A transform size based quantization for high efficiency video coding

Yangang Cai, Hanfei Xu, Xufeng Li, Ronggang Wang
School of Computer & Information Engineering, Peking University Shenzhen Graduate School
xiaoc@pku.edu.cn, hfxu@szpku.edu.cn, lixf@pkusz.edu.cn, rgwang@pkusz.edu.cn

Abstract. Quantization is a key component in High Efficiency Video Coding (HEVC) standard. It enhances coding efficiency, and also introduces quantization distortion which causes degradation of the quality of the reconstructed video. In current framework, HEVC has adopted more transform sizes, and it is obvious that the transform unit (TU) with different transform size has different feature. A TU-level quantization technique, namely an adaptive quantization step size Q along with a fixed transform size is proposed in this paper. The proposed algorithm is implemented in HM software. Compared with HM anchor, the proposed TU-level quantization step adjustment (TUQA) method achieves an average of 0.32%, 2.34% and 2.58% BD bitrate saving for luminance and chrominance components, respectively. The largest BD bitrate saving is up to 1.0% for luminance and 3.8% for chrominance.

1. Introduction
High Efficiency Video Coding (HEVC) is emerging as a new video coding standard with significant coding improvement and has achieved significant coding efficiency improvement beyond existing video coding standard, e.g. MPEG-2 and H.264/AVC, by employing many new coding tools. In the framework, HEVC has adopted more improved coding tools, e.g. large size coding unit (CU), prediction unit (PU) and transform unit (TU), rate distortion optimized quantization (RDOQ[1]) etc. A concept for variable block-size transform (ABT) coding is presented in[2]. In current framework, many coding systems have adopted more transform size, such as 4x4, 8x8, 16x16, 32x32. As in H.264/MPEG-4 AVC, uniform reconstruction quantization (URQ) is used in HEVC, with quantization scaling matrices supported for the various transform block sizes. To reduce the memory needed to store frequency-specific scaling values, only quantization matrices of sizes 4x4 and 8x8 are used[3].

Many quantization step adjustment methods are widely used in video coding system: frame-level quantization step adjustment, macroblock-level quantization step adjustment, subband-level quantization step adjustment, adaptive quantization and reconstructed level (adaptive rounding) and Rate-Distortion optimized transform-coefficient-level adaptive quantization and reconstructed level. In Test Model 5 (TM5)[4], the problem is separated into three steps: 1) to allocate bits to each picture according to image activities and buffer fullness, 2) to set nominal slice quantization parameters to meet the target rate, and 3) to derive MQUANT of each MB from its nominal QP according to its activities to exploit spatial masking effect. Sullivan developed an expected-value based technique in H.264/AVC encoding system [5]. Moreover, a transform-unit-level quantization step adjustment (TUQA) scheme based on transform size is proposed for video coding quantization.

The remainder of this paper is organized as follows. In Section II, a brief review of the High Efficiency Video Coding (HEVC) quantization is provided. The technical details of the proposed
TUQA and the statistics diagram that different quantization step size with different TU size are introduced in Section 3. Section 4 shows the extensive experimental results and the better vector we used in the proposed method. Finally, this paper is concluded in Section 5.

2. Quantization in HEVC

For quantization, HEVC uses essentially the same URQ scheme controlled by a quantization parameter (QP) as in H.264/MPEG-4 AVC. The range of the QP values is defined from 0 to 51, and an increase by 6 doubles the quantization step size such that the mapping of QP values to step sizes is approximately logarithmic. Quantization scaling matrices are also supported[3].

\[
Q_{\text{step}}(QP) = f(QP\%6) \times (QP/6) / 2^{14}
\]

(1)

Where,

\[
f(x) = \{26214,23302,20560,18396,16384,14564\}, \ x=0,...,5
\]

As describe in[6], quantizer and dequantizer in HEVC can be implemented as follows:

\[
y = \text{round}((\text{coef} \times Q) >> (21 + QP/6 - M - (B - 8)))
\]

(2)

\[
\text{coef'} = \text{round}((y' \times IQ < QP/6) >> (M - 1 + (B - 8)))
\]

(3)

Where \( B \) is the internal bit depth (as specified by InternalBitDepth in the common config files). \( M \) is the logarithm to base 2 of the transform size. \( Q \) and \( IQ \) are specified as follows:

\[
Q = f(QP\%6), \ IQ = g(QP\%6)
\]

(4)

\[
g(x) = \{40,45,51,57,64,72\}, \ x=0,...,5
\]

(5)

3. Transform-Unit-Level Quantization Step Adjustment Scheme

In this section, we start with several motivating observations which provide useful guidelines for introducing the proposed algorithm. Then the detailed implementation of the proposed transform-unit-level quantization step adjustment scheme for quantization is described. Furthermore, to obtain the optimal group of vector candidates, the training process is analysed.

3.1. Observations the relationship of TU size and Qstep

Although RDOQ efficiently improves the quantizer efficiency by employing different values in transform-coefficient-level, in our simulations, it is observed that even in the same transform unit size with different Qstep, the BD-rate always presents different statistical characteristics. To verify this case, several BD-rate of transform unit size with different Qstep are shown in Fig. 1. As it shows, different TU size has the different better QPoffset. The detail of the statistical characteristics is shown in Table I.
TABLE I. BD-Rate(%) Of Tu Size With Different Qpoffset

| Quoffset (100 frames) | -6 | -5 | -4 | -3 | -2 | -1 | 0  | 1  | 2  | 3  | 4  | 5  | 6  |
|----------------------|----|----|----|----|----|----|----|----|----|----|----|----|----|
| 32x32                |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Y                    | 0.8| 0.5| 0.2| 0.1| -0.1|0.0| 0.1| 0.2| 0.5| 0.6| 0.7| 0.9|    |
| U                    | 2.4| 2.5| 2.3| 1.8| 1.1 |0.4| 0.0| 0.2| 0.0| 0.6| 0.8| 0.9| 1.2|
| V                    | 2.4| 2.5| 2.3| 1.8| 1.1 |0.5| 0.0| 0.0| 0.1| 0.6| 0.7| 1.0| 1.2|
| 16x16                |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Y                    | 1.3| 0.9| 0.5| 0.2| 0.0 |0.1| 0.0| 0.1| 0.2| 0.4| 0.6| 0.7| 0.9|
| U                    | 2.2| 2.4| 2.3| 1.7| 0.9 |0.0| 0.0| 0.8| 2.1| 3.3| 4.7| 6.0| 7.2|
| V                    | 2.5| 2.5| 2.3| 1.7| 1.0 |0.0| 0.0| 0.8| 2.0| 3.2| 4.4| 5.9| 6.9|
| 8x8                  |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Y                    | 1.9| 1.3| 0.8| 0.4| 0.0 |0.0| 0.2| 0.4| 0.7| 1.0| 1.4| 2.4|    |
| U                    | 2.3| 2.9| 2.8| 2.5| 1.6 |0.5| 0.0| 0.2| 1.3| 2.3| 3.7| 5.0| 7.7|
| V                    | 2.5| 2.8| 2.9| 2.5| 1.7 |0.7| 0.0| 0.4| 1.0| 2.3| 3.3| 4.5| 6.3|
| 4x4                  |    |    |    |    |    |    |    |    |    |    |    |    |    |
| Y                    | 2.4| 1.6| 1.0| 0.5| 0.1 |0.0| 0.0| -0.1|0.5| 1.0| 1.6| 2.3| 3.1|
| U                    | -2.8| -3.9| -4.3| -4.6| -3.9| -2.2| 0.0| 2.9| 6.8| 11.0| 15.9| 20.1| 24.2|
| V                    | -3.6| -4.5| -4.8| -4.7| -4.1| -2.4| 0.0| 3.0| 6.5| 11.0| 15.9| 20.1| 24.2|

The above observations imply the possibility to further improve the quantizer efficiency, and it leads us to the idea of using multiple vector candidates in quantizer which is naturally capable to accommodate the various TUs better. The candidate vector means that different TU size with different Qpoffset, for example, the vector (1,0,-1,-2) indicate that the 32x32 transform blocks have the QPoffset 1, the 16x16 transform blocks have the QPoffset 0, the 8x8 transform blocks have the QPoffset -1 and the 4x4 transform blocks have the QPoffset -2.

3.2. Proposed TU-level quantization step adjustment

To illustrate the distinctive elements of different methods of quantization step adjustment, frame-level quantization step adjustment, CU-level quantization step adjustment, Rate-Distortion optimized transform-coefficient-level adaptive quantization and the proposed TU-level quantization step adjustment(TUQA) are compared in Fig.2.

- In frame-level quantization step adjustment (Fig.2(b)) scheme, the quantizer uses different Qstep in the different coding layer.
- In CU-level quantization step adjustment (Fig.2(c)) scheme, the quantizer uses different Qstep in the different coding unit by adopting a QPoffset.
- In Rate-Distortion optimized transform-coefficient-level adaptive quantization (Fig.2(d)), the quantizer adjusts the quantized value q=q ± δ to achieve a better performance.
- In the proposed TU-level quantization step adjustment (Fig.2(e)) scheme, the quantizer uses different Qstep in different transform unit by adopting a QPoffset.
In the proposed TUQA, there are $K$ vector candidates for quantizer, however, the explicit signaling of vector indexes inevitably introduces overhead bits for each block. So a fixed quantization step size $Q$ along with a transform size is used in TUQA. If the vector is $(\Delta \alpha_1, \Delta \alpha_2, \Delta \alpha_3, \Delta \alpha_4)$, then the QP would be changed by following the equation:

$$Q_P = Q_P + (\Delta \alpha_1, \Delta \alpha_2, \Delta \alpha_3, \Delta \alpha_4)$$  \hspace{1cm} (6)

So the proposed TUQA doesn't require additional signaling of the vector indexes between encoder and decoder, such that the decoding process can also be correctly implemented without any mismatch. Within the framework of TUQA, the vector can be fixed in encoder and decoder.

3.3. Analysis and implementation of the training process

Within the framework of our proposed TUQA, there is still one important issue which has not been discussed but evidently affects the coding performance. How to obtain the optimal group of vector candidates? In the following, we will focus on this issue and present both in-depth analysis and detailed technical design on the optimal group of vector candidates.

1) Train the optimal vector in TUQA

To measure the efficiency of a certain vector, the BD-rate criterion is employed [7] and introduced in the following. The basic elements are:

- Fit a curve through 4 data points (PSNR/bitrate are assumed to be obtained for QP = (22,27,32,37).
- Based on this, use an expression for the integral of the curve.
- The average difference is the difference between the integrals divided by the integration interval.
An interpolation curve through 4 data points can be obtained by:

\[ SNR = \frac{(a + b \times \text{bit} + c \times \text{bit}^2)/(\text{bit}+d)}{7} \]

Where \(a, b, c, d\) are determined such that the curve passes through all 4 data points. With logarithmic bitrate scale the interpolation can also be made more straightforward with a third order polynomial of the form:

\[ SNR = a + b \times \text{bit} + c \times \text{bit}^2 + d \times \text{bit}^3 \]

In this way, the following both can be found:

- Average PSNR difference in dB over the whole range of bitrates.
- Average bitrate difference in % over the whole range of PSNR.

To train the optimal vector in TUQA, we would search in the range of (-6,6) of QP offset and use the BD-rate as the criterion.

\[(\Delta a_1, \Delta a_2, \Delta a_3, \Delta a_4) = \text{ArgminBDrate}(QP(x,y,z,k))\]

\[x,y,z,k \in (-6,6)\]

Where the \(QP(x,y,z,k)\) means the vector \((x,y,z,k)\) is used in TUQA. The time used in encoder is shown in Table 2. In the worst case the time we used should be \(6^2 \times 58862.67 = 76286020.32\) s = 882.94 days. In the next subsection, a divided vector training process will be developed for searching a sub-optimal solution of (9).

2) Simple ways to train the optimal vector in TUQA

The proposed method is to divide the vector into four vectors as the following equation firstly:

\[(a,b,c,d) = (\alpha,0,0,0) + (0,\beta,0,0) + (0,0,\gamma,0) + (0,0,0,\eta)\]

The proposed training method for generating the vector candidates steps are introduced details in the following items:

- **Step 1**: Assign the vector \((\alpha, 0, 0, 0), (0, \beta, 0, 0), (0,0,\gamma,0), (0,0,0,\eta)\) to 32x32, 16x16, 8x8, 4x4 of transform unit.
- **Step 2**: For each vector, calculate the optimal BD-rate of the sequence, and update the best value in the vector.
- **Step 3**: Gain a sub-optimal solution of the vector \((a,b,c,d)\).

4. Experimental results

In this section, extensive experiments are conducted to verify the performance of the proposed algorithm. We implement our algorithm in HM software. Simulations are conducted with the basic test sequence[8]. The performance of original software HM is used as an anchor, and the common test condition is defined in[9]. We carry out our experiments on the random access, which is denoted as HE_RA. For each video sequence, four quantization parameter values are tested: 22, 27, 32 and 37. All the frames in the test sequences are encoded. The BD-rate is calculated according to the method in[10].

From the experimental result in TABLE II, we can see that our proposed algorithm improves the quantization performance in all the test sequence. It achieves about 0.32%, 2.34% and 2.58% BD bitrate saving for luminance component and chrominance component, respectively. The largest BD bitrate saving is up to 1.0% for luminance and 3.8% for chrominance. The improvement is larger on high resolution sequences. The reason is that the high resolution sequences are usually selected more TU size.

| Sequence          | Y   | U   | V   |
|-------------------|-----|-----|-----|
| Traffic 2560x1600_30 | -0.3% | -2.0% | -2.7% |
5. Conclusion
This paper proposes a TU level adaptive quantization scheme for HEVC standard, and the concept of TUQA for application in HEVC was presented. The choice of an appropriate QP parameter is made according to the BD-rate and TU size. Extensive simulations show that, compared to HEVC, the proposed method achieves significant coding gain for a wide range of video resolutions. Compared with HM anchor, the proposed TUQA method achieves an average of 0.32%, 2.34% and 2.58% BD bitrate saving for luminance, chrominance component, respectively.

| Video Name          | Luminance BD-Rate | Chrominance BD-Rate | Average BD-Rate |
|---------------------|-------------------|---------------------|-----------------|
| PeopleOnStreet_2560x1600_30 | 0.3%              | -1.9%               | -2.4%           |
| ParkScene_1920x1080_24    | -0.2%             | -1.7%               | -2.2%           |
| Cactus_1920x1080_50      | -0.3%             | -2.5%               | -2.3%           |
| BQTerrace_1920x1080_60   | -0.5%             | -5.1%               | -7.0%           |
| BasketballDrill 832x480_50| 0.0%              | -2.3%               | -2.1%           |
| BQ Mall_832x480_60       | -0.1%             | -2.2%               | -2.1%           |
| PartyScene_832x480_50    | -0.3%             | -2.1%               | -2.4%           |
| BQ Square_416x240_60     | -0.4%             | -3.5%               | -2.5%           |
| BlowingBubbles_416x240_50| -0.3%             | -2.1%               | -2.8%           |
| RaceHorses_416x240_30    | 0.2%              | -1.5%               | -1.3%           |
| FourPeople_1280x720_60   | -0.9%             | -2.7%               | -2.5%           |
| Johnny_1280x720_60       | -0.8%             | -3.4%               | -3.9%           |
| KristenAndSara_1280x720_60| -1.0%             | -2.7%               | -3.8%           |
| SlideEditing_1280x720_30 | -0.1%             | 0.3%                | 0.2%            |
| SlideShow_1280x720_20    | -0.4%             | -2.0%               | -1.5%           |
| average               | -0.32%            | -2.34%              | -2.58%          |

References
[1] M. Karczewicz, Y. Ye, and I. Chong, Rate distortion optimized quantization, VCEG document VCEG-AH21, 34th VCEG Meeting, Antalya, Turkey, Jan. 2008.
[2] Wien, M., “Variable block-size transforms for H.264/AVC,” Circuits and Systems for Video Technology, IEEE Transactions on, vol.13, no.7, pp.604,613, July 2003
[3] Sullivan, G.J.; Ohm, J.; Woo-Jin Han; Wiegand, T.; Wiegand, T., “Overview of the High Efficiency Video Coding (HEVC) Standard,” Circuits and Systems for Video Technology, IEEE Transactions on, vol.22, no.12, pp.1649,1668, Dec. 2012
[4] Wei Ding, “Rate control of MPEG video coding and recording by rate-quantization modeling,” Circuits and Systems for Video Technology, IEEE Transactions on, vol.6, no.1, pp.12,20, Feb 1996
[5] G. Sullivan, “Adaptive quantization encoding technique using an equal expected-value rule,” ITUT/ISO/IEC document no. JVT-N011, Hongkong, Jan. 2005
[6] A. Fuldseth, Cisco Systems, “Core transform design for HEVC”, ISO/IEC JTC1/SC29/WG11 document G495, Nov. 2011.
[7] G. Bjontegaard, “Calculation of Average PSNR Differences Between RD Curves”, document VCEG-M33, ITU-T SG16/Q6, 13th VCEG Meeting, Austin, TX, Apr. 2001.
[8] Test sequences are available on ftp://hevc@ftp.tnt.uni-hannover.de/testsequences/
[9] Frank Bosse, “Common test conditions and software reference configurations,” ITU-T SG16 Contribution, JCTVC-E700, Daegu, Jan. 2011.
[10] S. Pateux and J. Jung, An Excel Add-in Computing Bjontegaard Metric and its Evolution, ITU-T SG16 Q.6 Document, VCEG-AE07, Mar- rakech, Jan. 2007.