Fast Transform Kernel Selection Based on Frequency Matching and Probability Model for AV1

Zhijian Hao, Heming Sun, *Member, IEEE*, Guohao Xu, Jiaming Liu, Xiankui Xiong, Xuanpeng Zhu, Xiaoyang Zeng, *Member, IEEE*, and Yibo Fan

***Abstract*—As a fundamental component of video coding, transform coding concentrates the energy scattered in the spatial domain onto the upper-left region of the frequency domain. This concentration contributes significantly to Rate-Distortion performance improvement when combined with quantization and entropy coding. To better adapt the dynamic characteristics of image content, Alliance for Open Media Video 1 (AV1) introduces multiple transform kernels, which brings substantial coding performance benefits, albeit at the cost of considerably computational complexity. In this paper, we propose a fast transform kernel selection algorithm for AV1 based on frequency matching and probability model to effectively accelerate the coding process with an acceptable level of performance loss. Firstly, the concept of Frequency Matching Factor (FMF) based on cosine similarity is defined for the first time to describe the similarity between the residual block and the primary frequency basis image of the transform kernel. Statistical results demonstrate a clear distribution relationship between FMFs and normalized Rate-Distortion optimization costs (nRDOC). Then, leveraging these distribution characteristics, we establish Gaussian normal probability model of nRDOC for each FMF by characterizing the parameters of the normal model as functions of FMFs, enhancing the normal model's accuracy and coding performance. Finally, based on the derived normal models, we design a fast selection algorithm with scalability and hardware-friendliness to skip the non-promising transform kernels. Experimental results show that the performance loss of the proposed fast algorithm is 1.15% when 57.66% of the transform kernels are skipped, resulting in a saving of 20.09% encoding time, which is superior to other fast algorithms found in the literature and competitive with the pruning algorithm based on the neural network in the AV1 reference software.**

***Index Terms*—AV1, transform kernel selection, matching factor, probability model.**

Manuscript received 15 October 2023; revised 4 February 2024; accepted 26 February 2024. Date of publication 26 March 2024; date of current version 7 June 2024. This work was supported in part by the National Key Research and Development Program of China under Grant 2023YFB4502802; in part by the National Natural Science Foundation of China under Grant 62031009; in part by the "Ling Yan" Program for Tackling Key Problems in Zhejiang Province under Grant 2022C01098; in part by the Alibaba Innovative Research (AIR) Program; in part by the Alibaba Research Fellow (ARF) Program; in part by the Fudan-ZTE Joint Lab; in part by the CCF-Alibaba Innovative Research Fund For Young Scholars; and in part by JSPS KAKENHI under Grant JP21K17770 and Grant JP23K16861. *(Corresponding author: Heming Sun.)*

Zhijian Hao, Guohao Xu, Jiaming Liu, Xiaoyang Zeng, and Yibo Fan are with the State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai 200433, China (e-mail: zjhao19@fudan.edu.cn; ghxu20@fudan.edu.cn; liujm22@m.fudan.edu.cn; xyzeng@fudan.edu.cn; fanyibo@fudan.edu.cn).

Heming Sun is with the Faculty of Engineering, Yokohama National University, Yokohama 240-0067, Japan (e-mail: sun-heming-vg@ynu.ac.jp).

Xiankui Xiong and Xuanpeng Zhu are with the State Key Laboratory of Mobile Network and Mobile Multimedia Technology, ZTE Corporation, Shenzhen 518057, China (e-mail: xiong.xiankui@zte.com.cn; Zhu.xuanpeng@zte.com.cn).

Digital Object Identifier 10.1109/TBC.2024.3374078

I. Introduction

WITHIN the hybrid coding framework, transform coding has consistently held a pivotal role as it concentrates the energy scattered in the spatial domain onto the upper-left region of the frequency domain. Cooperating with quantization and entropy coding, this concentration yields a considerable benefit in term of Rate-Distortion (RD) performance. According to the proof in [1] and [2], under the first-order Markov hypothesis, the Discrete Cosine Transform (DCT) stands out as the optimal approximation of the K-L transform, which achieves superior decorrelation with lower computational complexity. Consequently, in early coding standards like H.264 and H.265, the DCT was employed as the primary transform kernel.

More diverse residual blocks emerge with the continuous expansion of video services and the growth of video content at higher resolutions, alongside the refinement of intra and inter prediction tools. This diversification in residual blocks reflects that the image content no longer strictly adheres to the first-order Markov distribution [3]. As revealed in [4], it is further proved that when image pixels obey the first-order Gaussian-Markov hypothesis, the energy concentration effect of Discrete Sine Transform (DST) surpasses that of DCT so that the DST exhibits superior RD performance in this scenario.

Based on the above reasons, in the latest generation of video coding standards, such as Alliance for Open Media Video 1 (AV1), Versatile Video Coding (VVC), and Audio Video Coding Standard 3 (AVS3) [5], [6], [7], multiple transform types have been introduced to better accommodate the intricacies of current image content [1], [8], [9], [10]. Specifically, as a typical AV1 encoder implementation, the AV1 reference library libaom [11] introduces a comprehensive set of transform types, including four basic 1D transform types: DCT, Asymmetric Discrete Sine Transform (ADST), Flipped Asymmetric Discrete Sine Transform (Flip_ADST) and Identity Transform (IDT) [12]. These four 1D transform

types are combined in pairs in the horizontal and vertical directions to form 16 basic 2D transform kernels. The introduction of these 16 kinds of 2D transform kernels has brought a considerable Bjøntegaard Delta-Bit Rate (BD-BR) [13] performance improvement. It is imperative to acknowledge that this substantial performance boost comes at the cost of considerable computational complexity, as introducing these kernels significantly increases the complexity of Rate-Distortion Optimization (RDO). In the conventional coding process, the encoder employs the RDO strategy to select the optimal mode from a range of candidate modes, ultimately adopting it as the final coding mode. Consequently, in contrast to its predecessor, AV1 experiences a sixteenfold escalation in traversal complexity within the transform coding due to the expansion of the number of transform kernels from one to sixteen. According to the complexity profiling of the baseline AV1 tools in [14] and [15], the second most time-consuming step is the Transform Search, which is responsible for performing a transform search over a list of mode candidates.

Even though some efficient hardware implementations for multiple transform kernels has been proposed [16], [17], [18], [19], [20], [21], determining the optimal kernel among candidates is still of considerable complexity. To mitigate the computational complexity, the latest AV1 reference software incorporates specific fast algorithms. For intra coding, an approach of fast RDO decision made in the frequency domain is adopted. This method has two remarkable characteristics. On the one hand, to reduce the traversal complexity, the number of candidate transform kernels is reduced according to the residual block size, which leads to an inevitable performance loss. On the other hand, the calculation of RDO cost (RDOC) is performed in frequency domain, avoiding the calculation of inverse quantization and inverse transform. For inter coding, AV1 employs neural networks to prune the number of transform kernels, presented initially in [22]. The neural network architecture in AV1 has been meticulously fine-tuned for a balance between complexity and accuracy, with one hidden layer structure demonstrating superior performance retention. Nonetheless, the computational overhead of the neural network can not be ignored. Furthermore, regarding hardware implementation, the neural network-based methods are not hardware-friendly, and the fixed-point representation poses challenges in fully maintaining the effectiveness of such methods in hardware environments.

While numerous fast algorithms have been developed for other modes [23], [24], [25], [26], [27], [28], [29], such as intra prediction modes, inter prediction modes, and quantization modes, few of them aim to address the fast selection of transform kernels in AV1. Reference [30] has made efforts to address this problem and proposes a cost prediction method based on sparse Laplacian matrices, which is used to estimate the bit-rate of residual block after transformation. Based on this bit-rate, the number of transform kernels is pruned. However, there are two notable limitations to this method. Firstly, the authors do not provide the proper weights of coefficients, which significantly impacts its performance. Secondly, the higher the accuracy, the more Laplacian matrices must be weighted and combined. This weighted combination substantially reduces the sparsity of the derived Laplacian matrix, thereby diminishing the advantage of low computational complexity inherent to this method. Other approaches have also been explored in different standards, like HEVC and VVC [31], [32], [33], [34], [35], [36]. For example, in [33], an algorithm is introduced for VVC that leverages pixel values at the four corners to depict the distribution of residual blocks and determines the appropriate transform kernel set. While this method utilizes only four pixels and entails low computational overhead, it is difficult to accurately characterize the distribution of pixels, leading to apparent performance loss. Besides, as the number of transform kernels allowed in other standards is much smaller than that in AV1, those methods used in non-AV1 standards would be ineffective when applied to AV1.

To overcome the shortcomings of the above methods, this paper presents a fast transform kernel selection method based on frequency matching and probability model for AV1. This method has three main contributions:

- This paper investigates the mechanism of energy concentration of transform coding and introduces the concept of Frequency Matching Factor (FMF) based on cosine similarity for the first time. The FMF reflects the similarity between the residual block and the primary frequency basis image of the transform kernel. The statistical findings show a conspicuous distribution relationship between FMF and normalized RDOC (nRDOC). That is, the higher the FMF, the smaller the nRDOC tends to be. Such a relationship can serve as a basis for designing efficient fast algorithms.
- The normal probability distribution models are established for nRDOC, which offer a more precise depiction of the nRDOC distribution for each FMF. In addition, we parameterize the normal model as the functions of FMF, which significantly improves the accuracy and acquisition speed of the probability models.
- We propose a fast algorithm for the transform kernel selection process based on the derived normal models. When the normal model of nRDOC corresponding to the candidate transform kernel is available, we can calculate the probability that the nRDOC of the candidate transform kernel exceeds the current optimal nRDOC to decide whether to skip this kernel. In addition, this algorithm is hardware-friendly and scalable. By adjusting the skip threshold, it can provide diverse trade-offs between speed and accuracy to suit different requirements.

The rest of this paper is organized as follows. Section II describes the essential background of AV1 transforms. Section III introduces the proposed fast transform kernel selection algorithm in detail. Section IV presents the experimental results, which demonstrate the advantages of the proposed approach over existing works in the literature. Section V concludes this paper.

II. Background

A. Transforms in AV1

In video coding, DCT can concentrate the energy of flat residual blocks onto the upper left corner of coefficient blocks. The subsequent quantization [37] can reduce or remove the

TABLE I
BASIS FUNCTIONS OF DCT, ADST, FLIP_ADST & IDT

Transform Types	Basis Function, $i, j = 0, 1, ..., N-1$
DCT	$T_i(j) = w \cdot \sqrt{\frac{2}{N}} \cdot cos(\frac{\pi \cdot i \cdot (2j+1)}{2N})$, where $w = \begin{cases} \sqrt{\frac{2}{N}}, i = 0 \\ 1 \quad , i \neq 0 \end{cases}$
ADST	$T_i(j) = \sqrt{\frac{2}{N}} \cdot sin(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{2N})$
Flip_ADST	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot cos(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1})$
IDT	$T_i(j) = \begin{cases} 1 \quad , j = i \\ 0 \quad , j \neq i \end{cases}$

high-frequency components insensitive to human eyes. In addition, entropy coding encodes the coefficients with zigzag scanning [38], [39], where the shorter the length of the non-zero coefficients, the lower the code rate. Therefore, transform combined with quantization and entropy coding can effectively reduce the bit stream length and improve compression efficiency.

As previously mentioned, relying solely on the DCT to concentrate the residual block's energy proves inadequate. For instance, for the residual blocks whose values increase gradually, the energy concentration effect achieved by ADST surpasses that of DCT. Therefore, the transform scheme with multiple transform kernels is necessary for the latest video coding standards. Table I shows the basis functions of the four basic 1D transform types employed in AV1: DCT, ADST, Flip_ADST, and IDT, respectively [40]. In the following, we overview some fundamental transform concepts, taking the 4-point 1D and 4×4 2D DCT as examples. The 1D transform formula of DCT is as follows:

$$Y(k) = \sqrt{\frac{2}{N}} \varepsilon_k \sum_{n=0}^{N-1} x(n) cos\left[\frac{k(2n+1)\pi}{2N}\right],$$
$$\varepsilon_k = \begin{cases} \sqrt{\frac{2}{N}}, & k = 0 \\ 1, & k \neq 0 \end{cases} \tag{1}$$
$$r(n,k) = \sqrt{\frac{2}{N}} \varepsilon_k cos\left[\frac{k(2n+1)\pi}{2N}\right], \tag{2}$$

where k is the index of transform coefficients and $k = 0, 1, \ldots, N-1$. $Y(k)$ is the transformed coefficients, $x(n)$ is the input data, and N is the transform size. $r(n,k)$ is defined as the basis function of 1D DCT. By traversing k from 0 to 3 in (2), we can obtain the four basis vectors of the 1D DCT, as illustrated in Fig. 1. These 1D vectors form a transform matrix so that (1) can be expressed in matrix form as follows:

$$Y = A_N \times X^T, \tag{3}$$

where A_N is the transform matrix. In practical video coding, the integer transform is usually adopted considering the calculation complexity. By scaling and rounding the A_N matrix while preserving its orthogonality as much as possible [41], the integer transform matrix T_N can be obtained. Therefore, the integer transform can be conducted as (4):

$$Y = T_N \times X^T. \tag{4}$$

AV1 applied the 2D transform to the residual blocks. 2D DCT is formulated as (5).

$$F(k,l) = \sum_{m=0}^{N-1} \sum_{n=0}^{N-1} x(m,n) r(m,n,k,l), \tag{5}$$
$$r(m,n,k,l) = \epsilon_k \epsilon_l cos\left[\frac{k(2m+1)\pi}{2N}\right] cos\left[\frac{l(2n+1)\pi}{2N}\right],$$
$$\epsilon_k = \epsilon_l = \begin{cases} \sqrt{\frac{1}{N}}, & k, l = 0, \\ \sqrt{\frac{2}{N}}, & \text{else} \end{cases} \tag{6}$$

where $m, n, k, l = 0, 1, \ldots, N-1$. $r(m,n,k,l)$ represents the forward transform basis function. It is worth noting that the obtained basis functions are orthogonal under different values of k and l. A common method to reduce the computational complexity is to decompose the 2D transform into two 1D transforms according to the separability of the 2D transform. Therefore, the 2D transform is conducted by two successive 1D transforms and can be summarized as (7):

$$\begin{aligned} Y &= T_N \times Y_{tmp}^T \\ &= T_N \times \left(T_N \times X^T\right)^T \\ &= T_N \times X \times T_N^T, \end{aligned} \tag{7}$$

in this case, Y_{tmp} is the intermediate matrix that holds the result of the first 1D transform. Y is the transformed coefficient block. When reconstructing the residual block, the inverse transform is performed according to (8):

$$x(m,n) = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) s(m,n,k,l), \tag{8}$$
$$\begin{aligned} s(m,n,k,l) &= r(m,n,k,l) \\ &= \epsilon_k \epsilon_l cos\left[\frac{k(2m+1)\pi}{2N}\right] cos\left[\frac{l(2n+1)\pi}{2N}\right], \end{aligned} \tag{9}$$

where $s(m,n,k,l)$ is the inverse transform kernel, which depends on the value of m, n, k, l, rather than $x(m,n)$ or $F(k,l)$. Equation (8) reflected that the essence of transform is to decompose the spatial block into a set of weighted combinations of images in the frequency domain, and the transformed coefficient is the weight. It could be more clear by modifying the symbols used as (10):

$$X = \sum_{k=0}^{N-1} \sum_{l=0}^{N-1} F(k,l) S_{kl} \tag{10}$$
$$S_{kl} = \begin{Bmatrix} s(0,0,k,l) & s(0,1,k,l) & \ldots & s(0,\widetilde{N},k,l) \\ s(1,0,k,l) & s(1,1,k,l) & \ldots & s(1,\widetilde{N},k,l) \\ \vdots & \vdots & \vdots & \vdots \\ s(\widetilde{N},0,k,l) & s(\widetilde{N},1,k,l) & \ldots & s(\widetilde{N},\widetilde{N},k,l) \end{Bmatrix},$$
$$\widetilde{N} = N - 1, \tag{11}$$

the S_{kl} is defined as the basis image corresponding to the frequency index (k,l). Traversing k and l, we can get 16

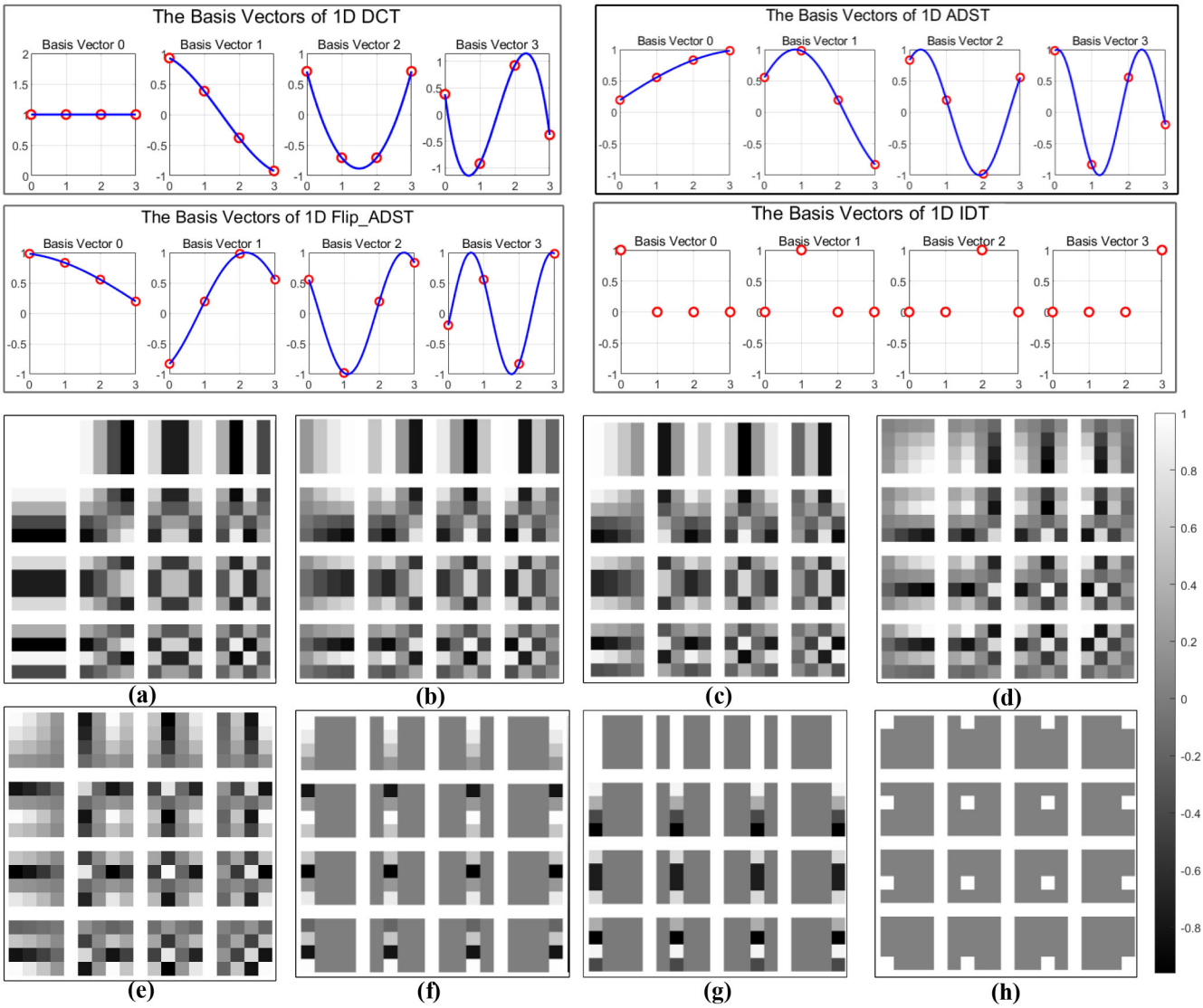


Fig. 1. The basis vectors of 1D DCT, ADST, Flip_ADST, and IDT and the basis images of 2D transform kernels, where (a) - (h) corresponds to C_C, A_C, F_C, A_A, F_F, I_F, I_C, and I_I. C, A, F, and I are short for DCT, ADST, Flip_ADST, and IDT. A_C is short for ADST_DCT, where the horizontal transform is ADST and the vertical transform is DCT.

4×4 basis images, and the basis image at position $(0, 0)$ is called the primary frequency basis image in this paper. When expanded into one-dimensional vectors, these 16 basis images are orthogonal to each other [42], as they are derived from the basis functions in the (9). Fig. 1a $\sim$ Fig. 1h visually presents the basis images of different transform kernels at the size of 4×4.

III. The Proposed Fast Algorithm

This section introduces the fast transform kernel selection algorithm proposed in this paper in detail. First, we explore the mechanism of energy concentration in the transform coding process. Based on this mechanism, we further put forward the fundamental concept of this paper, that is, the Frequency Matching Factor based on cosine similarity. Then, we describe the process of establishing nRDOC normal probability models for all FMFs. Finally, Section III-E presents the proposed transform kernel selection algorithm utilizing these normal models.

A. Energy Concentration Analysis for Transform Coding

As previously discussed, the effectiveness of transform coding lies in its ability to concentrate the energy scattered in the spatial domain into the low-frequency region of the frequency domain. The more energy is concentrated, the better compression effect can be achieved. In this section, we explore the precise sufficient and necessary condition under which the spatial energy can be perfectly concentrated in the frequency domain, specifically at the coefficient position where k, l are equal to 0.

Our investigation leads us to a straightforward conclusion: the necessary and sufficient condition for a perfect energy concentration is that the residual block is the scaling of the primary frequency basis image of the transform kernel. This condition can be formally expressed as follows:

$$X = s * S_{00}, \tag{12}$$

where X is the residual block, S_{00} is the primary frequency basis image, and s is a scaling factor, and it can be

any value. The following is the formal proof of this assertion.

First, the necessity of this condition is proved as follows.

Given that the transformation process is lossless, it follows that the total energy in the spatial domain equals the total energy in the frequency domain:

$$\sum_{m=0}^{N-1}\sum_{n=0}^{N-1} x^2(m,n) = \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F^2(k,l) = E, \tag{13}$$

where E is defined as the total energy of the residual block, which refers to the sum of squares of all terms in the spatial or frequency block. Assuming that the energy is concentrated in the upper left corner, which means:

$$F(0,0)^2 = E, \tag{14}$$

with (13) and (14), the following equation (15) can be derived:

$$\sum_{k\neq 0}^{N-1}\sum_{l\neq 0}^{N-1} F^2(k,l) = E - F^2(0,0) = 0 \tag{15}$$

Therefore, it can be concluded as (16),

$$\begin{cases} F(k,l) \neq 0,\ k = l = 0 \\ F(k,l) = 0,\ k, l \neq 0 \end{cases} \tag{16}$$

Substituting (16) into (10), we can get:

$$\begin{aligned} X &= \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F(k,l) S_{kl} \\ &= F(0,0) * S_{00}. \end{aligned} \tag{17}$$

Therefore, (12) is satisfied, and $s = F(0,0)$ in this case.

Then, we prove the sufficiency, that is, when the residual block satisfies (12), its energy will be perfectly concentrated onto the upper left corner of the coefficient block after the transform.

With (12) and (10), we can get:

$$\begin{aligned} X &= s * S_{00} \\ &= \sum_{k=0}^{N-1}\sum_{l=0}^{N-1} F(k,l) S_{kl} \end{aligned} \tag{18}$$

thus,

$$[s - F(0,0)] * S_{00} + \sum_{k\neq 0}^{N-1}\sum_{l\neq 0}^{N-1} F(k,l) S_{kl} = 0, \tag{19}$$

as mentioned in (11), the 16 S_{kl}s are orthogonal to each other. Therefore, (19) holds if and only if (20) are met.

$$\begin{cases} s - F(0,0) = 0, \\ F(k,l) = 0, k, l \neq 0. \end{cases} \tag{20}$$

Equation (20) elucidates that only at the position of $(0,0)$ are the coefficients nonzero, which means the residual block energy is concentrated into $F(0,0)$.

Based on the above analysis, we can conclude that when the residual block is the scaling of the primary frequency basis image of the current transform kernel, the energy of the residual block can be completely concentrated. This conclusion easily extends to the notion that the greater the similarity between the residual block and the scaling of the primary frequency basis image of the current transform kernel, the more pronounced the energy concentration effect becomes.

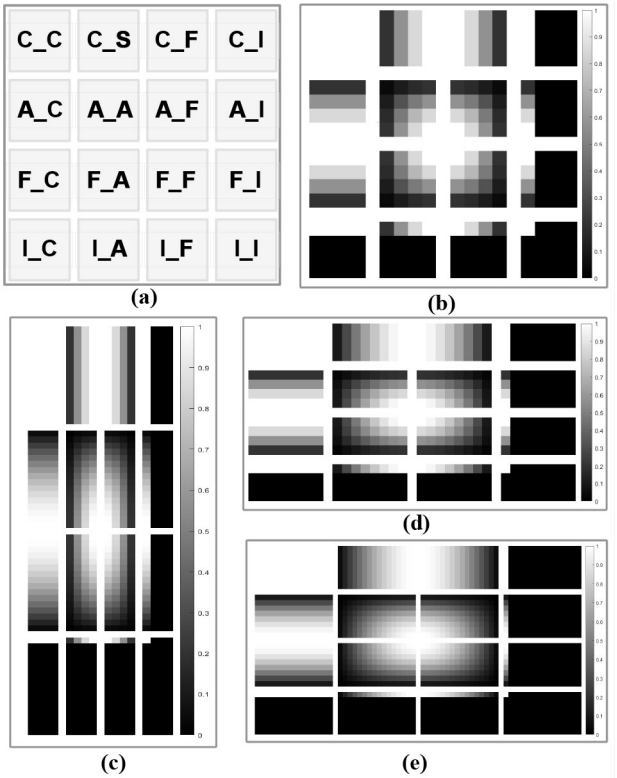


Fig. 2. The primary frequency basis image corresponds to 16 transform kernels. (a) indicates the position of each transform kernel. (b) presents the 16 primary frequency basis image at the transform size of 4×4. Similarly, (c)-(e) correspond to the sizes of 16×4, 4×8, and 8×16.

B. Frequency Matching Factor

In the last subsection, the energy concentration mechanism of transform coding has been explained in detail. It is evident that the similarity between residual block X and the scaling of the primary frequency basis image S_{00} of the current transform kernel significantly influences the compression performance. Fig. 2 illustrates the primary frequency images corresponding to different transform kernels under different sizes. The distribution of different primary frequency images varies, enabling them to accommodate the dynamic and diverse residual blocks effectively. As a known thing, the introduction of ADST is for residual blocks with the distribution that as pixels move farther away from the reference point, the magnitude of the residual increases. Such distribution aligns precisely with that of the primary frequency basis image of the ADST. Consequently, the ADST proves to be more suitable for transforming such residual blocks than the DCT.

Previous analysis reveals the importance of the similarity between the residual block and the primary frequency basis image. This prompts us to ponder whether we can define a factor that effectively characterizes this similarity, which satisfies at least the following two conditions:

a) The factor should be larger when the distribution of the residual block closely resembles that of the primary frequency basis image.

b) The factor should exhibit scale invariance. This means that as long as the two matrices present a scaling relationship, they are considered to be completely similar. For instance, when calculating the similarity corresponding to S_{00}, the result of matrix A and $2A$ should be identical.

To fulfill the two conditions outlined above, we introduce the concept of Frequency Matching Factor based on cosine similarity. Cosine similarity [43] quantifies the angle between two vectors, such as vectors A and B, and is defined as follows:

$$cos(A, B) = \frac{A \cdot B}{|A| \times |B|}, \tag{21}$$

where $A \cdot B$ represents the dot product of A and B, and $|A| \times |B|$ denotes the product of the magnitudes of A and B. The term $cos(A, B)$ reflects the correlation between these vectors, quantifying the cosine of the included angle between them, ranging from -1 to 1. When two vectors are more similar, this correlation tends to 1. Thus, condition a) is satisfied. Concerning condition b), it is important to note that cosine similarity inherently possesses scale in-variance. When two vectors are completely negatively correlated, the cosine similarity is -1. Yet, for the transform, as long as the residual exhibits a scaling relationship with the primary frequency basis image, the energy will be perfectly concentrated regardless of whether the scaling factor is positive or negative. To accommodate this, we modify equation (21) as follows:

$$|cos(A, B)| = \left| \frac{A \cdot B}{|A| \times |B|} \right|, \tag{22}$$

it is the absolute value of cosine similarity, better reflecting the scaling relation described in condition b).

The range of modified cosine similarity is now limited to $(0, 1)$. Considering the calculation accuracy, we enlarge it by 64 times and thus define the proposed Frequency Matching Factor as follows:

$$\begin{aligned} FMF &= \left| cos\left(\vec{X}, \vec{S_{00}}\right) \right| << 6, \\ &= \left| \frac{\vec{X} \cdot \vec{S_{00}}}{|\vec{X}| \times |\vec{S_{00}}|} \right| << 6, \end{aligned} \tag{23}$$

$\vec{X}$ and $\vec{S_{00}}$ are the expanded one-dimensional vectors corresponding to matrix X and S_{00}. For 16 transform kernels, there are 16 primary frequency basis images as depicted in Fig. 2. With (23), we can acquire the 16 FMFs, which reflect the degrees of the similarity between X to 16 S_{00}s.

The larger the FMF, the better the energy concentration effect, in other words, the lower the RDOC. To verify this view, we take 8×8 size as an example and count the relationship between FMF and RDOC. It is worth noting that the FMF is scale invariant. But, the RDOC is related to the amplitudes of the residual block. Hence, when calculating the RDOC, we normalize it to prevent any model confusion arising from scaling. This normalization is defined as follows:

$$nRDOC = \frac{RDOC}{\sqrt{E}}, \tag{24}$$

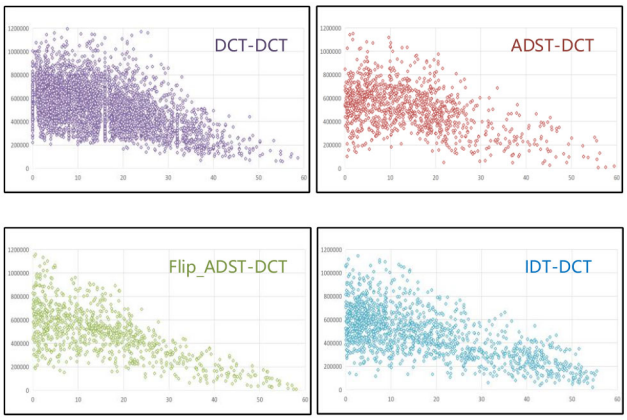


Fig. 3. Scatter diagrams depicting the relationship between nRDOC and FMF with various transformation kernels are presented. The X-axis represents FMF, while the Y-axis represents nRDOC. The data for these scatter plots belong to 8x8 data encoded by the libaom codec for the *BQSquare*.

where RDOC is the exact RDOC exported from the coding processing, E denotes the energy defined in (13), and nRDOC represents the normalized RDOC.

The statistical results exported from the coding process of sequence *BQSquare* are illustrated in Fig. 3, indicating a discernible distribution relationship between the FMF and nRDOC. Specifically, when FMF is larger, the nRDOC tends to be smaller. However, they are not strictly decreasing or inversely correlated. This is because the same FMF can correspond to different residual blocks. Even for two-dimensional vectors, the same FMF can correspond to four vectors on account of (22). As dimensions increase, the number of vectors corresponding to the same FMF increases as well. When the FMF is smaller, the residual block is less similar to the primary frequency basis image, and the number of residual blocks corresponding to this FMF will be greater. Therefore, the distribution of its costs is not concentrated. Fortunately, we do not make decisions solely on a single FMF, but use 16 FMFs corresponding to different kernels for assessment. The greater the FMF of a certain kernel, the smaller the nRDOC tends to be.

C. Introduction and Establish of Normal Models

As depicted in Fig. 3, while there is an evident correlation between nRDOC and FMF, it is not strictly linear. Therefore, solely relying on the value of FMF to predict nRDOC can potentially introduce significant errors, resulting in reduced coding performance. It is worth noting that nRDOC demonstrates distinct distribution characteristics of a single FMF. Specifically, it tends to cluster around the center and gradually disperse towards the extremes. Fig. 4a shows this phenomenon, where the frequency distribution histogram of nRDOC under FMF equals 0.5 is depicted. Intuitively, the histogram conforms to the Gaussian normal distribution. By calculating the mean and standard deviation of this data, we construct a corresponding normal probability density function curve, which can effectively model the distribution of the nRDOC. Therefore, we hypothesize that the nRDOC of each FMF follows normal distributions.

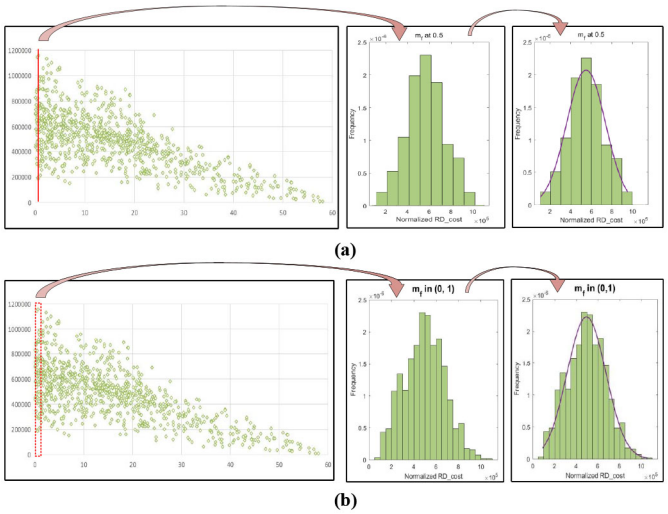


Fig. 4. (a). The left is the scatter plot corresponding to Flip_ADST-DCT. The frequency distribution histogram of nRDOC data at the FMF of 0.5 is in the middle. The right is the normal distribution curve with the mean and standard deviation calculated according to the nRDOC data. (b). The nRDOC distribution case within the FMF interval (0, 1).

To validate our hypothesis, we conducted a hypothesis test using the Kolmogorov-Smirnov (K-S) method [44], which is appropriate for the abundant sample size. In the K-S test, data follows a normal distribution if the obtained significance level (p-value) exceeds 0.05. Upon applying the K-S test to the data in Fig. 4, we obtained a p-value of 0.49, which surpasses the 0.05. Hence, it is reasonable to use the normal model to describe the data distribution in Fig. 4a. In addition, similar phenomena can be observed across various transform kernels, FMF, and transform sizes, with the nRDOC data consistently passing the K-S test.

However, exporting the corresponding nRDOC data for each FMF and then obtaining the Gaussian normal model by calculating the mean and standard deviation is impractical due to the limited accuracy of the computer's representation and the massive amount of data. Note that if the nRDOC normal models of different FMFs are relatively similar, the overall nRDOC distribution typically follows a normal distribution. In this case, the overall nRDOC can replace the nRDOC distribution of each FMF, as their means and standard deviations are also similar. Conversely, if the overall distribution of nRDOC is skewed, it indicates that nRDOC models of these FMFs are not similar, and such a replacement leads to severe inaccuracy. In general, the normality of the overall distribution qualitatively reflects the reasonableness of the replacement.

As shown in Fig. 4b, the normal model of the overall nRDOC data within the FMF interval (0, 1) closely resembles the normal model of nRDOC of $FMF_{0.5}$ in Fig. 4a. Hence,the overall normal model can effectively replace the model of $FMF_{0.5}$. Based on this theory, in this paper, the nRDOC distribution model of each FMF is replaced with the model of the FMF interval covering this FMF. Fig. 5a illustrates the rationality of this replacement affected by the interval length. It can be observed that as the interval length is reduced from 10 to 5, the skewed region is diminished from the interval (50, 60) to (55, 60). Furthermore, when the interval length is reduced to 1, the skewness phenomenon becomes less pronounced, and the skewed region is further reduced from (55, 60) to (57, 58). In this process, the nRDOC data in more and more regions exhibit good normality, indicating that the corresponding FMF normal models are more accurate. This observation validates that smaller FMF intervals yield more accurate normal models, better representing the nRDOC distribution.

D. Fitting of the Gaussian Normal Models

Utilizing fine-grained FMF interval divisions to construct normal models indeed enhances model accuracy, but it also leads to three challenges:

1. Loss of Accuracy: Even with relatively small interval granularities, there is still inherent inaccuracy. For instance, when the interval length equals 1, the normal model within the interval (57, 58) may not precisely describe the distribution of nRDOC, as shown in Fig. 5a.

2. High Storage Costs: To maintain accuracy, smaller intervals are required, which leads to an increased number of models. With FMF values ranging from 0 to 64 and an FMF interval length of 0.5, each transform kernel requires storage for 128 normal model parameters. Considering nine different transform sizes and two floating-point parameters (mean and standard deviation) for one normal model, it stores a total of $128 \times 9 \times 2 \times 16 = 36864$ floating-point numbers, increasing storage overhead significantly.

3. Slow Indexing Speed: After obtaining the frequency matching coefficient, the system must determine which of the 128 intervals it falls into. This indexing process consumes time and hardware computational resources.

To overcome the challenges mentioned above, we skillfully employ a fitting method to obtain the parameters of the normal models. As demonstrated by the scatter plot in Fig. 4, the central distribution is smooth, indicating a clear relationship between the mean value and the FMF, making it suitable for fitting. In addition, the envelope of the scatter plot is smooth, indicating a relationship between the standard deviation and the FMF that can be effectively fit. Based on this observation, we extracted the mean and standard deviation for all intervals when the interval length is 1. Intuitively, the distribution of mean and standard deviation, as we have been analyzing, has an obvious correlation with FMF. Therefore, we chose to characterize the mean and standard deviation as functions of FMF. To ensure fitting accuracy, we utilized quadratic polynomials for fitting. The fitting results, including fitted curves and correlation coefficients, are presented in Fig. 5b. Across different transform kernels and sizes, the goodness of fit for the mean and standard deviation is over or near 0.9, demonstrating that the normal model can be obtained very accurately by fitting.

This fitting approach has effectively addressed the three challenges mentioned earlier. Since the fitting function is continuous, segmenting the FMF into intervals is unnecessary. After obtaining the FMF, we can directly substitute it into the fitting function to derive the parameters of the corresponding normal model: the mean and standard deviation.

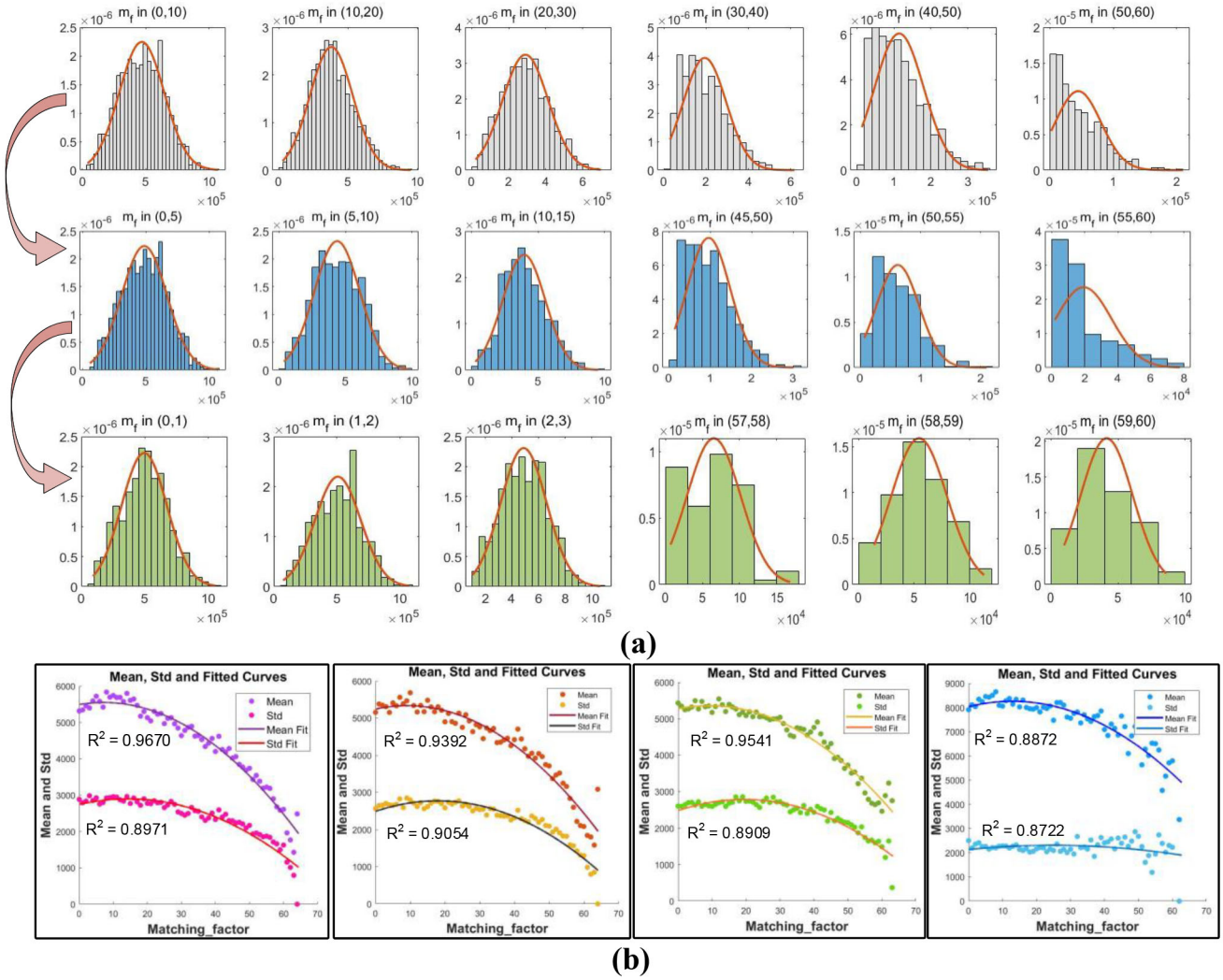


Fig. 5. (a) The relationship between the accuracy of the normal model and the length of FMF interval. As the interval length progressively decreases, the accuracy of describing the data distribution using the normal model continuously improves. (b) The mean and standard deviation of the normal model are represented as functions of the FMF, and a quadratic polynomial is employed to fit this relationship.

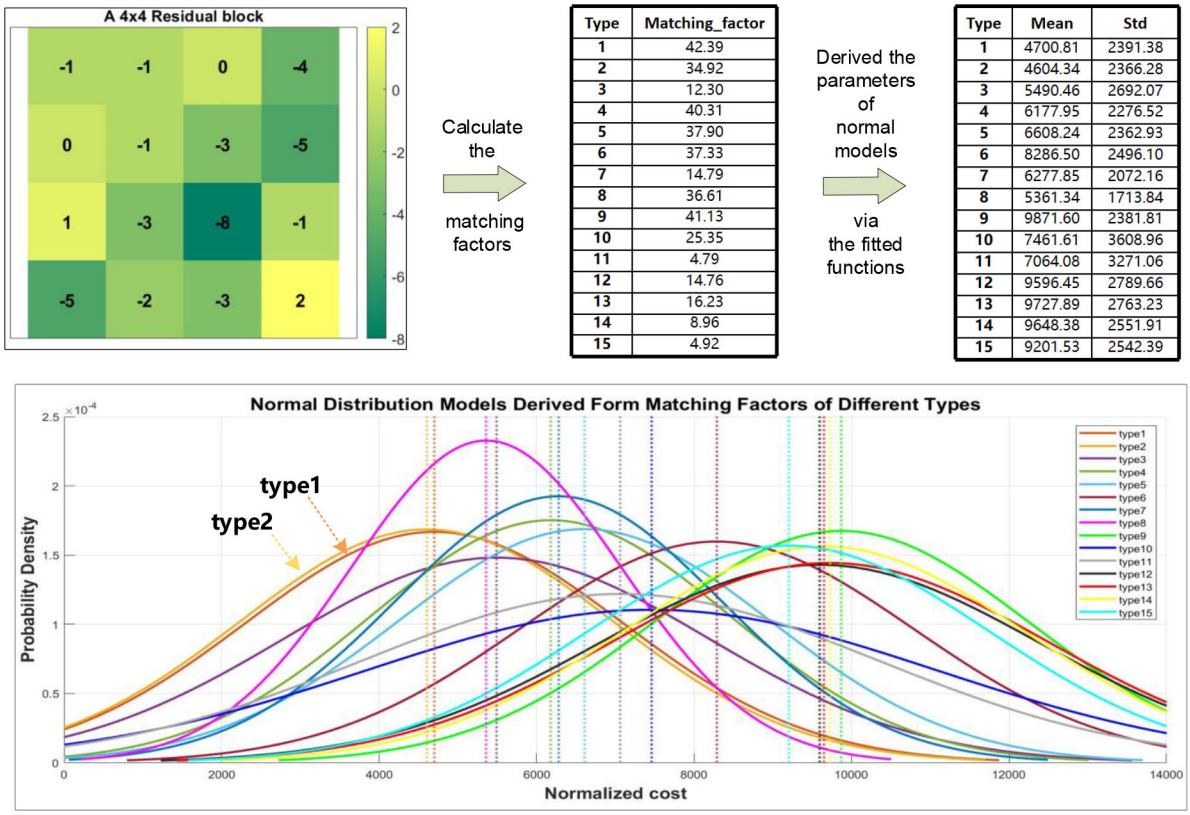
This eliminates the need for storing and accessing numerous models. Additionally, it avoids the reduced normal model accuracy due to segmentation.

To illustrate the process of acquiring the normal model more clearly, Fig. 6 presents a concrete example. For a 4x4 residual block, start by employing Equation (23) to calculate its FMFs corresponding to 15 transform kernels (excluding type0). Next, for each FMF, substitute it into the fitting functions, and the normal model of nRDOC for each transform kernel can be obtained. Fig. 6 visually depicts these models of nRDOC across different transform kernels. It's evident that type 1 and type 2 have smaller mean values, indicating that they tend to result in lower RDO costs and are potential to be selected as the optimal mode. The standard deviation reflects the degree of concentration of RDO cost of the current transform kernel. As shown in Fig. 6, the type with the smallest mean value is type2, but the actual optimal type selected by the encoder is type1. It is the existence of standard deviation that leads to this phenomenon, as there is a superposition between the nRDOC normal models corresponding to these two transform kernels. Only relying on the means, we are not absolutely sure we can determine the optimal transform kernel. Therefore, considering the influence of mean and variance jointly, we propose a fast algorithm based on the probability model.

E. Proposed Fast Selection Based on Normal Models

Although these obtained normal models can reflect the distribution of nRDOC relatively accurately, it is not advisable to use the normal model directly to make decisions. There are two main reasons. On the one hand, having the smallest mean RDO cost does not necessarily mean it's the optimal choice, as shown in Fig. 6. On the other hand, relying solely on the normal models of two transform kernels makes it extremely difficult to determine the probability that the nRDOC of one kernel is superior to the other. Giving two random variables X_1 and X_2, whose normal models have the parameters of (μ_1, σ_1) and (μ_2, σ_2) respectively, the probability P of $X_1 < X_2$ is calculated as follows:

$$P = P\{X_1 < X_2\},$$



Type	Matching_factor
1	42.39
2	34.92
3	12.30
4	40.31
5	37.90
6	37.33
7	14.79
8	36.61
9	41.13
10	25.35
11	4.79
12	14.76
13	16.23
14	8.96
15	4.92

Type	Mean	Std
1	4700.81	2391.38
2	4604.34	2366.28
3	5490.46	2692.07
4	6177.95	2276.52
5	6608.24	2362.93
6	8286.50	2496.10
7	6277.85	2072.16
8	5361.34	1713.84
9	9871.60	2381.81
10	7461.61	3608.96
11	7064.08	3271.06
12	9596.45	2789.66
13	9727.89	2763.23
14	9648.38	2551.91
15	9201.53	2542.39

Fig. 6. One case illustrates the procedure of calculating the FMF from a 4 × 4 residual block and subsequently deriving a normal model. The diagram at the bottom visually presents these normal models.

$$= \int_{-\infty}^{\infty} P\{X_1 = t\}P\{X_2 > t\}dt, \tag{25}$$

according to the error function $erf(x)$, the $P\{X_2 > t\}$ can be calculated by

$$P\{X_2 > t\} = \frac{1}{2}\left(1 - erf(\frac{\mu_2 - t}{2\sqrt{\sigma_2}})\right), \\ = \frac{1}{2}erfc\left(\frac{\mu_2 - t}{2\sqrt{\sigma_2}}\right), \tag{26}$$

substitute (26) into equation (25):

$$P = P\{X_1 < X_2\}, \\ = \int_{-\infty}^{\infty} \frac{1}{2}P\{X_1 = t\}erfc\left(\frac{\mu_2 - t}{2\sqrt{\sigma_2}}\right)dt. \tag{27}$$

It is not realistic to calculate (27) because of the heavy resource and time consumption. Nonetheless, given a specific nRDOC C and the normal model (μ, σ) of candidate transform kernel X, it is easy to compute the probability $P(X < C)$ that the nRDOC of this kernel is lower than the specified nRDOC using (28).

$$P\{X < C\} = \frac{1}{2}erf\left(\frac{\mu_2 - C}{2\sqrt{\sigma_2}}\right). \tag{28}$$

Therefore, this paper present a progressive skip algorithm, whose pseudo-code is shown in Algorithm 1. The algorithm is mainly divided into five steps.

Algorithm 1 Algorithm of the Proposed Fast Algorithm

Input: *residual block*, 15 *main frequency basis images*, *the fitted functions of normal models*,

Output: *best_type*, *best_rdoc*

```
cal FMF 1 ~ 15
cal the parameters of 15 normal models
sort kernels X_1 ~ X_15 in the order of increasing mean value: X̃_1 ~ X̃_15
cur_rdoc = cal_rdoc (DCT_DCT)
best_type = DCT_DCT, best_rdoc = cur_rdoc,
for each X̃_i ∈ X̃_1 ~ X̃_15 do
    best_nRDOC = best_rdoc/√E
    if P{X̃_i < best_nRODC} < TH then
        skip X̃_i
    else
        cur_rdoc = cal_rdoc (X̃_i)
        if cur_rdoc < best_rdoc then
            best_type = X̃_i
            best_rdoc = cur_rdoc
        end if
    end if
end for
return best_type, best_rdoc
```

1. Calculate the FMFs corresponding to 15 transform kernels except for DCT_DCT, which is the default transform kernel and must enter the RDO process.

2. Compute the parameters of the normal model corresponding to each FMF, including the mean and standard deviation, using the fitting functions.

3. Sort transform kernels in ascending order of their mean values. Since smaller mean values usually bring lower RDO costs, this sorting is more likely to skip the non-promising transform kernels, lead to a earlier finish of examination.

4. Compute the RDOC of DCT_DCT and set this RDOC as the current best RDOC. Then, traverse the sorted transform kernels. The probability that the nRDOC of the current kernel is lower than the best nRDOC is computed according to (28). If this probability falls below a certain threshold, it suggests that this kernel is unlikely to become the optimal mode. Therefore, it is skipped and will not enter the RDO process.

5. Update the cost and type if the RDOC of the non-skipped kernel is less than the best RDOC. Otherwise, retain the original model and cost. Continue this process until all the candidate kernels have been evaluated.

This algorithm comprehensively considers the influence of mean and standard deviation on decision-making. In summary, the mean affects ranking in Algorithm 1, Line 3, and mean and standard deviation jointly affect probability calculation in Algorithm 1, Line 8.

IV. Experimental Results

A. Performance Evaluation

We carry out our experiments using the AV1 reference software libaom v3.3.0, implementing the proposed fast transform kernel selection algorithm. Sequences recommended in [45] are tested under the configurations with QP values of 20, 30, 40, and 50. Besides, for aligning the test conditions of [30], some AV1 sequences tested in [30] are also tested. To ensure the fairness of the comparison, we reproduce all the methods in the compared list, and ensure that all the methods are under the same test conditions when evaluating.

To evaluate the performance of the proposed method, we measured several metrics, including the BD-BR, the encoding time saving, and the skip ratio of transform kernels relative to the anchor. Specifically, the time-saving (TS) and skip ratio are defined as (29) and (30):

$$TS(\%) = \frac{T_{Anchor} - T_{tested}}{T_{Anchor}} \times 100\%, \tag{29}$$

$$Skip\_ratio(\%) = \frac{N_{candidate} - N_{RDO}}{N_{candidate}} \times 100\%, \tag{30}$$

T_{Anchor} represents the total encoding time of libaom without employing any fast transform kernel selection algorithms, while T_{tested} corresponds to the encoding time of libaom with fast methods. $N_{candidate}$ is the number of transform kernels that are initially considered for testing, and N_{RDO} refers to the number of these kernels that actually enter the RDO process.

B. Performance Comparison

In libaom, when employing DCT_DCT as the only transform kernel, the average BD-BR performance loss is 4.78%, and the time-saving is 31.75%. While this configuration significantly reduces encoding time, such a performance degradation is deemed unacceptable in practical coding scenarios. After embedding our method into the encoder, the performance loss is notably mitigated to just 1.15%. This performance comes with a remarkable 57.66% skip ratio and a 20.09% reduction in encoding time.

To better illustrate the effectiveness of the proposed, we conducted a comparative analysis with similar approaches in existing literature as presented in Table II.

In [30], the Laplace matrix is employed to predict the bit rate after transformation, and transformation kernels with higher bit rates are excluded to achieve acceleration. This method benefits from the sparsity of the Laplacian matrix, which results in lower computational complexity compared with origin transforms. However, it's worth noting that [30] provides only a basic outline of the algorithm without offering a robust solution for weight selection. Consequently, this method suffers from low selection accuracy and significant degradation in coding performance. When the skip rate is 59.63% and time savings is 18.24%, the algorithm results in a performance loss of 2.37%, which is considerably higher than the algorithm proposed in this paper.

The method in [33], initially designed for the VVC standard, is effective in eliminating transform kernels among the five available in VVC that are unlikely to be optimal. However, when applied to AV1, where 16 transform kernels need to be tested, the limitations of the method become apparent, resulting in significant performance degradation. In other words, this method is not suitable for AV1. Notably, even with a skip rate of 82.09%, it already leads to a substantial 2.75% performance loss.

Method in [22] is a Neural Network-based (NN-based) algorithm eventually integrated into libaom. By carefully designing the neural network's hidden layers, this method achieves a good balance between performance and complexity. Additionally, since neural networks are naturally suited for classification and decision-making tasks, this approach outperforms all conventional solutions. As indicated in Table III, it incurs a 1.02% performance loss when the time saving is 21.7%, which slightly outperforms the proposed method.

In summary, compared to other methods found in the literature that are not based on neural networks, our algorithm stands out with superior performance in both compression efficiency and acceleration. In comparison to the neural network-based method, our approach remains highly competitive. Although the coding performance may slightly lag behind it, our acceleration benefits are similar. In addition, our proposed method offers scalability by adjusting the skip probability threshold. A scalable algorithm allows us to fine-tune its parameters, enabling a trade-off between various factors such as encoding time and performance. In the proposed algorithm, the scalability factor is represented by the skip probability TH, as defined in Algorithm 1, line 8. By configuring TH to different values, we can achieve diverse combinations of skip ratios and performance levels, briefly presented in Table IV. In this paper, 0.3 is selected because a similar acceleration effect can be achieved compared to the optimal performance algorithm (neural network). In this case, the effectiveness of the two algorithms can be fairly demonstrated by comparing

TABLE II
BD-BR (%), TIME SAVING (%), AND SKIP_RATE (%) OF THE PROPOSED METHODS AND OTHER WORKS AGAINST ANCHOR

Sequence	Resolution	[30] with Laplacian Operators			[33] with pixel values			Ours		
		BD-BR(%)	TS(%)	Skip_rate(%)	BD-BR(%)	TS(%)	Skip_ratio(%)	BD-BR(%)	TS(%)	Skip_ratio(%)
Tango2	Class A1 & A2 3840x2160	0.9	28.92	50.74	0.9	19.03	81.10	0.8	21.81	71.93
Campfire		0.9	18.70	57.82	1.1	34.32	82.80	0.3	36.71	58.00
CatRobot1		1.2	40.44	50.10	1.4	55.42	81.45	0.7	44.93	54.15
DaylightRoad2		2.0	1.43	53.10	2.0	56.48	81.63	0.5	49.56	45.00
ParkingRunning3		1.9	39.92	58.88	1.7	42.26	82.50	1.4	10.04	68.10
Traffic	Class A 2560x1600	1.6	43.40	59.16	1.6	12.81	81.95	0.7	9.62	55.78
PeopleOnstreet		2.0	40.04	60.94	1.7	31.07	82.85	1.4	23.40	72.55
ParkScene	Class B 1920x1080	2.2	11.23	58.44	2.0	28.10	82.05	1.2	22.15	43.55
Cactus		2.5	2.04	55.74	2.6	59.87	82.05	0.9	31.87	38.83
MarketPlace		1.9	1.31	58.32	1.7	56.13	81.78	1.3	26.93	56.25
RitualDance		0.9	12.57	65.74	0.8	15.15	82.70	0.7	8.43	80.85
BQTerrace		3.1	40.90	55.10	3.8	22.99	82.33	1.0	12.57	31.38
BasketballDrill	Class C 832x480	1.1	1.57	59.16	1.2	30.69	82.43	0.9	17.93	68.40
BQMall		3.1	1.52	57.20	4.0	21.51	82.55	1.7	4.02	59.10
PartyScene		2.8	5.13	55.72	3.7	37.08	83.03	1.4	37.83	53.68
RaceHorsesC		2.4	10.94	57.06	3.0	37.68	82.90	1.3	33.35	59.73
BasketballPass	Class D 416x240	4.2	11.73	61.38	4.9	23.53	82.08	1.4	11.67	56.58
BQSquare		4.6	11.23	53.12	7.6	10.73	81.15	1.4	16.46	41.43
BlowingBubbles		3.6	1.44	54.30	4.8	32.99	81.45	1.2	23.80	44.30
RaceHorses		2.6	1.90	58.82	3.2	39.40	82.68	1.9	27.69	56.00
FourPeople	Class E 1280x720	1.4	2.72	62.66	1.5	10.58	81.73	1.3	2.43	69.80
Johnny		1.5	1.81	64.68	1.8	10.29	81.35	1.1	7.91	64.30
KristenAndSara		1.2	0.65	62.80	1.4	12.42	81.35	1.0	14.52	66.60
AV1_Bus	AV1 352x288	4.1	36.11	59.88	4.6	29.53	82.85	1.6	16.52	56.98
AV1_foreman		2.1	34.46	64.86	1.5	26.36	81.75	1.1	19.05	67.53
AV1_Mobile		3.3	36.21	63.58	4.1	26.99	81.53	0.7	1.36	62.78
AV1_Crew	AV1 704x576	3.0	35.79	62.78	2.6	26.86	82.35	1.8	21.20	63.25
AV1_Harbour		4.3	36.71	57.46	5.4	19.65	82.13	1.4	8.70	47.78
Average		**2.37**	**18.24**	**59.63**	**2.75**	**29.64**	**82.09**	**1.15**	**20.09**	**57.66**

TABLE III
THE PERFORMANCE COMPARED TO ML-BASED METHOD

Method	Neural Network-based Algorithm [22]	Ours
BD-BR	1.02%	1.15%
TS	21.7%	20.09%

TABLE IV
THE SCALABILITY OF THE PROPOSED METHOD

TH	0.1	0.3	0.5	0.7
Skip_ratio	42.3%	57.66%	63.92%	80.4%
BD-BR	0.83%	1.15%	1.48 %	1.82%
TS	16.71%	20.09%	24.16%	28.54%

BD-BR. This flexibility caters to the specific demands of speed and accuracy in different scenarios, enhancing the algorithm's adaptability.

In our algorithm, we utilize the fitting method to derive the distribution model for each FMF. In Section III-C, we statistically demonstrate the effectiveness of this method, and here, we conduct a group of experiments to validate its efficacy. Focusing on 8x8 residual blocks, we acquire normal models with interval lengths of 10, 5, and 1, respectively. The performance is evaluated under a skip threshold of 0.3, and the experimental results are presented in Table V. Notably, a discernible trend emerges: as the interval length decreases, the corresponding performance loss also diminishes. This empirical observation aligns seamlessly with our earlier analysis, affirming the effectiveness of the fitting method.

TABLE V
THE PERFORMANCE UNDER OBTAINING NORMAL MODELS WITH INTERVAL LENGTHS OF 10, 5, AND 1 FOCUSING ON 8 × 8 RESIDUAL BLOCKS

Interval Length	10	5	1
Skip_ratio	59.94%	61.98 %	59.14%
BD-BR	0.84%	0.79%	0.75 %

It is worth noting that noticed that compared to the NN-based method, our method delivers similar acceleration benefits under a lower skip ratio. This suggests that the computational load associated with the neural network used for pruning is substantial in [22], far exceeding that of our method. This drawback often makes the NN-based method less desirable for hardware implementation, as it requires massive additional computational resources to run the neural network. Furthermore, when transitioning to hardware, quantizing the neural network to fixed-point arithmetic is necessary. This fixed-point conversion inevitably leads to a loss of accuracy. In other words, the performance loss of NN-based method after fixed-point quantization will exceed 1.02%.

The comparison of similar works has been discussed in detail. To fully demonstrate the effectiveness of the proposed

TABLE VI
THE COMPARISON BETWEEN THE PROPOSED METHOD AND OTHER FAST ALGORITHMS IN AV1 CODING PROCESS

Works	[46]	[47]	[48]	[49]	Ours
Fast methods	BP*	MD*	MD	MD	TKS*
BD-BR	0.83%	1.28%	0.95 %	7.41%	1.15%
TS	22.48%	29.80%	12.01%	50.19%	20.09%

∗ BP means the Block Partition, MD means the Mode decision, and TKS means the Transform Kernel Selection.

algorithm, we also compare it with the algorithms for speeding up other encoding processes in AV1, and the results are shown in Table VI. It can be seen that even compared with the algorithms of the preceding stage, such as block partition and mode decision, the proposed algorithm is considerably competitive in acceleration effect and coding performance. Moreover, the proposed method is orthogonal to these fast algorithms, suggesting that they can work together to speed up the coding process without conflicts.

C. Hardware-Friendliness Analysis

Here, we would like to emphasize another key advantage of the proposed algorithm: hardware-friendliness. In our algorithm, a large part of the computation is dedicated to acquiring the FMF, particularly in the matrix operation specified in (23). To reduce the complexity of this acquisition, we implemented two optimizations in the FMF calculation process:

Optimization 1). Scaling and Rounding of Primary Frequency Basis Images: In the original primary frequency basis image, each element is a floating-point number, computed from (9). However, floating-point operations consume more computational resources and time. Therefore, we have chosen to round these elements. Taking advantage of the scale invariance mentioned in Section III-B, we have chosen to upscale the primary frequency basis image by a factor of 128 and then round it, eliminating floating-point operations in matrix calculations while preserving performance with minimal loss.

Optimization 2). Leveraging Fitting Approach to Derive Certain Computational Intensive FMFs: Calculating the FMFs corresponding to certain transform kernels that are not composed of ASDT and Flip_ADST is relatively easy, as they have several identical rows or columns in their primary frequency basis images as shown in Fig. 2. Therefore, when calculating the $\vec{X} \cdot \vec{S_{00}}$ defined in (23), with the concept of consolidating identical types, we can accumulate the rows and columns of the residual blocks in advance and then perform operations with $\vec{S_{00}}$. This approach can significantly save the number of multipliers required for the computation. However, this method cannot be applied to compute the FMF corresponding to the transform kernels composed of ASDT and Flip_ADST. This is because the elements in their primary frequency basis images are different, lacking any identical elements that can be merged. To mitigate the difficulty of obtaining these FMFs, we employ several FMFs that are easier to acquire to fit them through linear combinations. Specifically, we use the FMFs corresponding to C_A, C_F, C_I, A_C, F_C, and I_C to fit the FMFs corresponding to A_A, A_F, F_A, and F_F. The results of the goodness of fit are illustrated in Table VII. The average goodness of fit for each size for the FMFs approaches 0.9. This outcome indicates that the fitting method is highly effective. By utilizing this approach, we further reduce the computational complexity of the matrix operations involved in the FMF calculation process.

TABLE VII
THE GOODNESS OF FITTING FOR CERTAIN FMFS UNDER DIFFERENT SIZES

size	goodness of fitting				
	A_A	**A_F**	**F_A**	**F_F**	**Average**
4x4	0.8817	0.8813	0.8742	0.8779	0.8788
8x8	0.8811	0.8877	0.8808	0.8882	0.8845
16x16	0.8720	0.9027	0.8891	0.9090	0.8932
4x8	0.8834	0.8790	0.8755	0.8812	0.8798
8x4	0.8952	0.8870	0.8723	0.8863	0.8852
8x16	0.8776	0.8936	0.8950	0.9012	0.8919
16x8	0.8812	0.8983	0.8911	0.8922	0.8907
4x16	0.8694	0.8871	0.8848	0.8836	0.8812
16x4	0.9116	0.9045	0.9046	0.8984	0.9048

TABLE VIII
THE ABLATION EXPERIMENTS OF TWO HARDWARE OPTIMIZATION

Methods	Baseline (w/o Opt 1&2)	with Opt 1	with Opt 1&2
Skip_ratio	56.63%	56.65 %	57.66%
BD-BR	1.12%	1.13%	1.15 %

The performance given in Table II - Table V is the test result after integrating the two hardware-friendly operations. In order to prove that these two operations will not lead to obvious performance degradation, we provide related ablation experiments, as shown in Table VIII. According to the experimental results, after integrating these two optimizations, the performance degradation is only 0.03% compared with the baseline, which is almost negligible.

Taking the $N \times N$ residual block as an example, we perform a simple analysis of hardware resource consumption for the methods in the compared list. In our algorithm, the main computing resources focus on calculating FMF and normal models. For 15 FMFs, the Equation (23) is calculated, where $|\vec{S_{00}}|$ is pre-calculated, and $|\vec{X}|$ needs N^2 multipliers $N^2 - 1$ adders, which can be shared across all FMFs. The resource for calculating the $|\vec{X} \cdot \vec{S_{00}}|$ is shown in Fig. 7a. Thus, the total resources for 15 FMFs are $N^2+15+10N+25 = N^2+10N+40$ multipliers, $N^2 - 1 + 2N^2 + 8N + 10 = 3N^2 + 8N + 9$ adders, and 15 dividers. For obtaining the normal models, the fitting method is adopted to derive the mean and standard deviation, with the fitting equation $y = ax^2 + bx + c$, requiring $2 \times 3 = 6$ multipliers and $2 \times 2 = 4$ adders per model. For 15 FMFs, the normal models can be obtained with 90 multipliers and 60

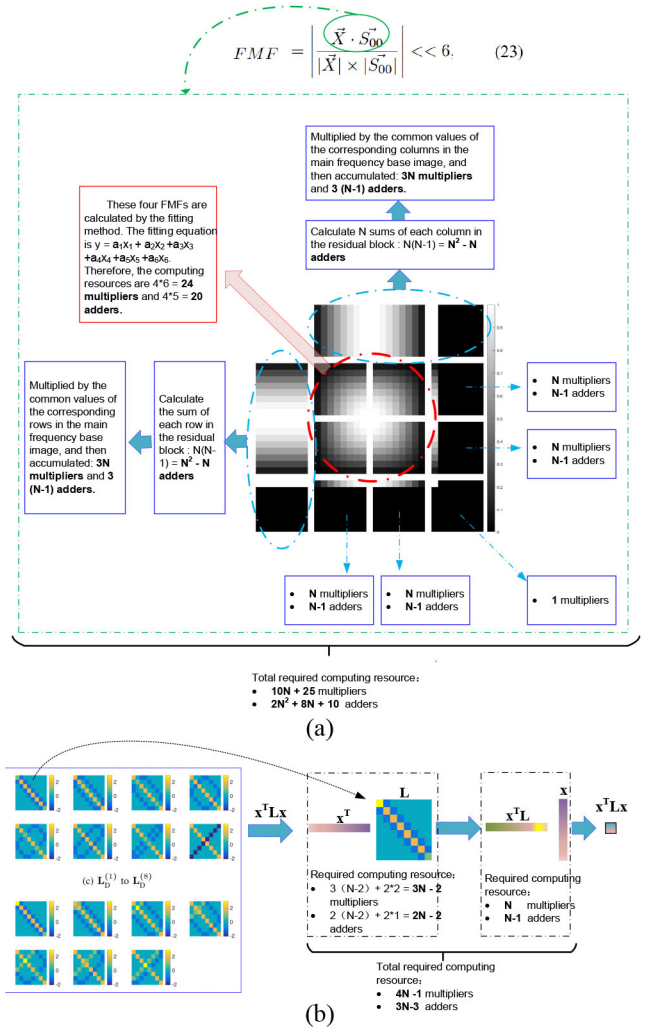


Fig. 7. (a). The resource analysis of $x^T Lx$ for a single L (corresponding to a single type). (b). The resource analysis for FMF calculation.

adders. Therefore, the main resources in our scheme are about $N^2 + 10N + 130$ multipliers, $3N^2 + 8N + 69$ adders, and 15 dividers.

For the algorithm in [30], the main computing resources focus on the calculation of $x^T Lx$, where L represents the Laplacian matrices and each row/column of it has at least three non-zero values. Therefore, to get the rate cost of a single row/column, $4N - 2$ multipliers, and $3N - 3$ adders are required, as depicted in Fig. 7b. For all rows in one direction, the computing resources required are $4N^2 - 2N$ multipliers and $3N^2 - 3N$ adders. Extending to two-dimensional evaluation doubles the resources to $8N^2 - 4N$ multipliers and $6N^2 - 6N$ adders. Additionally, it is necessary to calculate the costs corresponding to three basic 1D triangular types (DCT, ADST, and Flip_ADST), resulting in $24N^2 - 12N$ multipliers and $18N^2 - 18N$ adders. For evaluating the IDTX, this scheme calculates the residual energy E (same as the definition in the paper) to reflect the rate, requiring N^2 multipliers and $N^2 - 1$ adders. In total, the computing resource becomes $25N^2 - 12N$ multipliers and $19N^2 - 18N - 1$ adders.

As for [33], this scheme only needs to compare the relative sizes of pixel values on four corners without the need for addition and multiplication operations, and basically, no additional computing resources are consumed. When N equals 8, the resources of the Laplacian operators-based scheme are about 1504 multipliers and 1071 adders. That of the proposed scheme are about 274 multipliers, 325 adders, and 15 dividers, which present an apparent resource advantage to the former.

Once the FMFs and normal models have been obtained, the probability calculations can be handled efficiently and cheaply with the lookup table-based methods. In contrast to neural network-based algorithms, the computational resources required are comparatively minor, which highlights the hardware friendliness and efficiency of our approach. The feature of hardware-friendliness makes the proposed algorithm a dependable choice to facilitate the development and deployment of video codecs for the latest coding standards.

V. Conclusion

This paper introduces a fast transform kernel selection algorithm for AV1, leveraging the Frequency Matching Coefficient and a probability model. By analyzing the energy concentration mechanism of transform coding, we identify a critical factor influencing energy concentration: the distribution similarity between the residual block and the primary frequency basis image of the transform kernel. To quantitatively define this factor, we introduce the Frequency Matching Factor based on cosine similarity for the first time in this paper. Then, by analyzing the distribution relationship between FMF and nRDOC, we establish a normal distribution model for nRDOC. The models effectively represent the probability distribution of nRDOC of each FMF. Additionally, we characterize the model parameters as functions of FMF, enabling the rapid and accurate derivation of these models. Finally, we design a progressive skip scheme for transform kernel selection, which outperforms traditional approaches and remains highly competitive compared to neural network-based pruning schemes. Besides, the proposed method is scalable and hardware-friendly. The scalability enables a trade-off between encoding time and performance by adjusting the skipping threshold. The hardware-friendliness makes it a dependable choice to facilitate the development and deployment of video codecs based on the latest coding standards.

References

[1] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W.-J. Chien, "Enhanced multiple transform for video coding," in *Proc. Data Compres. Conf. (DCC)*, 2016, pp. 73–82.

[2] R. Clarke, "Relation between the Karhunen Loève and cosine transforms," *IEE Proc. F, Commun., Radar Signal Process.*, vol. 128, pp. 359–360, Nov. 1981. [Online]. Available: https://digital-library.theiet.org/content/journals/10.1049/ip-f-1.1981.0061

[3] X. Zhao, L. Li, Z. Li, X. Li, and S. Liu, "Coupled primary and secondary transform for next generation video coding," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, 2018, pp. 1–4.

[4] J. Han, A. Saxena, V. Melkote, and K. Rose, "Jointly optimized spatial prediction and block transform for video and image coding," *IEEE Trans. Image Process.*, vol. 21, pp. 1874–1884, 2012.

[5] J. Han et al., "A technical overview of AV1," *Proc. IEEE*, vol. 109, no. 9, pp. 1435–1462, Sep. 2021.

[6] B. Bross et al., "Overview of the versatile video coding (VVC) standard and its applications," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3736–3764, Oct. 2021.
[7] J. Zhang, C. Jia, M. Lei, S. Wang, S. Ma, and W. Gao, "Recent development of AVS video coding standard: AVS3," in *Proc. Picture Coding Symp. (PCS)*, 2019, pp. 1–5.
[8] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Video coding with rate-distortion optimized transform," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 1, pp. 138–151, Jan. 2012.
[9] Z. Zhang et al., "Fast DST-VII/DCT-VIII with dual implementation support for versatile video coding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 1, pp. 355–371, Jan. 2021.
[10] X. Zhao, J. Chen, M. Karczewicz, A. Said, and V. Seregin, "Joint separable and non-separable transforms for next-generation video coding," *IEEE Trans. Image Process.*, vol. 27, pp. 2514–2525, 2018.
[11] "Libaom." Accessed: Mar. 17, 2024. [Online]. Available: https://aomedia.googlesource.com/
[12] S. Parker et al., "On transform coding tools under development for VP10," in *Proc. XXXIX Appl. Digit. Image Process.*, 2016, Art. no. 997119. [Online]. Available: https://doi.org/10.1117/12.2239105
[13] G. Bjøntegaard, "Calculation of average PSNR differences between RD-curves," 2001. [Online]. Available: https://api.semanticscholar.org/CorpusID:61598325
[14] C. Siqueira, G. Corrêa, and M. Grellert, "Complexity and coding efficiency assessment of AOMedia video 1," in *Proc. IEEE 14th Latin America Symp. Circuits Syst. (LASCAS)*, 2023, pp. 1–4.
[15] X. Zhao et al., "Study on coding tools beyond AV1," in *Proc. IEEE Int. Conf. Multimedia Expo. (ICME)*, 2021, pp. 1–6.
[16] Z. Hao, Q. Zheng, Y. Fan, G. Xiang, P. Zhang, and H. Sun, "An area-efficient unified transform architecture for VVC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 2012–2016.
[17] M. J. Garrido, F. Pescador, M. Chavarrías, P. J. Lobo, C. Sanz, and P. Paz, "An FPGA-based architecture for the versatile video coding multiple transform selection core," *IEEE Access*, vol. 8, pp. 81887–81903, 2020.
[18] A. Kammoun, S. Ben Jdidia, F. Belghith, W. Hamidouche, J. F. Nezan, and N. Masmoudi, "An optimized hardware implementation of 4-point adaptive multiple transform design for post-HEVC," in *Proc. 4th Int. Conf. Adv. Technol. Signal Image Process. (ATSIP)*, 2018, pp. 1–6.
[19] A. Kammoun, W. Hamidouche, F. Belghith, J.-F. Nezan, and N. Masmoudi, "Hardware design and implementation of adaptive multiple transforms for the versatile video coding standard," *IEEE Trans. Consum. Electron.*, vol. 64, no. 4, pp. 424–432, Nov. 2018.
[20] Y. Fan, Y. Zeng, H. Sun, J. Katto, and X. Zeng, "A pipelined 2D transform architecture supporting mixed block sizes for the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 9, pp. 3289–3295, Sep. 2020.
[21] Z. Hao, H. Sun, G. Xiang, P. Zhang, X. Zeng, and Y. Fan, "A reconfigurable multiple transform selection architecture for VVC," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 31, no. 5, pp. 658–669, May 2023.
[22] H. Su, M. Chen, A. Bokov, D. Mukherjee, Y. Wang, and Y. Chen, "Machine learning accelerated transform search for AV1," in *Proc. Picture Coding Symp. (PCS)*, 2019, pp. 1–5.
[23] M. Jamali and S. Coulombe, "Fast HEVC intra mode decision based on RDO cost prediction," *IEEE Trans. Broadcast.*, vol. 65, no. 1, pp. 109–122, Mar. 2019.
[24] H. L. Tan, C. C. Ko, and S. Rahardja, "Fast coding quad-tree decisions using prediction residuals statistics for high efficiency video coding (HEVC)," *IEEE Trans. Broadcast.*, vol. 62, no. 1, pp. 128–133, Mar. 2016.
[25] M. Xu and B. Jeon, "Complexity-efficient dependent quantization for versatile video coding," *IEEE Trans. Broadcast.*, vol. 69, no. 3, pp. 832–839, Sep. 2023.
[26] N. Li, Y. Zhang, and C.-C. J. Kuo, "High efficiency intra video coding based on data-driven transform," *IEEE Trans. Broadcast.*, vol. 68, no. 2, pp. 383–396, Jun. 2022.
[27] Y. Li, G. Yang, Y. Song, H. Zhang, X. Ding, and D. Zhang, "Early intra CU size decision for versatile video coding based on a tunable decision model," *IEEE Trans. Broadcast.*, vol. 67, no. 3, pp. 710–720, Sep. 2021.
[28] C. Grecos and M. Y. Yang, "Fast inter mode prediction for P slices in the H264 video coding standard," *IEEE Trans. Broadcast.*, vol. 51, no. 2, pp. 256–263, Jun. 2005.
[29] L. Trudeau, S. Coulombe, and C. Desrosiers, "Cost-based search ordering for rate-constrained motion estimation applied to HEVC," *IEEE Trans. Broadcast.*, vol. 64, no. 4, pp. 922–932, Dec. 2018.
[30] K.-S. Lu, A. Ortega, D. Mukherjee, and Y. Chen, "Efficient rate-distortion approximation and transform type selection using Laplacian operators," in *Proc. Picture Coding Symp. (PCS)*, 2018, pp. 76–80.
[31] S. De-Luxán-Hernández et al., "Block adaptive selection of multiple core transforms for video coding," in *Proc. Picture Coding Symp. (PCS)*, 2016, pp. 1–5.
[32] C.-W. Wong and W.-C. Siu, "Transform kernel selection strategy for the H.264/AVC and future video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 11, pp. 1631–1645, Nov. 2011.
[33] Z. Wang, J. Wang, J. Yang, C. Luo, F. Liang, and K. Huang, "A fast transform algorithm for VVC intra coding," in *Proc. 11th Int. Conf. Commun., Circuits Syst. (ICCCAS)*, 2022, pp. 237–240.
[34] T. Fu, H. Zhang, F. Mu, and H. Chen, "Two-stage fast multiple transform selection algorithm for VVC intra coding," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, 2019, pp. 61–66.
[35] M. Saldanha, G. Sanchez, C. Marcon, and L. Agostini, "Fast transform decision scheme for VVC intra-frame prediction using decision trees," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 1948–1952.
[36] L. He, S. Xiong, R. Yang, X. He, and H. Chen, "Low-complexity multiple transform selection combining multi-type tree partition algorithm for versatile video coding," *Sensors*, vol. 22, no. 15, p. 5523, Jul. 2022. [Online]. Available: http://dx.doi.org/10.3390/s22155523
[37] H. Schwarz et al., "Quantization and entropy coding in the versatile video coding (VVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 31, no. 10, pp. 3891–3906, Oct. 2021.
[38] D. Marpe, H. Schwarz, and T. Wiegand, "Entropy coding in video compression using probability interval partitioning," in *Proc. 8th Picture Coding Symp.*, 2010, pp. 66–69.
[39] C.-W. Wong, O. C. Au, and R. C.-W. Wong, "Advanced macro-block entropy coding in H.264," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, vol. 1, 2007, pp. 1169–1172.
[40] X. Zhao, L. Shan, G. Adrian, and N. Andrey, "Tool description for AV1 and Libaom," Oct. 2021. [Online]. Available: extension://bfdogplmndidlpjfhoijckpakkdjkkil/pdf/viewer.html?file=https%3A%2F%2Faomedia.org%2Fdocs%2FAV1 ToolDescriptionv11-clean.pdf
[41] A. Kammoun et al., "Forward-inverse 2D hardware implementation of approximate transform core for the VVC standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 30, no. 11, pp. 4340–4354, Nov. 2020.
[42] O. Guleryuz and M. Orchard, "Optimized nonorthogonal transforms for image compression," *IEEE Trans. Image Process.*, vol. 6, pp. 507–522, 1997.
[43] M. Loog, "On distributional assumptions and whitened cosine similarities," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 6, pp. 1114–1115, Jun. 2008.
[44] D. J. De Priest, "Testing goodness-of-fit for the singly truncated normal distribution using the Kolmogorov-Smirnov statistic," *IEEE Trans. Geosci. Remote Sens.*, vol. GE-21, no. 4, pp. 441–446, Oct. 1983.
[45] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," ITU, Geneva, Switzerland, document JVET-B1010, 2016.
[46] J. Gu and J. Wen, "Mid-depth based block structure determination for AV1," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, 2019, pp. 1617–1621.
[47] M. Corrêa, D. Palomino, G. Corrêa, and L. Agostini, "Heuristic-based algorithms for low-complexity AV1 intraprediction," *IEEE Design Test*, vol. 40, no. 5, pp. 26–33, Oct. 2023.
[48] M. Corrêa, N. Roma, D. Palomino, G. Corrêa, and L. Agostini, "Mode-adaptive subsampling of SAD/SSE operations for intra prediction cost reduction," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, 2022, pp. 1808–1812.
[49] P. Rosa, D. Palomino, M. Porto, and L. Agostini, "GM-RF: An AV1 intra-frame fast decision based on random forest," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, 2022, pp. 3556–3560.

Zhijian Hao received the B.S. degree from the Department of Electronic Engineering, Xidian University, Shanxi, China, in 2019. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China. His research interests include a wide range of topics related with image processing, video processing, video coding, and associated VLSI architecture.

Heming Sun (Member, IEEE) received the B.E. degree in electronic engineering from Shanghai Jiao Tong University, Shanghai, China, in 2011, the M.E. degree from Waseda University and Shanghai Jiao Tong University in 2012 and 2014, respectively, through a double-degree program, and the Ph.D. degree from Waseda University. He was a Researcher with NEC Central Research Laboratories from 2017 to 2018. He was an Assistant Professor with Waseda University from 2018 to 2023. He is currently an Associate Professor with Yokohama National University, Japan. His interests are in algorithms and VLSI architectures for video processing and neural networks. He got several awards, including IEEE Computer Society Japan Chapter Young Author Award, IEEE VCIP Best Paper Award, and PCS Top-10 Best Paper Award.

Guohao Xu received the B.S. degree from the Department of Microelectronic, Xidian University, Shanxi, China, in 2020. He is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China. His research interests include video coding, video processing, and associated VLSI architecture.

Jiaming Liu received the B.E. degree in electronic science and technology from Southwest Jiaotong University, Chengdu, China, in 2019, the M.S. degree in circuit and system from Southwest Jiaotong University, Chengdu, in 2022, He is currently pursuing the Ph.D. degree with the State Key Laboratory of Integrated Chips and Systems, Fudan University, Shanghai, China. His research interests include image processing, channel coding, machine learning, and associated VLSI architecture.

Xiankui Xiong was born in 1973. He is the Chief Architect of ZTE Advanced Computing & Storage. His research interests include heterogeneous computing, high-efficiency computing, and advanced computing paradigm.

Xuanpeng Zhu was born in 1980. He received the M.Eng. degree. His research interests include artificial intelligence, computer vision, and video coding and decoding.

Xiaoyang Zeng (Member, IEEE) received the B.S. degree from Xiangtan University, Xiangtan, China, in 1992, and the Ph.D. degree from the Changchun Institute of Optics, Fine Mechanics, and Physics, Chinese Academy of Sciences, Changchun, China, in 2001. From 2001 to 2003, he was a Postdoctoral Researcher with Fudan University, Shanghai, China. Then, he joined the State Key Laboratory of ASIC and System, Fudan University, as an Associate Professor, where he is currently a Full Professor and the Director. His research interests include information security chip design, system-on-chip platforms, and VLSI implementation of digital signal processing and communication systems.

Yibo Fan received the B.E. degree in electronics and engineering from Zhejiang University, Hangzhou, China, in 2003, the M.S. degree in microelectronics from Fudan University, Shanghai, China, in 2006, and the Ph.D. degree in engineering from Waseda University, Tokyo, Japan, in 2009. He was an Assistant Professor with Shanghai Jiao Tong University and Fudan University from 2009 to 2014, and an Associate Professor with Fudan University from 2014 to 2019, where he is currently a Full Professor with the College of Microelectronics. His research interests include image processing, video coding, machine learning, and associated VLSI architecture.