,  
rdobre@elcom.pub.ro

**Abstract.** This paper presents an efficient coding method of transform coefficients in HEVC video encoding standard. Our approach makes use of spectral characteristics of the transform unit coefficients in inter predicted coding units. Also a coefficient significance assessment algorithm is integrated into the video encoder, for a non-zero coefficients number reduction. It is shown to be a good method from the compression point of view, without quality losses.

**Keywords:** HEVC · Transform coefficient · Residual coding  
Significant coefficients

## 1 Introduction

The successor of the video coding standard H.264/AVC, H.265/HEVC, brought innovation into multiple video encoding algorithms and succeeds to outperform the compression efficiency of its predecessor, without any visible loss in visual quality. HEVC includes the regular video coding sections: prediction in space and time, residual transform, and entropy coding.

Frames are split into large coding units (LCUs), which are further partitioned recursively in a quad-tree structure of coding units (CUs). CUs occur in miscellaneous sizes depending on the actual frame content needed to be predicted, starting from  $64 \times 64$  pixels all the way down to  $8 \times 8$  pixels. CUs represent the basic unit of prediction in HEVC and they come in two flavors: intra or inter coded. A CU can be divided using one of eight partition modes, depending on the prediction type, so inter CUs are allowed to be as small as  $8 \times 4$  pixels and  $4 \times 8$  pixels, indicating there is no  $4 \times 4$  motion compensation. In inter prediction, like AVC, HEVC has 2 reference lists for single or bi-directional prediction, and can hold up to 16 references each. By limiting the smallest inter block to be one directional predicted, there is less reading from memory and more time and power, so a longer battery life for example.

A residual signal is coded for each coding unit and HEVC supports four transform sizes:  $4 \times 4$ ,  $8 \times 8$ ,  $16 \times 16$  and  $32 \times 32$  pixels. The transform is based on the discrete cosine transform (DCT) and uses basis matrices with approximated integer coefficients, so there is not possible to have a perfect reconstruction after the inverse

transform. Non-square CUs are coded using square transform units (TUs). Another quad-tree structure is used to split the residual of a CU into TUs (residual quad-tree - RQT). Supposing a TU has non-zero coefficients, a coded block flag is set to notify it, while the residual coding is signaling the location and the values of those significant coefficients.

This paper focuses on the selection of the significant transform coefficients. Not all non-zero coefficients are actually important for a good frame reconstruction, while the elimination of those considered less important proves to be a favorable solution for bitstream reduction, without perceptual losses in visual quality. The simulations performed on five test videos, concerning only a  $4 \times 4$  TU size, show minor or no losses in perceptual quality and an improved bitrate value.

## 2 Residual Coding in HEVC

Four options regarding the residual coding are available in HEVC. First, the difference between the original block and its prediction is transformed, quantized and passed to the entropy coder. Second choice in the standard specification is to skip  $4 \times 4$  transform and apply the quantization directly to the non-transformed residual signal. Third, transform and quantization can be omitted and the residual is coded directly by CABAC (context adaptive binary arithmetic coding). Last option is to apply pulse-code modulation (PCM) to the original CU, without prediction, transform and everything else.

Our approach concerns the first option, the one most commonly used. The standard specification guide only the inverse transform at the decoder side, leaving the encoder implementation free from regulation. Obviously, the direct transform in the encoder will use the same basis matrices specified for the decoder, while any other method regarding a significant coefficient selection is left at the developer's choice.

There is a single entropy coding method used in HEVC, context adaptive binary arithmetic coding (CABAC), unlike AVC that sustains also context adaptive variable length coding (CAVLC). TUs coefficients are coded in the bitstream differently from AVC: starting with the last position, the coefficients are scanned diagonally, and for each group of  $4 \times 4$  values, a bit indicates a non-zero value in the 16 coefficients group. Then for each of the non-zero coefficients in a group, the remainder of the level is signaled. Naturally, any residual coefficient decided to be less important by an external algorithm will be omitted from the bitstream, thus conducting to a smaller video bitrate. Such an algorithm should maintain overall visual quality.

Similar attempts were made in the past. An all-zero block detection scheme is proposed in [1], prior to DCT, to reduce the encoding complexity. The aim is the detection of all-zero-quantized blocks, before DCT, so that the following stages of the transform and quantization can be skipped. The drawback of this approach is that it uses the relationship between Hadamard and DCT transform kernels, in a scenario when the encoder performs rate distortion optimization (RDO). RDO is scarcely ever done in real time low-complexity encoders.

More complex research in [2] develop a DCT-based local distortion detection probability (LDDP) model, that can estimate a degree of distortion visibility for any

distribution of the transform coefficients regardless the TU size. A perceptual video coding scheme, HEVC compliant, is implemented based on the LDDP model, so that transform coefficients are sufficiently suppressed when possible, in order to obtain a significant bitrate reduction. Another kind of optimization approach is described in [3], which is to early detect zero quantized transform coefficients in  $4 \times 4$  TUs, before implementing transform and quantization. The authors propose two sufficient prediction conditions that are mathematically deduced and relate to DCT, then show the proposed algorithm is able to efficiently predict all-zero  $4 \times 4$  blocks when higher QP (quantization parameter) values are used.

In [4] is presented a method of transcoding of HEVC encoded video. The approach enables a fast transcoding by removing some carefully selected coefficients from the bitstream. The removal of coefficients is done on the bitstream level and provides a bit rate reduction of up to 10%, with the image quality decrease of about 0.2–0.5 dB.

Our approach follows the idea in [4], but is not influenced by the quantization parameter value, nor uses complex algorithms to perceptually assess the transform coefficients' significance. It is inspired by an older algorithm proposed in H.264/AVC reference software, JM [5], and can be used by low-complexity encoders.

### 3 Transform Coefficient Significance Method

It is quite common in video coding that a substantial number of transform coefficients of the prediction residual are quantized to zero or very small amplitude levels. Considerable bitrate savings can be obtained if these small amplitude levels coefficients, when placed nearby other zero-coefficients, could be reduced to zero as well, without losing quality. This usually orders zeros in the quantized residual coefficient block into long sequential runs that can be encoded efficiently with run-length coding, as previously has been done in JM reference software for H.264/AVC. In the following, the  $4 \times 4$  DCT and the associated quantization functions used by HEVC are briefly presented, which is useful to derive the proposed algorithm for coefficient significance assessment.

Consider the residual  $4 \times 4$  block  $r(x, y)$ ,  $0 \leq x, y \leq 3$ , applied to a  $4 \times 4$  transform in the horizontal and vertical directions, where  $C$  is the two-dimensional transform matrix:

$$C = \begin{bmatrix} 64 & 64 & 64 & 64 \\ 83 & 36 & -36 & -83 \\ 64 & -64 & -64 & 64 \\ 36 & -83 & 83 & -36 \end{bmatrix}, \quad (1)$$

In order to achieve more computational efficiency, the standard specifies only multiplication, addition and shift operations in integer arithmetic in the formula of the transform coefficient block,  $t(u, v)$ ,  $0 \leq u, v \leq 3$ :

$$t(u, v) = \left\{ \sum_{x=0}^3 C(u, x) \cdot \left[ \left( \sum_{y=0}^3 r(x, y) \cdot C^T(y, v) + 1 \right) \ggg 1 \right] + 2^7 \right\} \ggg 8, \quad (2)$$

where  $C^T$  is the transposed matrix of  $C$ , and  $\ggg$  represents a binary shift to the right.

The quantized transform coefficient,  $z(u, v)$ ,  $0 \leq u, v \leq 3$  is obtained as

$$z(u, v) = \text{sign}(t(u, v)) \cdot [(|t(u, v)| \cdot m + e) \gg \text{qbits}], \quad (3)$$

where  $\text{qbits} = 19 + \text{floor}(QP/6)$ ,  $QP$  is the quantization parameter, and  $e = \vartheta \ll (\text{qbits} - 9)$ . The offset parameter  $\vartheta$  is 171 in intra prediction and 85 for inter prediction, while  $\ll$  represents a binary shift to the left. The multiplication factor  $m$  depends on the quantization parameter and is given in Table 1, as specified by the standard documentation [6].

**Table 1.** Multiplication factor  $m$ .

QP%6	0	1	2	3	4	5
$m$	26,214	23,302	20,56	18,396	16,384	14,564

At this stage of the encoding process, having the transformed and quantized coefficients in a  $4 \times 4$  block, the following modified algorithm in [5] is applied. The main purpose is to prevent that single or so-called “expensive” coefficients are coded. Using a TU size of  $4 \times 4$ , there are chances that a single coefficient in a  $8 \times 8$  CU or  $16 \times 16$  CU may be non-zero. For example, in JM reference software for H.264/AVC, one single small coefficient (AC level equal to 1) costs at least 3 bits for the coefficient, 4 bits for the EOBs (end of block) for the  $4 \times 4$  blocks, possible even more bits for the coded block pattern. The total cost for that single small coefficient will then typically be around 10–12 bits.

```

initialize number_zeros = 16;
for(all z(u,v) coefficients) {
  if(current_coef_z(u,v) != 0) {
    decrease number_zeros;
    if(current_coef_z(u,v) > amplitude_threshold) {
      cost(current_coef_z(u,v)) += maximum_cost;
    }
    else {
      cost(current_coef_z(u,v)) += cost_threshold(number_zeros);
    }
    next current_coef_z;
    update number_zeros;
  }
}
for(all z(u,v) coefficients) {
  if(cost(current_coef_z(u,v)) > 4) {
    keep coefficient value;
  }
  else {
    discard coefficient value;
  }
}

```

The basic idea from H.264/AVC reference software was to keep tracking the number of consecutive zero values in the  $4 \times 4$  transformed and quantized vector of

coefficients. While the consecutive zeros count increases, the cost threshold for a discarded coefficient decreases, thus making the encoding process more efficient. The thresholds in JM have been adapted to the corresponding values needed in HEVC, also the amplitude threshold has been updated: only coefficients having the quantized value smaller than the amplitude threshold can be discarded (set to zero).

## 4 Implementation and Experimental Results

We simulate the algorithm for coefficient discarding using HEVC reference software provided by ITU-T [7]. The test configuration used a TU size of  $4 \times 4$  pixels independent of the CU size. The algorithm for coefficient elimination run at three QP values, uniformly distributed in the overall QP range (from 0 to 51):  $QP \in \{10, 26, 42\}$ . Two types of amplitude threshold were taken into consideration: a fix threshold and a content adaptive threshold, computed as the local mean luminance, transformed and quantized according to the current QP value. The fix amplitude threshold is set to a relatively reduced value in order to select only coefficients that have very small values and assign them a low cost of discarding. When the elimination cost is high, the coefficient will remain as it is in the final encoded bitstream, while a lower cost indicates a coefficient to be discarded.

Simulations have been performed for a set of five test videos, all taken at full high definition resolution,  $1920 \times 1080$  p. Objective perceptual visual quality evaluation has been applied to establish the performance of the algorithm presented in this study. We used the structural similarity distortion metric (SSIM) described in [8] for perceptual quality evaluation. In Table 2, the simulations show a distortion metric index that is not implying a decrease in visual quality, since the difference relative to the original sequence can be found at the third and fourth decimal place. There is though a better performance in bitrate in the case of the algorithm with adaptive local amplitude threshold (*ATh*) compared to the fix threshold algorithm (*FTh*), that suggests there is a correlation between the coefficient elimination and the visual contrast sensitivity threshold.

**Table 2.** BD rate and encoding time of the proposed method with respect to the reference software HM in main profile.

Test sequences	BD rate ( <i>FTh</i> ) Y[%]	Ratio-time ( <i>FTh</i> ) [%]	BD rate ( <i>ATh</i> ) Y[%]	Ratio-time ( <i>ATh</i> ) [%]
CrowdRun	-0.2	-4.3	0.14	-2.7
DucksTakeOff	0.1	-5.0	0.22	-2.1
ParkJoy	-0.1	-3.7	-0.08	-1.7
RiverBed	-3.3	-3.5	-3.5	-2.04
Sunflower	-2.6	-2.4	-2.6	-1.9

The efficiency of the proposed method is measured by the Bjontegaard Distortion Rate (BD-Rate) in Table 2. Also called Bjontegaard delta rate, this parameter

corresponds to the average bit rate difference in percent for the same PSNR (Peak Signal to Noise Ratio). The bit rate difference is considered between the proposed algorithm and the original unmodified HM encoder. The percentage ratio-time measure in Table 2 [9] was used to compare the encoding time between the proposed algorithm and the reference software.

## 5 Conclusions

This paper presents an improved transformed and quantized coefficient elimination system using two types of amplitude thresholding. The bitrate reduction for the proposed algorithm is visible in Table 2 from the BD rate parameter. Further investigations will be done in the future for the following TU sizes, to analyze the same approach for extended depths in the RQT structure.

**Acknowledgment.** This work was supported by a grant of the Romanian National Authority for Scientific Research and Innovation, CNCS/CCCDI - UEFISCDI, project number PN-III-P2-2.1-PED- 2016-1465, within PNCDI III.

## References

1. Bumshik, L., Jaehong, J., Munchurl, K.: An all-zero block detection scheme for low-complexity HEVC encoders. *IEEE Trans. Multimedia* **18**(7), 1257–1268 (2016)
2. Sung-Ho, B., Jaeil, K., Munchurl, K.: HEVC-based perceptually adaptive video coding using a DCT-based local distortion detection probability model. *IEEE Trans. Image Process.* **25**(7), 3343–3357 (2016)
3. Hanli, W., Han, D., Weiyao, L., Sam, K., Oscar, C., Jun, W., Zhihua, W.: Early detection of all-zero  $4 \times 4$  blocks in high efficiency video coding. *J. Vis. Commun. Image R* **25**, 1784–1790 (2014)
4. Wegner, K., Karwowski, D., Klimaszewski, K., Stankowski, J., Stankiewicz, O., Grajek, T.: Homogenous HEVC video transcoding by transform coefficient removal. In: *International Conference on Systems, Signals and Image Processing (IWSSIP)* (2017). ISBN: 978-1-5090-6344-4
5. H.264/AVC JM Reference Software, August 2008
6. ITU-T, Series H: Audiovisual and Multimedia Systems, H.265. High Efficiency video coding (2015)
7. ITU-T Recommendation H.265.2: Reference software for ITU-T H.265 high efficiency video coding
8. Wang, Z., Bovik, A.C., Sheikh, H.R., Simoncelli, E.P.: Image quality assessment: from error visibility to structural similarity. *IEEE Trans. Image Process.* **13**(4), 600–612 (2004)
9. Oprea, C.C., Udrea, R.M., Pirnog, I.: HEVC intra partitioning and mode decision using histograms of oriented gradients. In: *12th IEEE International Symposium on Electronics and Telecommunications (ISETC)* (2016)