IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 2, 2019, accepted May 13, 2019, date of publication May 16, 2019, date of current version May 24, 2019.

*Digital Object Identifier 10.1109/ACCESS.2019.2917260*

# Joint Optimization of Transform and Quantization for High Efficiency Video Coding

**MIAOHUI WANG[1], (Member, IEEE), WUYUAN XIE[2], (Member, IEEE), JIAN XIONG[3], (Member, IEEE), DAYONG WANG[4], (Member, IEEE), AND JING QIN[5], (Senior Member, IEEE)**

[1]Guangdong Key Laboratory of Intelligent Information Processing, National Engineering Laboratory for Big Data System Computing Technology, College of Information Engineering, Shenzhen University, Shenzhen 518060, China
[2]College of Computer and Software Engineering, Shenzhen University, Shenzhen 518060, China
[3]College of Communication and Information Technology, Nanjing University of Posts and Telecommunications, Nanjing 210023, China
[4]Institute of Bioinformatics, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
[5]School of Nursing, The Hong Kong Polytechnic University, Hong Kong

Corresponding author: Wuyuan Xie (wuyuan.xie@gmail.com)

This work was supported in part by the National Natural Science Foundation of China under Grant 61701310 and Grant 61701258, in part by the Free Exploration Project for Basic Research of Shenzhen City under Grant JCYJ20180305124209486, in part by Shenzhen University Start-Up Grant 2018080, in part by the Chongqing Science and Technology Commission Project under Grant CSTC2016JCYJA0543, in part by Natural Science Foundation of Jiangsu Higher Education Institutions Grant 17KJB510044, and in part by the Hong Kong Innovation and Technology Commission under Grant ITS/319/17.

**ABSTRACT** In *high efficiency video coding* (HEVC), transformation and quantization are separately performed to eliminate the perceptual redundancy of visual signals. However, a uniform quantizer can inevitably degrade the compression efficiency of fixed transform matrices due to varying space-frequency characteristics of video content. This paper introduces a joint optimization of transform and quantization approach for video coding. First, we compute a content dependent transform from the reconstructed reference by a fast Karhunen-Loéve transform (KLT). Second, using a template-based rate regularization, we jointly optimize transform and quantization (JOTQ) as a rate constrained optimization problem and obtain a feasible solution to improve coding performance. Finally, we design fast algorithms and early terminations to reduce the computational complexity of JOTQ. The experimental results show that JOTQ outperforms several previous methods by providing Bjøntegaard Delta rate reductions of 4.11% and 3.38% on average under the low-delay and random-access configuration, respectively.

**INDEX TERMS** Video coding, content dependent transform, block adaptive quantization, high efficiency video coding (HEVC).

## I. INTRODUCTION

With the rapid development of Internet of Video Things (IoVT), video applications [1]–[3] (*e.g. video-on-demand* (VoD), *live over-the-top* (OTT), *video surveillance*, *etc.*) have become ubiquitous, which also promotes the fast development of video coding techniques [4], [5]. The ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) jointly released H.264/Advanced Video Coding (AVC) [6] and High Efficient Video Coding (HEVC) [7] in the past two decades. However, basic transform and quantization methods have not changed much. For example, H.264/AVC uses integer discrete cosine transform (DCT), while HEVC adopts almost the same scheme, except for 4×4 intra-coding using discrete sine transform (DST). Moreover, both of them adopt uniform reconstruction quantization (URQ), except that HEVC employs an additional rate distortion optimized quantization (RDOQ) method.

HEVC takes a quadtree-based partitioning scheme to divide similar content into a coding block (CB), which boosts the content adaptability performance of an encoder. To enhance the coding performance of prediction, it employs more flexible prediction block (PB) partitions, and more intra-prediction directions. HEVC supports various transform block (TB) sizes to compress the energy of prediction residue. In URQ, it adopts different dead-zone (DZ) offsets for intra- and inter-mode coding when the same quantization parameter (QP) is applied at the block level. Furthermore, HEVC uses a hierarchical QP cascading (QPC) approach, where frames in a lower layer are encoded by smaller QPs, and

The associate editor coordinating the review of this manuscript and approving it for publication was Nilanjan Dey.

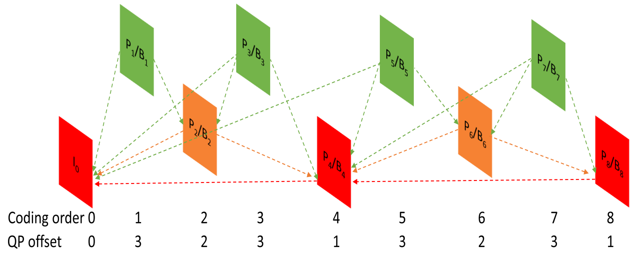

**FIGURE 1. Example of hierarchical low-delay P and B structure in HEVC. Frames in lower layers are used as reference for higher layers. A "P" frame is encoded by uni-prediction, while a "B" frame is encoded by bi-prediction.**

**TABLE 1. Main abbreviations used in this work.**

| | |
|---|---|
| PB | Prediction block (PB) is the basic block for intra- or inter-prediction. |
| TB | Transform block (TB) is the basic block to perform a specific transform, such as discrete cosine transform. |
| CB | Coding block (CB) is the basic block for prediction and transform coding, which can be split to multiple PBs or TBs. |
| QP | Quantization parameter (QP) is used to control the quantizer, ranging from 0 to 51 in HEVC. |
| DZ | Dead-zone (DZ) is used to control the range of the coefficients being quantized to zero. |

hence high-quality reconstructions can be used as reference frames for a higher layer (see Fig. 1). Specifically, a QP offset value between adjacent layers is set as one [8]. Since the legacy video coding standards take a "first transform then quantization" strategy, it is beneficial to review recent studies that improve the performance of transform or quantization in video coding. We summarize some main abbreviations of this work in Table 1.

Various transform methods have been designed to better compress the energy of residual blocks, which can be roughly categorized into directional transform [9], [10], non-square transform [11], secondary transform [12], [13], and content dependent transform [14]–[16]. Directional transform is based on that DCT is less efficient in the edge region (except vertical and horizontal edges), and 1D transform along edge direction is used to address this problem. Non-square transform is designed for non-square PBs, where it is reported that splitting non-square PBs and applying to the non-square transform can improve the coding performance by 1.0% BD-rate (Bjøntegaard Delta rate [17]) reduction. Secondary transform is used to further compact the coefficient energy of the primary transforms (*i.e.*, DCT), where the energy is clustered in the low-frequency region after performing a primary transform, and then a secondary transform is conducted on the primary result. Experiments show that secondary transform can provide about 0.5-2.5% BD-rate saving.

Karhunen-Loé ve transform (KLT) is considered as an optimal content dependent transform, which utilizes online transform matrices to compensate the coding loss of a fixed transform. One key problem of KLT is that transform matrices are required to send to a decoder, which results in overhead bits problem [15]. To address this overhead problem, Biswas *et al.* [14] proposed to rotate motion compensated prediction block by some certain degrees (e.g. −0.5°), and then shift it horizontally (or vertically) to produce training dataset which was used to compute transform matrices by KLT. Wang *et al.* [15] proposed to down-sample a residual block into four equal-sized sub-blocks. The first sub-block was transformed by DCT, while the other three sub-blocks were de-correlated by transform matrices that were estimated from the first reconstructed sub-block by Singular Value Decomposition (SVD). Lan *et al.* [16] proposed to establish a training dataset by searching for similar blocks in a region that are used to compute transform matrices by KLT. Since searching similar content is computationally expensive, it greatly increases encoding and decoding complexity.

In addition, many efforts [18]–[24] have been dedicated to explore a better quantizer for video coding. Sullivan [18] introduced an adaptive rounding-based quantizer, where the adaptive rounding could first obtain the statistics of residual coefficients and then adjust the rounding offsets accordingly. Karczewicz *et al.* [19] used RDOQ to obtain an optimal quantized level for a transform coefficient among a ceiling rounding value, zero and a floor rounding value. Yu *et al.* [20] proposed to use a hard decision partition and adaptive reconstruction level to do quantization, where the reconstruction levels were computed adaptively based on the statistics of reconstructed residual blocks. Lee *et al.* [21] introduced a soft thresholding quantization method by adjusting multiplication factor (MF) in URQ, where MF was weighted by Euclidean distance between the current coefficient and DC coefficient. Wang *et al.* [22] proposed an optimized quantization method for motion compensation prediction (MCP) residual coding, where an adaptive QP is computed for each block. Ropert *et al.* [23] proposed a spatial-temporal scheme to compute a local quantization parameter based on a temporal distortion propagation model. Xiang *et al.* [24] proposed to estimate adaptive quantization by a spatial-temporal just noticeable distortion (JND) model, where both bit-rate saving and subjective quality can be improved.

As can be seen from the above discussions, the existing works mainly focus on how to improve compression efficiency by individually optimizing transform or quantization to reduce the perceptual redundancy [25]. However, the transform coding gain can be degraded by a quantizer due to the "first transform then quantization" strategy, as residual blocks usually exhibit varying space-frequency characteristics [22]. To address this challenge, we propose to jointly optimize the conventional transform-quantization (T-Q) paradigm with the goal of finding optimal quantized transform coefficients (QTCs).

The main contributions of this study comparing with the existing publications are summarized as follows.

1) We design a content adaptive transform (CAT) that can compute online transform matrices from some reconstructed similar blocks by a fast KLT decomposition, which avoids sending block-based transforms but preserves data dependent merit.

2) By employing a template-based rate regulation in block adaptive quantization (BAQ) optimization, we jointly optimize transform and quantization (JOTQ) as a rate constrained optimization problem, and obtain a feasible solution which outperforms previous methods for various video sequences.
3) We introduce fast algorithms and early termination strategies which can reduce the computational complexity of JOTQ.

The reminder of this study is organized as follows. Section II gives problem formulations used in the proposed JOTQ. Section III demonstrates the details of JOTQ. Section IV presents experimental results. Section V concludes this work.

## II. BACKGROUND AND PROBLEM FORMULATION

In this section, we introduce some necessary background, and establish the formulations of transform and quantization used in the proposed JOTQ model.

### A. ADOPTED TRANSFORMS IN HEVC

Suppose that $\boldsymbol{x}_i$ is a row-ordered column vector of the $i$-th residual block $\boldsymbol{X}^i_{N\times N}$, $\boldsymbol{C}$ is a 2D orthogonal transform matrix with the size of $N \times N$, and $\boldsymbol{y}_i$ is a row-ordered column vector of the transform coefficient block $\boldsymbol{Y}^i_{N\times N}$. Then, in video coding, transform coding can be formulated as

$$\boldsymbol{y}_i = (\boldsymbol{C} \otimes \boldsymbol{C}) \times \boldsymbol{x}_i \tag{1}$$

where $\otimes$ is the Kronecker product.

In HEVC, $\boldsymbol{C}$ takes an approximated integer implementation of core transform [26]. In order to improve the coding performance, there are multiple transform sizes as well as transform types. For instance, $\boldsymbol{C}$ supports four DCT matrices of size 4×4, 8×8, 16×16, and 32×32 in the context of motion compensated video coding. Meanwhile, $\boldsymbol{C}$ includes an alternate DST matrix for the encoding of 4×4 luma intra-prediction residue.

#### 1) DISCRETE COSINE TRANSFORM (DCT)

The forward orthogonal DCT-II can be expressed as

$$\boldsymbol{C}^{m,n}_{dct} = \frac{\sigma_m}{\sqrt{N}} \cos\left(\frac{m\pi}{N}\left(n+\frac{1}{2}\right)\right), \tag{2}$$

where $m, n = 0, 1, \ldots N-1$ and $\sigma_m$ is equal to 1 when $m = 0$, otherwise $\sigma_m$ is equal to $\sqrt{2}$.

The integer approximation of Equation (2) is to scale each matrix element by $2^{6+log(N)/2}$. Keeping the first row of $\boldsymbol{C}_{dct}$ to be equal to 64, the rest matrix elements are carefully hand-tuned to achieve a balanced performance. A detailed description is provided in [27].

#### 2) DISCRETE SINE TRANSFORM (DST)

For a 4×4 luma intra-prediction block, the forward transform matrix is given by a fixed-point representation of DST-VII,

$$\boldsymbol{C}^{m,n}_{dst} = round\left(\frac{256}{\sqrt{2N+1}} \sin\left(\frac{(2m+1)(n+1)\pi}{2N+1}\right)\right). \tag{3}$$

In intra-prediction, prediction error tends to be bigger away from the left and/or top boundary due to the direction prediction scheme, which can be better modeled by a DST basis. A theoretical analysis of this phenomenon is given in [28]. DST can provide about 1% bit-rate saving.

### B. BLOCK ADAPTIVE QUANTIZATION

URQ can be expressed as

$$\begin{aligned} &\boldsymbol{Q}_{QP}(\boldsymbol{y}_i) \\ &= \boldsymbol{sgn}(\boldsymbol{y}_i) \cdot \left\lfloor \frac{\boldsymbol{Diag}(\Phi(QP)) \times |\boldsymbol{y}_i| + \boldsymbol{1} \cdot f}{2^{qbits}} \right\rfloor \\ &= \boldsymbol{sgn}(\boldsymbol{y}_i) \cdot \left\lfloor \frac{\boldsymbol{Diag}(\Phi(QP)) \times |(\boldsymbol{C} \otimes \boldsymbol{C}) \times \boldsymbol{x}_i| + \boldsymbol{1} \cdot f}{2^{qbits}} \right\rfloor, \end{aligned} \tag{4}$$

where $\boldsymbol{sgn}(\cdot)$ is a sign function, $\boldsymbol{Diag}(\cdot)$ is a diagonal matrix operation, $\boldsymbol{\Phi}(QP)$ is the row-ordered column vector of $\phi(QP)$ which is a map function of six constant multiplication factors related to an argument $QP$, $f$ is a constant which is used to adjust the width of DZ, $qbits$ is a right-shifted factor (*e.g.* $qbits = 21 - \log_2(N) + QP/6$), and $\boldsymbol{1}$ is a vector with the size of $N^2 \times 1$. According to (4), a vector result of quantized coefficient $\tilde{\boldsymbol{y}}_{i,QP} = \boldsymbol{Q}_{QP}(\boldsymbol{y}_i)$.

However, URQ is frequency independent, so all coefficients are uniformly quantized in a TB. To improve coding performance, BAQ can obtain the best QP for a coding block (CB) in a rate-optimization (R-D) sense, where the encoder needs to search over multiple QPs in a small range. If the number of QP candidates is bigger than 1 (*i.e.*, $\Delta QP > 0$), then it employs the following R-D optimization process,

$$\underset{QP \in \left[\substack{QP_b - \Delta QP, \\ QP_b + \Delta QP}\right]}{\arg\min} \boldsymbol{J}_i(QP) = D\left(\tilde{\boldsymbol{y}}_{i,QP}\right) + \lambda \cdot R\left(\tilde{\boldsymbol{y}}_{i,QP}\right), \tag{5}$$

where $\boldsymbol{J}_i$ is a R-D cost of the $i$-th CB, $R(\cdot)$ denotes a bit-rate cost, $D(\cdot)$ denotes a reconstruction distortion cost, $\lambda$ denotes the Lagrange multiplier, and $QP_b$ denotes a base QP.

Some previous studies [22], [29], [30] show that BAQ is helpful to enhance the compression performance. BAQ can provide the bit-rate reduction by about 1.0-2.0% [22], [30].

## III. PROPOSED JOINT OPTIMIZATION METHOD

### A. CONTENT ADAPTIVE TRANSFORM (CAT)

In this section, we introduce an efficient CAT method. The proposed CAT can be trained in a block-by-block way. CAT is designed to avoid encoding transform matrices while maintaining data dependent merit.

The CAT matrices can be computed as follows. First, suppose that we have $K$ residue vectors $\boldsymbol{x}^r_k$, $k = 1, 2, \ldots, K$. The mean values of $K$ residue vectors is $\boldsymbol{\mu} = \frac{1}{K}\sum_{k=1}^{K} \boldsymbol{x}^r_k$. Let $\boldsymbol{u}_k$=$\boldsymbol{x}^r_k$-$\boldsymbol{\mu}$. Then, we take these residual vectors $\boldsymbol{u}_k$ as the training samples to compute a data dependent transform.

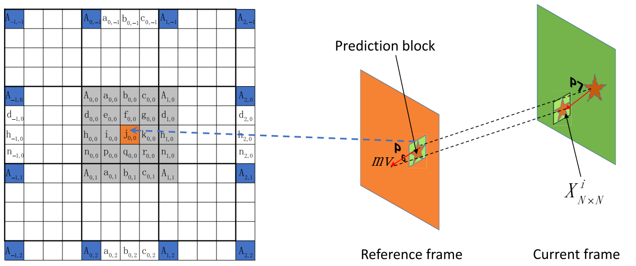

**FIGURE 2. Illustration of uni-prediction (P frame) for the luma motion compensation with 1/4 pixel accuracy and the construction of the training set $U$. The training data of frame P7 is collected from its reference frame P6.**

We denote the covariance matrix of those residual vectors as

$$\boldsymbol{R_u} = \boldsymbol{U}\boldsymbol{U}^T, \tag{6}$$

where $\boldsymbol{U} = (\boldsymbol{u}_1, \boldsymbol{u}_2, \cdots, \boldsymbol{u}_K)$ with the size of $N^2 \times K$.

Since $\boldsymbol{R_u}$ is real and symmetric, there is an orthogonal matrix $\boldsymbol{C}_{cat}$ that diagonalizes $\boldsymbol{R_u}$.

$$\boldsymbol{C}_{cat}^T\boldsymbol{R_u}\boldsymbol{C}_{cat} = \boldsymbol{\Lambda}, \tag{7}$$

where $\boldsymbol{\Lambda}$ is diagonal.

The performance of $\boldsymbol{C}_{cat}$ is based on the similarity between training vector $\boldsymbol{u}_k$ in $\boldsymbol{U}$ and $\boldsymbol{x}_i$. In order to obtain a better $\boldsymbol{C}_{cat}$ in Equation (7), we need to establish a high-quality training data $\boldsymbol{U}$. Searching similar $\boldsymbol{x}_k^r$ involves a large amount of computation. Our empirical studies suggest a light-weight estimation of $\boldsymbol{x}_k^r$. We introduce the detailed construction of $\boldsymbol{U}$, and computation of $\boldsymbol{C}_{cat}$ as follows.

### 1) TRAINING DATA $U$

For an inter-coding block, we first collect highly correlated blocks $\boldsymbol{u}_k$ to construct the training set $\boldsymbol{U}$. We then compute $K$ similar patches from a reference frame at a quarter-pixel level for the luma component. A symmetric 8-tap, $[-1, 4, -11, 40, 40, -11, 4, 1]/64$, for half-pixel accuracy and an asymmetric 7-tap filter, $[-1, 4, -10, 58, 17, -5, 1]/64$, for quarter-pixel accuracy are used to interpolate the fractional positions in this research.

More precisely, $\boldsymbol{U}$ is estimated by shifting the motion compensated predictor onto a small grid of pixels centered around the motion vector associated with $\boldsymbol{x}_i$. For uni-prediction, we collect $K$ shifting residues of $\boldsymbol{x}_k^r$ in the fractional positions. For bi-prediction, we also obtain $2K$ residues of $\boldsymbol{u}_k$ by the same way.

Fig. 2 shows an example of obtaining $\boldsymbol{u}_k$ for a P frame in uni-prediction. In the frame **P**7 of Fig. 1, the half-pixel sample $b_{0,0}$ is computed by the symmetric 8-tap filter. The quarter-pixel sample $a_{0,0}$ is computed by the asymmetric 7-tap filter. Suppose that the motion vector of a CB is at the center position (*i.e.*, $j_{0,0}$) of Fig. 2. Then the neighboring 5×5 candidate blocks are used to construct the training set $\boldsymbol{U}$.

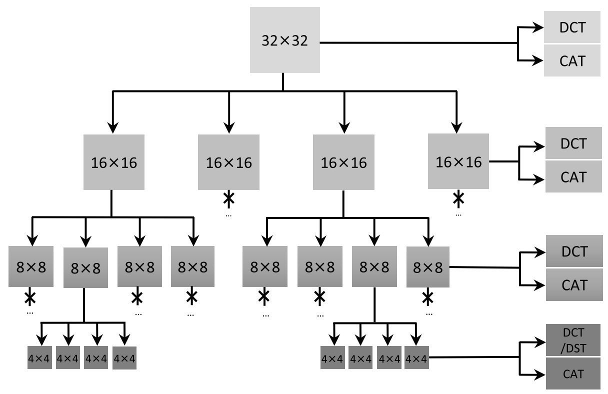

**FIGURE 3. Example of the proposed CAT framework.**

### 2) ONLINE MATRIX DECOMPOSITION AND INTEGER IMPLEMENTATION

To directly obtain $\boldsymbol{C}_{cat}$ that diagonalizes $\boldsymbol{R_u} \in R^{N^2\times N^2}$ can introduce intolerable complexity in video coding. We diagonalizes $\boldsymbol{R_u^T}$ because it has a much lower dimension, $\boldsymbol{R_u^T} \in R^{K\times K}$ when $K < N^2$. The eigenvectors of the original high dimensional covariance matrix $\boldsymbol{R_u}$ can be obtained by multiplying $\boldsymbol{U}$ with the eigenvectors of the lower dimensional covariance matrix $\boldsymbol{R_u^T}$. A detailed description is given in [16].

In order to reuse the quantization design of HEVC, we scale $\boldsymbol{C}_{cat}$ by $N \times 2^{10}$, and scale the transform entries of each to its nearest integer. Experiments validate the effectiveness of this design in Section IV-A.

### 3) SELECTION BETWEEN CAT AND DCT/DST

CAT is data dependent, whose performance is dependent on the similarity between the current block $\boldsymbol{x}_i$ to be transformed and the training data $\boldsymbol{U}$. It is noted that $\boldsymbol{U}$ has lower correlations with $\boldsymbol{x}_i$ in rich texture area where pixels vary significantly. To compensate for sub-optimal estimation of $\boldsymbol{U}$ in Equation (6), we employ an alternative method between CAT and DCT/DST.

We use the R-D optimization to determine the best coding mode between $\boldsymbol{C}_{cat}$ and $\boldsymbol{C}_{dct/dst}$. Furthermore, a new binary flag *cat _ coding _ mode* is added in the HEVC standard syntax to notify the mode selection between $\boldsymbol{C}_{dct/dst}$ and $\boldsymbol{C}_{cat}$. The associated semantic is given as follows: *cat _ coding _ mode* equal to 0 means that $\boldsymbol{C}_{dct/dst}$ is used for $\boldsymbol{x}_i$, and otherwise $\boldsymbol{C}_{cat}$ is used.

**Algorithm 1** gives the modified syntax of our method. We use a function, `transform _ tree(···)`, to demonstrate how to parse *cat _ coding _ mode*. It is worth noting that the syntax decoding of our CAT is same as that of the HEVC standard except for parsing the additional flag *cat _ coding _ mode*, where the corresponding syntax change is highlighted in blue. For the detailed description of HEVC decoding, we refer the interested reader to [31].

Fig. 3 shows the framework of CAT. In our implementation, CAT supports four transform sizes, including 4×4, 8×8, 16×16, and 32×32. In inter-mode coding, we propose

**Algorithm 1:** Proposed CAT Syntax Modification

```
Data: transform split condition $T_{split}$, coded block
      flag condition of chroma $F_{cbf,cbcr}$, coded block
      flag condition of luma $F_{cbf,luma}$, transform
      block index $tbi$
Result: transform split flag $split\_transform\_flag$,
        coded block flags of chroma $cbf\_cb$, $cbf\_cr$,
        coded block flag of luma $cbf\_luma$, CAT
        coding flag $cat\_coding\_mode$
function transform _ tree (..., $tbi$)
  Updating $T_{split}$
  if $T_{split}$ then
    Parsing $split\_transform\_flag$
  end if
  Updating $F_{cbf,cbcr}$
  if $F_{cbf,cbcr}$ then
    Parsing $cbf\_cb$ and $cbf\_cr$
  end if
  if $split\_transform\_flag$ then
    transform _ tree (..., 0)
    transform _ tree (..., 1)
    transform _ tree (..., 2)
    transform _ tree (..., 3)
  else
    Updating $F_{cbf,luma}$
    if $F_{cbf,luma}$ then
      Parsing $cbf\_luma$
    end if
    if $cbf\_luma$ then
      Parsing $cat\_coding\_mode$
    end if
  end if
end function
```

to choose the best transform mode between DCT/DST and CAT in the R-D sense. A detailed performance evaluation of CAT is provided in Section IV-B.

## B. BLOCK ADAPTIVE QUANTIZATION (BAQ) OPTIMIZATION

In Equation (5), BAQ can choose a unique QP for a TB, and achieve a better coding performance. However, it significantly increases the computational complexity. The reason is that the HEVC reference software currently employs a brute-force search scheme which needs to perform forward transform, quantization, de-quantization, inverse transform, and entropy coding for each QP candidate to find the best result of (5) in the R-D sense. Consequently, we introduce an efficient approach to obtain a feasible solution for the BAQ optimization.

### 1) TEMPLATE-BASED RATE REGULATION

In minimizing $\boldsymbol{J}_i$, the computational complexity to obtain the rate cost cannot be ignored. We estimate $R(\tilde{\boldsymbol{y}}_{i,QP})$ by fixing the number of QTCs instead of computing its practical value for the $i$-th CB. Specifically, we select the number of non-zero QTCs as one of the factors to estimate the rate cost. Also, the position information of these non-zero QTCs is utilized based on the transform theory. The rate constraint is computed by a binarization process,

$$\boldsymbol{\xi}_{\tilde{\boldsymbol{y}}_{i,QP_b}}(l) = \begin{cases} 1, & \tilde{\boldsymbol{y}}_i(l) \neq 0 \\ 0, & \tilde{\boldsymbol{y}}_i(l) = 0 \end{cases}. \tag{8}$$

where $\boldsymbol{\xi}_{\tilde{\boldsymbol{y}}_{i,QP_b}}$ is a QTC template, and the sample position $l = 1, 2, ..., N^2$.

The proposed template-based rate regularization is designed for the purpose of determining the most important QTC positions. We select the important QTCs of $\tilde{\boldsymbol{y}}_{i,QP}$ by

$$\tilde{\boldsymbol{y}}^*_{i,QP} = \boldsymbol{\xi}_{\tilde{\boldsymbol{y}}_{i,QP_b}} \circ \tilde{\boldsymbol{y}}_{i,QP}, \tag{9}$$

where $\tilde{\boldsymbol{y}}^*_{i,QP}$ is the selected QTCs to be transmitted to the decoder, and $\circ$ is the Hadamard product.

Fig. 4 shows the performance of the proposed rate regulation method (9), where experiments are conducted on numerous standard videos with various spatial resolutions, frame rates, and different scenarios. BD-rate is used to measure the rate regulation accuracy. It can be observed that our method can accurately control output bit-rate under the same distortion. For a detailed analysis, it is referred to our previous work [22].

### 2) FAST BAQ MODEL

As discussed above, the proposed template-based rate regulation method can effectively control the rate cost. In this case, Equation (5) is reformulated as

$$\begin{aligned} &\underset{QP \in \boldsymbol{S}_{QP} \quad \tilde{\boldsymbol{y}}^*_{i,QP} = \boldsymbol{\xi}_{\tilde{\boldsymbol{y}}_{i,QP_b}} \circ \tilde{\boldsymbol{y}}_{i,QP}}{\arg\min} \quad \boldsymbol{J}_i(QP) \\ &= D\left(\tilde{\boldsymbol{y}}_{i,QP}\right) + \lambda \cdot R\left(\tilde{\boldsymbol{y}}_{i,QP}\right) \\ &= \left\| \boldsymbol{T}^{-1} \boldsymbol{Q}^{-1}_{QP}\left(\tilde{\boldsymbol{y}}^*_{i,QP}\right) - \boldsymbol{x}_i \right\|_2^2 + \lambda \cdot R\left(\tilde{\boldsymbol{y}}^*_{i,QP}\right) \\ &= \left\| \boldsymbol{T}^{-1} \boldsymbol{Q}^{-1}_{QP}\left(\tilde{\boldsymbol{y}}^*_{i,QP}\right) - \boldsymbol{x}_i \right\|_2^2 + \lambda \cdot R\left(\tilde{\boldsymbol{y}}_{i,QP_b}\right). \end{aligned} \tag{10}$$

where $\boldsymbol{S}_{QP} = \{QP | QP \in [QP_b - \Delta QP, QP_b + \Delta QP]\}$.

In Equation (10), we can obtain $D(\tilde{\boldsymbol{y}}_{i,QP})$ in the transform domain, where the inverse integer transform $\boldsymbol{C}^{-1}$ results in negligible error. In addition, the entropy coding of $R(\tilde{\boldsymbol{y}}_{i,QP})$ can be saved in minimizing $\boldsymbol{J}_i$. Thus, a fast algorithm is developed for BAQ optimization.

## C. PROPOSED JOTQ MODEL

To improve video coding efficiency, we formulate the problem of JOTQ under a given $QP_b$ as

$$\begin{aligned} &\underset{\boldsymbol{T} \in \{\boldsymbol{C}_{dct/dst}, \boldsymbol{C}_{cat}\}, QP \in \boldsymbol{S}_{QP}}{\arg\min} \quad \boldsymbol{J}_i(\boldsymbol{T}, QP) \\ &= \left\| \boldsymbol{T}^{-1} \boldsymbol{Q}^{-1}_{QP}\left(\boldsymbol{Q}_{QP} \boldsymbol{T}\left(\boldsymbol{x}_i\right)\right) - \boldsymbol{x}_i \right\|_2^2 + \lambda R\left(\boldsymbol{Q}_{QP} \boldsymbol{T}\left(\boldsymbol{x}_i\right)\right) \end{aligned} \tag{11}$$

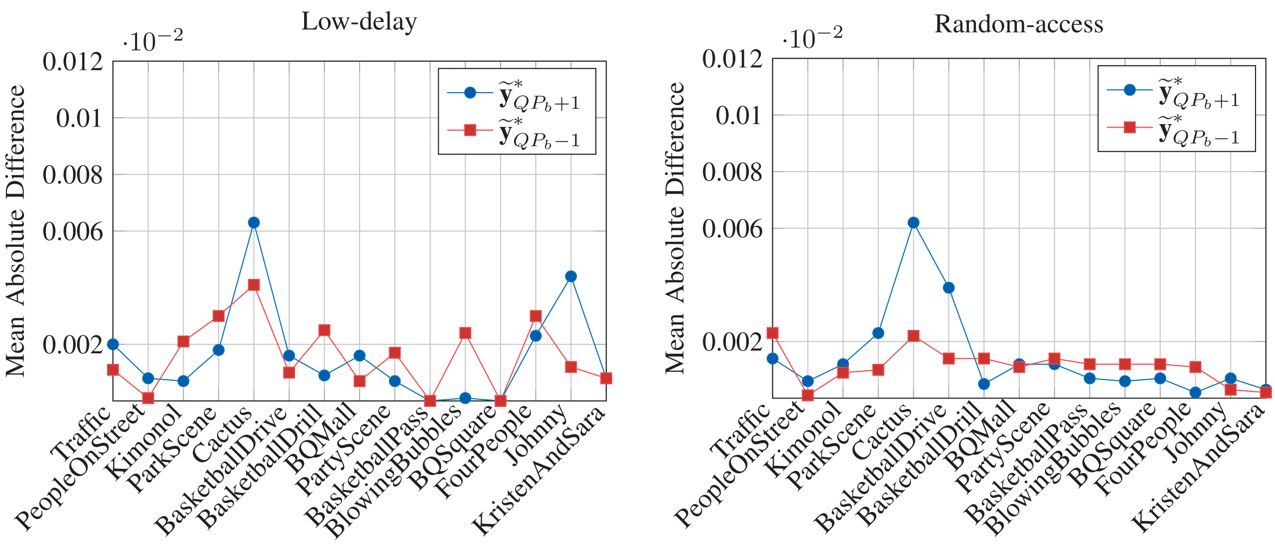

**FIGURE 4.** Mean absolute difference of the proposed template-based rate regulation method.

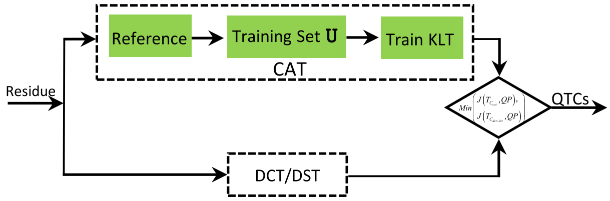

**FIGURE 5.** Flowchart of the proposed JQTQ method.

where $T^{-1}$ is the inverse of $T = C \otimes C$, and $Q^{-1}_{QP}$ is a de-quantization function (*i.e.*, the inverse process of Equation (4)).

In Equation (11), our target is to find the optimal transform and quantization combination. According to the optimization theory [32], it inevitably introduces a large amount of computational complexity to find the best $(\boldsymbol{T}, QP)$ due to the number of combinations.

In Fig. 5, we show the R-D process of a single T-Q combination $(\boldsymbol{T}, QP)$. Suppose that the input residue is $\boldsymbol{x}_i$. The encoding results of CAT are conducted by first collecting the reference frames (see Fig. 2) to obtain the associated training dataset (see Section III-A.1), then calculating the KLT transform matrices by (7), and finally obtaining the R-D cost $\boldsymbol{J}_i(\boldsymbol{C}_{cat}, QP)$, which is illustrated at the top row of Fig. 5. Similarly, the R-D results $\boldsymbol{J}_i(\boldsymbol{C}_{dct/dst}, QP)$ of DCT/DST are performed by (2) and (4) as shown in the bottom row of Fig. 5. The best R-D cost of encoding $\boldsymbol{x}_i$ is determined by minimizing $\boldsymbol{J}_i(\boldsymbol{C}_{cat}, QP)$ and $\boldsymbol{J}_i(\boldsymbol{C}_{dct/dst}, QP)$.

### 1) OVERALL JOTQ ALGORITHM

Using the joint T-Q model (11) [33] and the appropriate coding parameters, the encoder can achieve the R-D optimized quantization coefficients. The JOTQ method is implemented as follows.

Step 1) *Constructing the KLT training dataset:* For uni-prediction, the neighboring quarter-pixel samples of a prediction block are used to construct $\boldsymbol{u}_k$ as shown in Fig. 2. For bi-prediction, the elements of $\boldsymbol{u}_k$ are computed as the same way as that of a bi-prediction reference block.

Step 2) *Computing the CAT matrix and converting to integer transform*: A fast KLT decomposition is used to obtain $\boldsymbol{C}_{cat}$ in Equation (7). The elements of $\boldsymbol{C}_{cat}$ are multiplied by $N \times 2^{10}$ and rounded to the nearest integers.

Step 3) *Performing template-based rate regulation:* The optimal quantization coefficients are selected by Equation (9).

Step 4) *Determining the best QTCs:* The optimal QTCs for a single transform $\boldsymbol{T}$ can be computed by Equation (10). After obtaining the R-D costs of $\boldsymbol{C}_{dct/dst}$ and $\boldsymbol{C}_{cat}$, the best coding method is determined for a single T-Q combination as shown in Fig. 5.

Step 5) *Overall optimization:* If there is still a T-Q combination which is not checked, the encoder goes to *Step 3)*. The best T-Q result is computed after searching all of the T-Q combinations in Equation (11).

### 2) ACCELERATION STRATEGY

To reduce the computational complexity of JOTQ, we do not fully compute $\boldsymbol{C}_{cat}$ for all types of PB partitions during the R-D optimization process. Instead, we perform CAT only when the best PB is determined. In addition, when all QTCs are zeros, the computation of $\boldsymbol{C}_{cat}$ is skipped. In such a case, *cat_coding_mode* is not explicitly encoded, because the decoder can parse *cat_coding_mode* in the same way. When QTCs contain non-zero elements and the CAT method is selected, *cat_coding_mode* $= 1$ is signaled in bitstream, and otherwise *cat_coding_mode* $= 0$.

**TABLE 2.** General test conditions.

| | |
|---|---|
| Benchmark | HM 16.6 Main Profile |
| Coding Structure | Low-delay Main<br>Random-access Main |
| Entropy Coding | CABAC |
| Maximum Coding Unit | 64×64 |
| RDOQ | On |
| QP | 22, 27, 32, 37 |
| $\Delta QP$ | 2 |
| Other settings | Default |

The average R-D costs of the same temporal layer of size 4 × 4, 8 × 8, 16 × 16, and 32 × 32 in a GOP is computed as

$$RD_{N\times N}^{layer} = \frac{1}{I}\sum_{i} \boldsymbol{J}_i\,(\boldsymbol{T}, QP\,), \tag{12}$$

where *layer* denotes the hierarchical layer as shown in Fig. 1, *e.g.* $layer = 0, 1, 2, 3$. We use $\alpha \cdot RD_{N\times N}^{layer}$ as a threshold to reduce the computational complexity. When the R-D cost of the current TB is smaller than $\alpha \cdot RD_{N\times N}^{layer}$ (*e.g.* $\alpha$ is empirically set to 0.4), the computation of $\boldsymbol{C}_{cat}$ is skipped and $cat\_coding\_mode = 0$.

## IV. EXPERIMENTS AND DISCUSSIONS

In order to verify the effectiveness of JODQ, we implement it into the HEVC reference software HM 16.6 [8]. There are two different configurations to meet various video applications (*e.g.* VoD, OTT, broadcasting, video surveillance and conferencing, *etc.*), including low-delay (LD) main *encoder _lowdelay _main.cfg* and random-access (RA) main *encoder _ randomaccess _ main.cfg*.

Due to the simplicity of standard broadcast sequences (*i.e.*, *Class E*), we mainly focus on the experiments of more complex video sequences (*i.e.*, *Class A* ∼ *Class D*) according to common test condition. Moreover, JOTQ is compared with several advanced methods, including Saxena and Fernandes [13], Wang *et al.* [15], Lan *et al.* [16] and Lee *et al.* [21]. The detailed test conditions are tabulated in Table 2, where all the other encoding parameters in the experiments are set as the same for all methods.

We tabulate the experimental results in terms of Bjø ntegaard Delta rate (BD-rate) [17] metric, which has been widely used to measure the performance of video coding tools. This metric gives an average performance difference between the benchmark and the comparison methods. The BD-rate value for each sequence is computed by four `QPs`={22, 27, 32, 37}, where a negative value indicates bit-rate saving with the same video quality in terms of PSNR.

We measure the average complexity $T_{avg}$ by

$$T_{avg} = {}^{T_{mod}}\!/_{T_{ben}} \times 100\%, \tag{13}$$

where $T_{mod}$ is the practical running time of a modified method, and $T_{ben}$ is that of the benchmark method.

**TABLE 3.** Bjøntegaard delta rate (BD-rate) results of the proposed CAT method ($QP_b$ = 22, 27, 32, 37).

| | Sequence | LD main | RA main |
|---|---|---|---|
| Class A | Traffic | -1.64% | -1.17% |
| | PeopleOnStreet | -0.89% | -0.92% |
| Class B | Kimono | -0.26% | -0.55% |
| | ParkScene | -0.83% | -0.39% |
| | Cactus | -2.50% | -1.97% |
| | BasketballDrive | -1.65% | -1.78% |
| | BQTerrace | -6.72% | -7.21% |
| Class C | BasketballDrill | -5.90% | -3.04% |
| | BQMall | -3.11% | -1.85% |
| | PartyScene | -3.86% | -2.51% |
| | RaceHorses | -1.58% | -0.86% |
| Class D | BasketballPass | -1.18% | -0.69% |
| | BQSquare | -9.03% | -6.92% |
| | BlowingBubbles | -3.57% | -1.55% |
| | RaceHorses | -1.36% | -0.64% |
| **Average BD-rate saving (%)** | | **-2.94%** | **-2.14%** |

### A. BD-RATE PERFORMANCE OF CAT

Table 3 shows the overall performance of CAT. It can be seen that the average BD-rate improvements for the LD and RA configuration are 2.94% and 2.14%, respectively. Meanwhile, CAT reduces the bit-rate reduction up to 9.03% in terms of BD-rate.

From Table 3, we have two observations based on the characteristics of video content: (1) CAT obtains a higher bit-rate reduction for video sequence with rich edges (e.g. *BQTerrace*, *BasketballDrill* and *BQSquare*). The sharp edges may raise a great challenge for the fixed kernel transform $\boldsymbol{C}_{dct/dst}$, while $\boldsymbol{C}_{cat}$ can be trained to efficiently compact the energy of edge pattern. (2) CAT achieves a higher coding gain in the LD configuration than in the RA configuration. RA employs bi-prediction and complex hierarchical frame structure, which makes the residue has a lower correlation in a block. In such a case, the CAT training dataset has lower quality samples, which results in a lower coding gain.

### B. ABLATION STUDY

To fully demonstrate the performance of CAT, we conduct an ablation study to evaluate the influence under a different number of transform sizes, including (1) a single transform size 4 × 4 (marked as "4 × 4 only"), (2) two transform sizes 4×4 and 8×8 (marked as "4×4−8×8"), (3) three transform sizes 4 × 4, 8 × 8, and 16 × 16 (marked as "4 × 4 − 16 × 16"), and (4) four transform sizes ranging from 4 × 4 to 32 × 32 (marked as "4 × 4 − 32 × 32").

Table 4 gives the encoding results of our CAT method with various transform settings from "4 × 4 only" to "4 × 4 − 32 × 32". The detailed results for each video sequence of "4 × 4 − 32 × 32" are provided in Table 3. We have the following observations: (1) When the supported transform size

**TABLE 4.** **Bjøntegaard delta rate (BD-rate) results of different transform settings ($QP_b = 22, 27, 32, 37$).**

| Sequence | Transform types | LD main | RA main |
|---|---|---|---|
| | 4×4 only | -0.25% | -0.24% |
| Class A | 4×4-8×8 | -0.54% | -0.49% |
| | 4×4-16×16 | -0.88% | -0.75% |
| | 4×4-32×32 | -1.27% | -1.05% |
| | 4×4 only | -0.37% | -0.36% |
| Class B | 4×4-8×8 | -0.72% | -0.67% |
| | 4×4-16×16 | -1.65% | -1.59% |
| | 4×4-32×32 | -2.39% | -2.38% |
| | 4×4 only | -0.88% | -0.86% |
| Class C | 4×4-8×8 | -1.79% | -1.71% |
| | 4×4-16×16 | -2.83% | -1.98% |
| | 4×4-32×32 | -3.61% | -2.07% |
| | 4×4 only | -0.94% | -0.89% |
| Class D | 4×4-8×8 | -2.13% | -2.05% |
| | 4×4-16×16 | -3.24% | -2.32% |
| | 4×4-32×32 | -3.79% | -2.45% |

increases, the coding gain is greatly increased. For example, a single 4 × 4 transform size obtains the lowest coding gain among four settings. As 4 × 4 and 8 × 8 transform size are allowed, the performance is better than "4×4 only". The similar trends apply to "4×4−16×16" and "4×4−32×32". (2) When video resolution increases, the coding gain is reduced under the same transform configuration. For video sequence, larger resolution usually has higher prediction efficiency [34], leaving less redundancy for exploration.

### C. OVERALL BD-RATE PERFORMANCE COMPARISONS

Table 5 tabulates the overall performance comparisons. The result of Saxena and Fernandes [13] is provided in Table 5, where the average BD-rate savings of the LD and RA configuration are about 0.71% and 0.67% in terms of BD-rate, respectively. Wang *et al.* [15] obtains 1.58% bits saving in LD and 1.39% in RA. Lan *et al.* [16] obtains a significant coding gain, where the average BD-rate reductions for the LD and RA configuration are about 3.01% and 2.20%, respectively. Meanwhile, Lee *et al.* [21] achieves 0.37% and 0.41% BD-rate savings in the LD and RA configuration, respectively. One can see that JOQT achieves the best coding efficiency in comparison with [13], [15], [16], and [21]. JOTQ is able to save BD-rate up to 10.14%, and obtain the average BD-rate reductions of 4.11% and 3.38% under the LD and RA configuration, respectively.

### D. R-D PERFORMANCE

Fig. 6 shows the average R-D results of *Class A* and *Class B*. The X-axis is the average bit-rate results which are produced at $\texttt{QP} = \{22, 27, 32, 37\}$, while the Y-axis is the corresponding average PSNR values. From these R-D curves, we can observe that the benchmark provides the lowest coding gain for *Class B*. In addition, JOTQ significantly improves the compression efficiency. Our experimental results show that similar results can also be observed for *Class C* and *Class D*.

### E. OVERALL COMPUTATIONAL COMPLEXITY

For JOTQ, since the optimization process is implemented based on the quadtree structure, additional encoder and decoder complexities are increased. We collect the average results of the practical running time of all five methods as tabulated in Table 6. Specifically, it is observed that Lan *et al.* [16] has the highest computational complexity. The reason is that Lan *et al.* [16] needs to search 100 similar patches to construct the training data $U$, and computes a data dependent transform on both the encoder and decoder side. Especially, it is the most time-consuming part in decoding.

**TABLE 5.** **BD-rate comparisons between the proposed method and state-of-the-art methods under the low-delay (LD) and random-access (RA) configuration.**

| | Sequence | fps | Low-delay | | | | | Random-access | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | Saxena [13] | Wang [15] | Lan [16] | Lee [21] | JOTQ | Saxena [13] | Wang [15] | Lan [16] | Lee [21] | JOTQ |
| Class A | PeopleOnStreet | 30 | -0.92% | -0.68% | -0.95% | -0.52% | -2.10% | -1.14% | -0.61% | -0.95% | -0.67% | -1.82% |
| 2500×1600 | Traffic | 30 | -1.05% | -1.14% | -1.67% | -0.64% | -2.71% | -0.93% | -0.94% | -1.20% | -0.53% | -2.96% |
| | BasketballDrive | 50 | -0.84% | -1.52% | -1.68% | -0.33% | -2.65% | -0.75% | -1.16% | -1.82% | -0.38% | -3.43% |
| Class B | BQTerrace | 60 | -0.28% | -2.59% | -6.77% | -0.25% | -7.93% | -0.26% | -2.35% | -7.35% | -0.26% | -8.58% |
| 1920×1080 | Cactus | 50 | -0.94% | -1.14% | -2.53% | -0.51% | -3.75% | -0.71% | -1.11% | -2.02% | -0.27% | -2.93% |
| | Kimono | 24 | -0.87% | -1.38% | -0.27% | -0.47% | -1.58% | -1.02% | -1.18% | -0.58% | -0.31% | -1.72% |
| | ParkScene | 24 | -0.89% | -1.01% | -0.85% | -0.36% | -2.09% | -0.94% | -0.90% | -0.46% | -0.58% | -1.67% |
| | BasketballDrill | 50 | -0.84% | -2.30% | -5.95% | -0.27% | -6.89% | -0.85% | -2.09% | -3.07% | -0.54% | -4.65% |
| Class C | BQMall | 60 | -0.67% | -1.46% | -3.14% | -0.19% | -3.95% | -0.47% | -1.27% | -1.92% | -0.41% | -2.61% |
| 832×480 | PartyScene | 50 | -0.55% | -1.52% | -4.01% | -0.34% | -5.14% | -0.25% | -1.44% | -2.64% | -0.37% | -3.29% |
| | RaceHorses | 30 | -0.83% | -1.64% | -1.63% | -0.42% | -2.78% | -0.58% | -1.34% | -0.91% | -0.34% | -1.70% |
| | BasketballPass | 50 | -0.66% | -1.47% | -1.29% | -0.29% | -2.16% | -0.66% | -1.05% | -0.76% | -0.39% | -2.23% |
| Class D | BlowingBubbles | 50 | -0.37% | -1.75% | -3.68% | -0.26% | -4.87% | -0.45% | -1.66% | -1.65% | -0.38% | -2.91% |
| 416×240 | BQSquare | 60 | -0.25% | -2.69% | -9.26% | -0.25% | -10.14% | -0.27% | -2.51% | -6.92% | -0.33% | -8.45% |
| | RaceHorses | 30 | -0.72% | -1.37% | -1.45% | -0.43% | -2.85% | -0.71% | -1.30% | -0.69% | -0.34% | -1.82% |
| **Average BD-rate saving (%)** | | | **-0.71%** | **-1.58%** | **-3.01%** | **-0.37%** | **-4.11%** | **-0.67%** | **-1.39%** | **-2.20%** | **-0.41%** | **-3.38%** |

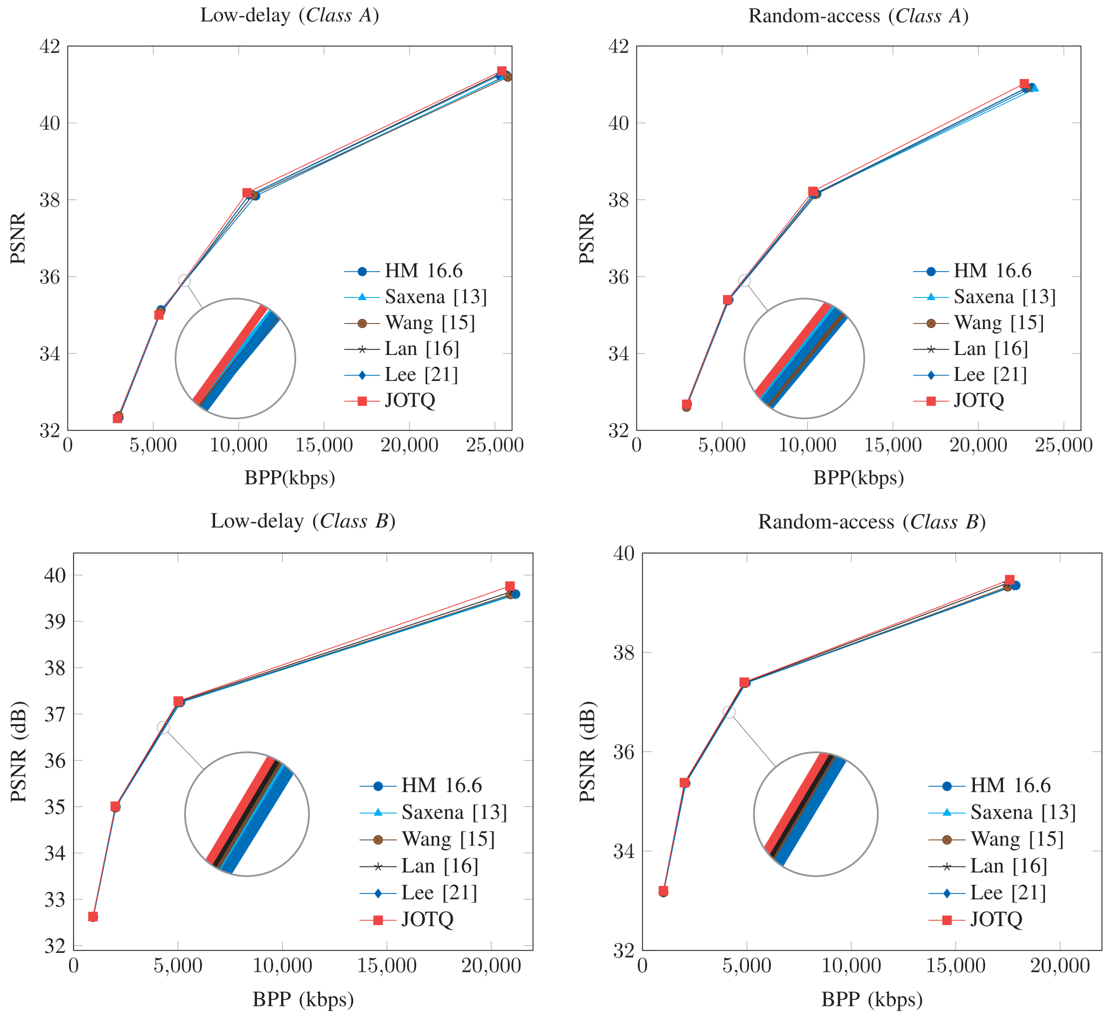

**FIGURE 6.** Illustrations of the R-D curves ($QP_b = 22, 27, 32, 37$). Top row: Average results of *Class A* under the LD main and RA main configuration. Bottom row: Average results of *Class B* under the LD main and RA main configuration.

**TABLE 6.** Average encoding and decoding complexity.

| | Saxena [13] | Wang [15] | Lan [16] | Lee [21] | JOTQ |
|---|---|---|---|---|---|
| Enc. $T_{avg}$ | 103.94% | 182.06% | 473.33% | 100.31% | 358.09% |
| Dec. $T_{avg}$ | 101.81% | 141.39% | 1107.40% | 99.75% | 391.14% |

Meanwhile, JOTQ consumes an average 358.09% running time on the encoder side, and 391.14% running time on the decoder side. On the one hand, JOTQ needs to compute block-based transform $\boldsymbol{C}_{cat}$ and QP in encoding, which contribute to the most time consumption functional parts. On the other hand, JOTQ saves a large amount of computation to search similar patches $\boldsymbol{x}_k^r$ compared to Lan *et al.* [16]. Meanwhile, the increased complexity is only consumed by $\boldsymbol{C}_{cat}$ on the decoder side, while BAQ is directly parsed in bitstream. Thus, JOTQ outperforms Lan [16] in terms of computational complexity $T_{avg}$. It should be admitted that although some fast algorithms have reduced the computation of JOQT compared to Lan [16], the complexity reduction still needs more effort.

## V. CONCLUSION

In this work, we attempt to jointly optimize transform and quantization for HEVC. To achieve this objective, we first compute CAT matrices from the neighboring reconstruction blocks by a fast KLT decomposition. We then introduce an efficient BAQ approach to obtain the best QP by a template-based rate regularization. Finally, we model JOTQ as a rate constrained optimization problem, and exploit fast algorithms for a feasible solution. Experiments show that the proposed method improves the coding efficiency by 3.75% on average in terms of BD-rate.

## REFERENCES

[1] D. Liu, P. An, R. Ma, W. Zhan, and L. Ai, "Scalable omnidirectional video coding for real-time virtual reality applications," *IEEE Access*, vol. 6, pp. 56323–56332, 2018.

[2] J. Xiong, X. Long, R. Shi, M. Wang, J. Yang, and G. Gui, "Background error propagation model based RDO in HEVC for surveillance and conference video coding," *IEEE Access*, vol. 6, pp. 67206–67216, 2018.

[3] X. Liang, Z. Li, Y. Yang, Z. Zhang, and Y. Zhang, "Detection of double compression for HEVC videos with fake bitrate," *IEEE Access*, vol. 6, pp. 53243–53253, 2018.

[4] S. Zhu, S. Zhang, and C. Ran, "An improved inter-frame prediction algorithm for video coding based on fractal and H.264," *IEEE Access*, vol. 5, pp. 18715–18724, 2017.

[5] M. Asif, M. B. Ahmad, I. A. Taj, and M. Tahir, "A generalized multi-layer framework for video coding to select prediction parameters," *IEEE Access*, vol. 6, pp. 25277–25291, 2018.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[7] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[8] *HM16.6 Reference Software*. Accessed: 2019. [Online]. Available: https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags

[9] B. Zeng and J. Fu, "Directional discrete cosine transforms—A new framework for image coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 18, no. 3, pp. 305–313, Mar. 2008.

[10] C. L. Chang, M. Makar, S. S. Tsai, and B. Girod, "Direction-adaptive partitioned block transform for color image coding," *IEEE Trans. Image Process.*, vol. 19, no. 7, pp. 1740–1755, Jul. 2010.

[11] Y. Yuan *et al.*, *CE2: Non-Square Quadtree Transform for Symmetric and Asymmetric Motion Partition*, document JCTVC-F412, 6th Meeting, Torino, Italy, 2011, pp. 14–22.

[12] J. Dong and K. N. Ngan, "Two-layer directional transform for high performance video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 4, pp. 619–625, Apr. 2012.

[13] A. Saxena and F. Fernandes, "Low latency secondary transforms for intra/inter prediction residual," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 4061–4071, Oct. 2013.

[14] M. Biswas, M. R. Pickering, and M. R. Frater, "Improved H.264-based video coding using an adaptive transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2010, pp. 165–168.

[15] M. Wang, K. N. Ngan, and L. Xu, "Efficient H.264/AVC video coding with adaptive transforms," *IEEE Trans. Multimedia*, vol. 16, no. 4, pp. 933–946, Jun. 2014.

[16] C. Lan, J. Xu, W. Zeng, G. Shi, and F. Wu, "Variable block-sized signal-dependent transform for video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 28, no. 8, pp. 1920–1933, Aug. 2017.

[17] G. Bjontegard, *Improvements of the BD-PSNR Model*, document ITU-T VCEG-AI11, 2008.

[18] G. J. Sullivan, *Adaptive Quantization Encoding Technique Using an Equal Expected-Value Rule*, document, Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG, JVT-N011, 2005.

[19] M. Karczewicz, Y. Ye, and I. Chong, *Rate Distortion Optimized Quantization*, document, ITU-TQ, 2008, vol. 6.

[20] X. Yu, D.-K. He, and E.-H. Yang, *Improved quantization for HEVC*, document JCTVC-B035, Joint Collaborative Team on Video Coding, 2010.

[21] P. Lee, S. Victor, and V. Rahul, "Adaptive quantization by soft thresholding in HEVC," in *Proc. Picture Coding Symp. (PCS)*, 2015, pp. 35–39.

[22] M. Wang, K. N. Ngan, H. Li, and H. Zeng, "Improved block level adaptive quantization for high efficiency video coding," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jul. 2015, pp. 509–512.s

[23] M. Ropert, J. Le Tanou, M. Bichon, and M. Blestel, "RD spatio-temporal adaptive quantization based on temporal distortion backpropagation in HEVC," in *Proc. IEEE 19th Int. Workshop Multimedia Signal Process. (MMSP)*, Oct. 2017, pp. 1–6.

[24] G. Xiang *et al.*, "A perceptually temporal adaptive quantization algorithm for HEVC," *J. Vis. Commun. Image Represent.*, vol. 50, pp. 280–289, Jan. 2018.

[25] M. Khosravy, N. Gupta, N. Marina, I. K. Sethi, and M. R. Asharif, "Perceptual adaptation of image based on Chevreul–Mach bands visual phenomenon," *IEEE Signal Process. Lett.*, vol. 24, no. 5, pp. 594–598, May 2017.

[26] V. Sze, M. Budagavi, and G. J. Sullivan, "Quantization matrix," in *High Efficiency Video Coding (HEVC): Algorithms and Architecture*. Springer, 2014, pp. 158–159.

[27] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, and M. Sadafale, "Core transform design in the high efficiency video coding (HEVC) standard," *IEEE J. Sel. Topics Signal Process.*, vol. 7, no. 6, pp. 1029–1041, Dec. 2013.

[28] A. Saxena and F. C. Fernandes, "DCT/DST-based transform coding for intra prediction in image/video coding," *IEEE Trans. Image Process.*, vol. 22, no. 10, pp. 3974–3981, Oct. 2013.

[29] T. D. Chuang, C. Y. Chen, Y. L. Chang, Y. W. Huang, and S. Lei, *AHG Quantization: Sub-LCU Delta QP*, document JCTVC-E051, Joint Collaborative Team on Video Coding, 2011.

[30] B. Li, J. Xu, D. Zhang, and H. Li, "QP refinement according to Lagrange multiplier for high efficiency video coding," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2013, pp. 477–480.

[31] *Information Technology—High Efficiency Coding and Media Delivery in Heterogeneous Environments—Part 2: High Efficiency Video Coding*, document ISO/IEC 23008-2:2013, 2013, vol. 1, p. 20.

[32] S. P. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[33] M. Wang, Q. Bi, and Y. Zhu, "Video compression: A jointly optimized transform-quantization method," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput.*, Oct. 2017, pp. 1–5.

[34] H. Takeda, P. Milanfar, M. Protter, and M. Elad, "Super-resolution without explicit subpixel motion estimation," *IEEE Trans. Image Process.*, vol. 18, no. 9, pp. 1958–1975, Sep. 2009.

**MIAOHUI WANG** (S'13–M'16) received the Ph.D. degree from The Chinese University of Hong Kong (CUHK), Hong Kong, in 2015.

From 2014 to 2015, he was involved in the standardization of video coding with the Innovation Laboratory, InterDigital, Inc., San Diego, CA, USA. From 2015 to 2017, he was a Senior Research Engineer in computer vision and machine learning with The Creative Life (TCL) Research Institute of Hong Kong, Hong Kong. He joined Shenzhen University (SZU), Shenzhen, China, as an Assistant Professor, in 2017, where he is currently with the College of Information Engineering. He has authored or coauthored over 30 refereed technical papers in international journals. His current research interests include a wide range of topics related with image/video compression, transmission and analysis, computer vision, and machine learning. He was a recipient of the Best Thesis Award from Shanghai (Ministry of Education of Shanghai City) and Fudan University (FDU), in 2012, China. He has received the Best Paper Award from the International Conference on Advanced Hybrid Information Processing, in 2018. He has received the second runner-up place of Grand Challenge on Learning-Based Image Inpainting from the International Conference on Multimedia & Expo, in 2019. He is a member of the IEEE Circuits and Systems Society.

**WUYUAN XIE** (S'15–M'19) received the B.S. degree from Central South University (CSU), in 2006, the M.S. degree from the South China University of Technology University (SCUT), in 2009, and the Ph.D. degree from the Department of Mechanical and Automation Engineering, The Chinese University of Hong Kong (CUHK), in 2016.

From 2016 to 2017, she held a postdoctoral position at The Hong Kong Polytechnic University. From 2008 to 2011, she was with the Shenzhen Institutes of Advanced Technology, Chinese Academy of Science (CAS). She is currently an Assistant Professor with Shenzhen University. Her research interests include 3D computer vision and machine learning.

**JIAN XIONG** (M'18) received the Ph.D. degree in single and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2015.

He was a Research Assistant with the Image and Video Processing Laboratory, The Chinese University of Hong Kong, Hong Kong, in 2014. Since 2015, he has been an Assistant Professor with the College of Telecommunications and Information Engineering, Nanjing University of Posts and Telecommunications, Nanjing, China. He has authored or coauthored over 30 refereed technical papers in international journals. His current research interests include image and video coding, computer vision, and machine learning. He is a member of the IEEE Circuits and Systems Society.

**DAYONG WANG** (M'19) received the Ph.D. degree in computer science from the University of Electronic Science and Technology of China (UESTC), in 2010. He was a Lecturer with the Hubei University of Arts and Science, from 2010 to 2012. He held a postdoctoral research position at the Graduate School, Tsinghua University, Shenzhen, from 2012 to 2015. He is currently an Associate Professor with the Chongqing University of Posts and Telecommunications, and also a Postdoctoral Fellow of the University of Electronic Science and Technology of China. His main research interest includes video coding.

**JING QIN** received the Ph.D. degree in computer science and engineering from The Chinese University of Hong Kong, in 2009.

He is currently an Assistant Professor with the Center for Smart Health, School of Nursing, The Hong Kong Polytechnic University. His research interests include medical image processing, virtual/augmented reality for healthcare and medicine training, deep learning, visualization and human–computer interaction, and health informatics. He and his collaborators were nominated for the outstanding paper award by the International Simulation and Gaming Association 40th Annual Conference, in 2009. He has received the Champion at the Third Hong Kong Innovation Day and Innovation Awards Competition, the Global Scholarship Program for Research Excellence (CNOOC Grants) from CUHK, in 2008, the Hong Kong Medical and Health Device Industries Association Student Research Award, in 2009, the Best Paper Award in Medical Image Computing from the International Conference on Medical Imaging and Augmented Reality 2016, and the Medical Image Analysis-MICCAI 2017 Best Paper Award.