MODE DEPENDENT VECTOR QUANTIZATION WITH A RATE-DISTORTION OPTIMIZED CODEBOOK FOR RESIDUE CODING IN VIDEO COMPRESSION

Bihong Huang[1,2]*, Felix Henry*[1]*, Christine Guillemot*[2]*, Philippe Salembier*[3]

[1]Orange Labs
4 Rue du Clos Courtel, 35512 Cesson Sevigne, France
[2]INRIA
Campus de Beaulieu, 35042 Rennes Cedex, France
[3]Universitat Politecnica de Catalunya
Jordi Girona 1-3, 08034 Barcelona, Spain

ABSTRACT

The High Efficiency Video Coding standard (HEVC) supports a total of 35 intra prediction modes which aim at reducing spatial redundancy by exploiting pixel correlation within a local neighborhood. In this paper, we show that spatial correlation remains after intra prediction, leading to high energy prediction residues. We propose a novel scheme for encoding the prediction residues using a Mode Dependent Vector Quantization (MDVQ) which aims at reducing the redundancy in residual domain. The MDVQ codebook is optimized in a rate-distortion (RD) sense. Experimental results show that the codebook can be independent of the quantization parameter (QP) with no loss in terms of coding efficiency. A bitrate reduction of 1.1% on average compared to HEVC can be achieved, while further tests indicate that codebook adaptivity could substantially improve the performance.

Index Terms— HEVC, vector quantization, residue coding, intra prediction

1. INTRODUCTION

The High Efficiency Video Coding (HEVC) standard [1] developed by the Joint Collaborative Team on Video Coding has been ratified as an international Video Coding Standard in 2013. HEVC contains several elements improving the coding efficiency of intra prediction over the H.264/AVC standard. A wider range of coding block size is supported using a flexible quad-tree block partitioning structure, which allows the splitting of an image into Coding Units (CU) with sizes from 8×8 to 64×64. By introducing a larger number of prediction modes, the intra prediction in HEVC can more accurately model smooth regions as well as directional structures. Inside a CU, one or four Prediction Units (PU) are defined, each of which specifies a region with an individual intra prediction mode. Up to 35 intra prediction modes are available for a PU. A CU is further split into a quad-tree of Transform Units (TU), on which transform, scalar quantization and entropy coding of residual signals are performed. Although the conventional intra prediction in HEVC is efficient at reducing the local spatial correlation in the pixel signals, the accuracy of intra prediction is limited in regions with complex textures or structures, so the residues in these regions have larger magnitudes. Vector quantization is a well-known technique which provides better performance than scalar quantization. However, it is unclear how to make use of it for residual signal compression and the expected gain is unknown. In this paper, we propose a novel approach to encode the intra prediction residues based on a Mode Dependent Vector Quantization (MDVQ). We will demonstrate that VQ is efficient at reducing the remaining correlation in the residual domain by using a special codebook for residuals of eachn intra prediction mode. Furthermore, one of the key points of our proposal is that these codebooks are learned from training sequences and are optimized in a rate-distortion sense. It is shown that the codebook does not need to be adapted for each Quantization Parameter (QP), which is advantageous for the simplicity of the design. Simulation results indicate that the method provides interesting bitrate savings, especially for low resolution sequences that are usually more difficult to compress.

The remainder of this paper is organized as follows. We introduce in the first part of Section 2 the method of residue coding using MDVQ in HEVC. In section 2.2 we present the MDVQ codebook training procedure based on Rate Distortion Optimization (RDO). The QP-independent codebook construction is described in Section 2.3. Experimental results are presented in Section 3. We finally conclude this paper and discuss the perspectives for future research in Section 4.

2. RESIDUE CODING USING MODE DEPENDENT VECTOR QUANTIZATION IN HEVC

The intra coding process of HEVC and H.264 constructs the intra predictor by extrapolating reference samples from pre-

viously reconstructed image blocks. In a block of samples predicted with planar mode or DC mode, the residual signals have a relatively homogeneous structure, whereas those derived from angular prediction modes tend to have directional structures. This residue characteristic has been previously investigated to improve coding efficiency. In [2], a directional transform and an adaptive coefficient scanning are used for coding the intra prediction residual. In our approach, a vector quantization codebook is learned with the aim of modelling the directional characteristics of the intra prediction residual signals.

Vector quantization [3] is an efficient data compression technique which exploits correlation in a vector of samples. Theoretically, better performance can always be obtained from coding vectors of source samples as a unit rather than individually. However, as the coding efficiency increases with the growth of the vector dimension, so does the size of required codebook and the complexity of searching for a best matching codeword. Since the codebook size is enormous for vectors of higher dimension, vector quantization is generally considered to be too complex to implement for practical use. We will demonstrate in the rest of this paper that, with our approach, good performance can be achieved for high dimensional vectors by using relatively small codebooks.

2.1. MDVQ-based residue coding

The proposed scheme of residue coding using vector quantization is shown in Fig. 1.

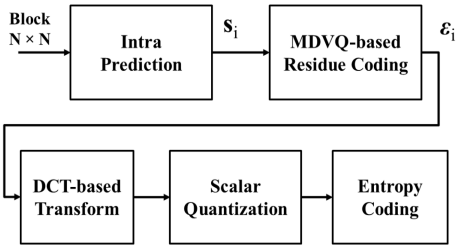

Fig. 1. The proposed approach of residue coding using mode dependent vector quantization

In our scheme, the MDVQ-based residue coding is performed after the intra prediction, and then followed by the usual steps of transform, quantization and entropy coding as in HEVC. Firstly, an $N \times N$ image block is intra predicted by mode $i \in M$ (the set of available modes) and the corresponding original residual signal $\mathbf{s}_i$ of the same dimension is computed. It is then transformed into a $1\times$ k dimensional vector where k$= N \times N$ and is quantized by MDVQ using a pre-generated codebook. We will explain later in Section 2.2 how to generate this codebook. The index of the matching codeword will be sent to the receiver and the final quantization error (the difference between the residual signal and the matching codeword) will be processed by a series of operations: DCT-based transform, scalar quantization and entropy coding.

Let us denote by $\mathcal{C}_i$ the VQ codebook for intra prediction mode i, and by $\mathbf{v} \in \mathcal{C}_i$ a codeword in this codebook. The conventional vector quantization approach aims at finding a codeword $\mathbf{v}^*$ which minimizes the distortion measured by the sum of square errors (SSE). Let $\mathcal{E}(\mathbf{s}_i, \mathbf{v})$ denote the quantization error of an input residual vector $\mathbf{s}_i$ quantized by a vector $\mathbf{v}$. The distortion measured on the prediction residue coded with MDVQ can be expressed as:

$$\mathcal{D}(\mathbf{s}_i, \mathbf{v}) = ||\mathbf{s}_i - \mathbf{v} - \mathcal{E}^{'}(\mathbf{s}_i, \mathbf{v})||^2 \quad (1)$$

where $\mathcal{E}^{'}(\mathbf{s}_i, \mathbf{v})$ denotes the reconstructed quantization error. Let $\mathcal{R}(\mathbf{v})$ represent the number of bits required for signaling the vector in the codebook, plus the bits needed to code the quantization error. The best matching vector in an RD optimization sense for a prediction residue block is the one minimizing the Lagrangian cost function [4] [5]:

$$\mathbf{v}^* = \arg\min_{\mathbf{v}\in\mathcal{C}_i} \mathcal{D}(\mathbf{s}_i, \mathbf{v}) + \lambda \cdot \mathcal{R}(\mathbf{v}) \quad (2)$$

where λ denotes the so-called Lagrange multiplier. MDVQ-based residue coding will be used for a given residue block if the rate-distortion trade-off given by this mode satisfies:

$$\mathcal{D}(\mathbf{s}_i, \mathbf{v}^*) + \lambda \cdot \mathcal{R}(\mathbf{v}^*) < \mathcal{D}(\mathbf{s}_i, \mathbf{s}_i') + \lambda \cdot \mathcal{R}_i \quad (3)$$

In other words, the MDVQ-based residue coding is performed only if the condition of Eq.(4) is satisfied.

2.2. RD optimized codebook construction

In the proposed method, the MDVQ codebooks are learned using training video sequences. The training set of residue vector are built by extracting from the training video sequences the actual set of residual vectors that satisfies Eq. (4). As a consequence, the training set is not "polluted" by the residual vectors that would not be coded by MDVQ. In more details an iterative approach is used in our codebook construction procedure, as follows.

Iteration 0:

- Step A: The training video sequences are encoded with the conventional HEVC mode (residue coding by MDVQ is not applied). Original residues $\mathbf{s}_i$ are gathered to form the training set for codebook construction.

- Step B: The codebooks are constructed with k-$means$ algorithm.

Iteration 1:

- Step A: The training video sequences are encoded with residue coding by MDVQ, using the codebooks generated at

the previous iteration. Original residues are extracted from TU blocks where residue coding by MDVQ is selected by RDO to build the training set for the next iteration. It should be noted that the MDVQ will only be selected for blocks where the conditions described by Eq.(3) and Eq.(4) are fulfilled.

- Step B: The codebooks are constructed with k-means algorithm.

Then the steps in Iteration 1 can be repeated several times. The whole procedure stops when a predefined number of iterations is reached.

2.3. QP independent codebook construction

When a video sequence is coded with different Quantization Parameters (QP) in HEVC, the coding modes such as intra prediction mode or block partitioning can change. For instance, at higher QPs, less bits are used to describe the predictor, leading to a lower quality prediction, and an increase of the average energy of the residue. In order to take this into account, in our early experiments, four groups of MDVQ codebooks for QP values of 22, 27, 32 and 37 were learned separately. However, we have observed that the codebooks learned for different QP values were actually very similar. As a consequence, a generic MDVQ codebook has been learned by including in the set of training residue vectors obtained with the different QP values. The tests have confirmed that the same coding performance could be achieved using the QP independent codebook compared to the groups of codebooks learned separately for the different QP values. Therefore, a generalized QP-independent MDVQ codebook can be used. This property is very useful since it helps reducing storage requirements, and simplifies the design of the codec: it is not necessary to scale the vectors to match a certain rate-distortion compromise.

2.4. New coding mode signalling

In the HEVC standard, each intra predicted TU block contains a vector of residual samples. In our experiments, the proposed approach was applied for coding TU blocks of size 4×4 and 8×8. Obviously, larger TU blocks for example of size 16×16 and 32×32 can adopt the VQ residual coding as well. However, this option is eliminated due to the increase of storage requirements and encoding complexity.

In reality, 4x4 TU blocks are much more frequently represented in 8×8 CUs than in 16x16 CU blocks, so we limit the residue coding by MDVQ on 4×4 TU blocks which reside inside a 8×8 CU block. For an 8×8 CU block which is split into four 4×4 TU blocks, we use three syntax elements and a tree structure for the signalization of residue coding by MDVQ. The syntax element $res\_vq\_cu\_flag$ indicates whether at least one of the four TU blocks inside the CU 8x8 block uses MDVQ. This flag is coded using one bit and an associated context-adaptive binary arithmetic coding (CABAC) context. When $res\_vq\_cu\_flag$ is set to 1 , for each 4×4 TU block a syntax element $res\_vq\_tu\_flag$ indicates whether MDVQ is used. This syntax element is also coded using one bit and an associated CABAC context. When MDVQ is used on a TU, a fixed-length syntax element res_vq_idx is transmitted, representing the index of the codeword in the codebook associated with the intra prediction mode and TU size. As the distribution of 8×8 TU blocks in different sizes of CU block are relatively uniform, only two syntax elements $res\_vq\_tu\_flag$ and res_vq_idx are signalling the use of MDVQ on 8×8 TU blocks. Except for the CU flag parsing step for 4×4 TU blocks, the reconstruction steps are shared between TU 4×4 blocks and TU 8×8 blocks.

3. EXPERIMENTAL RESULTS

In current version of implementation, the VQ residual coding is applied for TU blocks of size 4×4 and 8×8. The version with MDVQ for larger TU blocks is undergoing. For each of them, 35 codebooks corresponding to the 35 intra prediction modes have been derived using the codebook construction procedure described in section 2.2. Each codebook contains 256 codevectors, which represents a good compromise between storage, complexity and performance. However our tests indicate that further performance gains can be obtained with larger codebooks.

The proposed method has been implemented in the reference software HM8.0 [6] of HEVC. The encoder is configured according to the JCT-VC common test conditions with the "all intra" profile. Sequences of six test classes with different resolutions specified in HEVC [7] were encoded. The experiments are performed under the mid-bitrate range of quantization parameters: 22, 27, 32 and 37. The BD-Rate values are measured with the method in [8]. The performance is evaluated by comparing our method against HM8.0, where negative values indicate a bitrate saving.

3.1. Training with test sequences

In order to evaluate the potential of the learning method in the ideal case where the codebook is well suited to the input sequence, we have first learned the codebook using training vectors from the same sequence. Table 1 shows the performance of our method on class B sequences using this approach. In this ideal - but not realistic - case, one can observe that MDVQ can provide average bitrate savings of 4.9% on class B.

3.2. Training with external sequences

Table 2 shows the performance when the MDVQ codebooks are trained on sequences that are different from those used to measure the performance. This is the realistic use case. One can observe an average bitrate saving of 1.1%. Interestingly,

sequence class	bit-rate saving (%)
Kimono1_1920x1080p	−0.25
ParkScene_1920x1080p	−3.19
Cactus_1920x1080p	−13.66
BQTerrace_1920x1080p	−1.99
BasketballDrive_1920x1080p	−5.16
average of class B	−4.95

Table 1: Bitrate savings using sequence-dependent codebooks

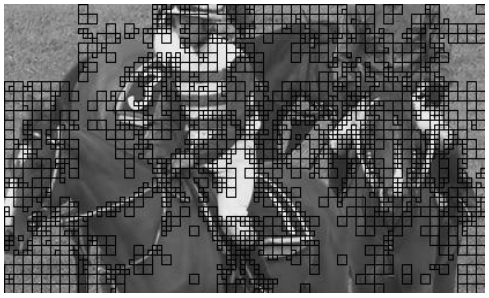

Fig. 2. Blocks on which MDVQ-based residue coding is performed

a larger gain is observed for low-resolution sequences which are usually more difficult to compress, this being attributed to the larger proportion of 4×4 and 8×8 TU blocks. The method also performs well on videoconference (class E) and screen content (class F) sequences. The higher performance shown in section 3.1 indicates that gains may be obtained with a realistic adaptive codebook approach. We show in Fig. 2 for the sequence RaceHorse of class D, the blocks on which the MDVQ-based residue coding is performed.

3.3. Complexity aspects

The proposed approach requires that codebooks are stored in the encoder and decoder. In general, vector quantization tends to impose a much larger complexity burden on the encoder because of the search for the best code vector than it does on the decoder where a simple table look-up is needed. Therefore, it can be viewed as a tool that the encoder can optionally choose to use to further improve the compression, at the expense of some complexity, while the decoder can always easily support it. On the docoding side, the proposed VQ introduces simple additional steps: retrieving and adding the code vector. Moreover, there is less informations related to the final residuals have to be decoded. Overall, the execution time of decoding increases by only 3% on average compared with the reference software HM8.0. We have not investigated the complexity on the encoder side, since the encoding can be accelerated by using fast vector quantization methods such as tree-structured VQ or other encoder-side optimization. For instance, to reduce the complexity, the use of MDVQ can be limited to the most frequent intra prediction modes.

sequence class	bit-rate saving (%)
NebutaFestival	−0.09
PeopleOnStreet	−1.62
SteamLocomotiveTrain	0.44
Traffic	−1.51
average of class A	−0.7
BasketballDrive	0.12
BQTerrace	−1.33
Cactus	−1.03
Kimono	0.02
ParkScene	−1.44
average of class B	−0.5
BasketballDrill	−2.27
BQMall	−1.34
ParkScene	−0.95
RaceHorse	−0.52
average of class C	−1.3
BasketballPass	−1.26
BlowingBubbles	−1.24
BQSquare	−0.98
RaceHorse	−1.68
average of class D	−1.3
FourPeople	−1.61
Johnny	−1.67
KristenAndSara	−1.2
average of class E	−1.6
BasketballDrillText	−2.53
ChinaSpeed	−1.29
SlideEditing	−0.91
SlideShow	−1.80
average of class F	−1.6

Table 2: Bitrate savings using sequence-independent codebooks

4. CONCLUSION

We have presented a new approach for coding the intra prediction residues in an HEVC based encoder. This novel method can also be seen as a second-order prediction which aims to further remove remaining correlation in first-order intra prediction residues. The method is based on vector quantization with a codebook tailored for each intra prediction mode. Each codebook is hence learned considering a set of residual signals obtained when using a specific intra prediction mode. This training set of residual signals is further limited to the blocks for which the VQ will bring improved RD performance when compared to the reference HEVC encoding. Finally, these codebooks are QP independent, limiting the coder and decoder storage requirements.

5. REFERENCES

[1] W.J. Han G.J. Sullivan, J.R. Ohm and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," in *Transaction on Circuits and Systems for Video Technology*. IEEE, December 2012, vol. 22, pp. 1646–1668.

[2] Y. Ye and M. Karczewicz, "Improved h.264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *IEEE Conf. Image Process*, October 2008, pp. 2116–2119.

[3] A. Gersho and R. M. Gray, "Vector quantization and signal compression," in *Kluwer Academic Press*. Springer, 1992, pp. 649–654.

[4] S.-W. Wu and A. Gersho, "Enhanced video compression with standardized bitstream syntax," in *Proc. IEEE Int. Conf. Acoust., Syst., Signal Process*. IEEE, April 1993, pp. 103–106.

[5] D. Mukherjee T. G. Campbell T. Wiegand, M. Lightstone and S. K. Mitra, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging h.263 standard," in *Transaction on Circuits and Systems for Video Technology*. IEEE, April 1996, vol. 6, pp. 182–190.

[6] HEVC Test Model (online), "Available: http://hevc.hhi.fraunhofer.de/svn/svn_hevcsoftware," .

[7] F. Bossen, "Common test conditions and software reference configurations," in *Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T VCEG and ISO/IEC MPEG*, Febery 2008.

[8] G. Bjøntegaard, "Calculation of average psnr differences between rd-curves," in *ITU-T SG16 Q.6 Document*. VCEG-M33, April 2008.