A Novel Video Coding Framework using Self-adaptive Dictionary

Yuanyi Xue, Student Member, IEEE, and Yao Wang, Fellow, IEEE

Abstract—In this paper, we propose to use a self-adaptive redundant dictionary, consisting of all possible inter and intra prediction candidates, to directly represent the frame blocks in a video sequence. The self-adaptive dictionary generalizes the conventional predictive coding approach by allowing adaptive linear combinations of prediction candidates, which is solved by an rate-distortion aware L0-norm minimization problem using orthogonal least squares (OLS). To overcome the inefficiency in quantizing and coding coefficients corresponding to correlated chosen atoms, we orthonormalize the chosen atoms recursively as part of OLS process. We further propose a two-stage video coding framework, in which a second stage codes the residual from the chosen atoms using a modified DCT dictionary that is adaptively orthonormalized with respect to the subspace spanned by the first stage atoms. To determine the transition from the first stage to the second stage, we propose a rate-distortion (RD) aware adaptive switching algorithm. The proposed framework is further extended to accommodate variable block sizes (16 × 16, 8 × 8, and 4 × 4), and the partition mode is derived by a fast partition mode decision algorithm. A context-adaptive binary arithmetic entropy coder is designed to code the symbols of the proposed coding framework. The proposed coder shows competitive, and in some cases better rate-distortion performance, compared to the HEVC video coding standard for P-frames.

Index Terms—Video coding, Sparse coding, Rate-distortion optimization, L0-norm minimization

I. INTRODUCTION

The currently prevalent video coding framework (e.g. HEVC [1]) relies on a block based hybrid coding scheme. An input frame is first partitioned into non-overlapping frame blocks, the blocks are coded successively, often in a row-major scanning order. For each block, a predictive coding is first used to find an approximation of the block by considering either spatial or temporal neighboring blocks, which are referred as intra and inter prediction, respectively. The prediction error, is then coded by transform coding, where a fixed orthonormal transform basis is frequently used. The transform coefficients are quantized and all symbols involved in this process are input to an entropy coder to produce the bitstream.

Although the hybrid coding scheme has demonstrated its great capability in exploiting the spatial and temporal redundancy in video signals, there are intrinsic limitations that can be lifted, which can lead to further improvement on the coding efficiency. In this work, we propose a novel video coding framework by employing the idea of sparse recovery theory, which provides significantly more flexibility on how to approximate a frame block. The framework, which represents each frame block by an adaptive weighted sum of prediction candidates, generalizes the predictive coding method widely used in the existing framework. By regarding all the prediction candidates for the block as atoms in a redundant dictionary, our goal is to find a sparse representation of the block using this locally adaptive dictionary. We further propose to use a second stage transform, to code the residual after the sparse representation using the dictionary. In the following subsection, we share our motivation for developing the proposed framework.

A. Motivation

In the hybrid coding scheme, predictive coding is usually done in this way: given a frame block to be coded, a set of inter or intra prediction candidates is assembled. The inter prediction candidates consist of overlapping blocks of the same size from previously reconstructed frames, and the intra prediction candidates are formed from the neighboring casual region of the current frame. The best prediction candidate (or a predefined fixed number of candidates) is then chosen by minimizing the prediction error (which is usually defined as the sum of squared or absolute differences between the original and predicted pixels). Without loss of generality, let y denote the vectorized block, and Φ the matrix whose columns are all the prediction candidate blocks. Furthermore, assume that s(> 0) candidate blocks can be chosen. The problem can be formulated as finding a linear combination of s candidates with fixed weights that minimizes the prediction error. We would like to note that the widely used fractionalpel motion estimation fits the above formulation, with all the blocks corresponding to integer motion vectors nearest to the fractional motion vector being the s chosen candidates, and the interpolation filter coefficients being the fixed weights.

Note that, the vectorized block y is in $\mathbb{R}^N$, and the matrix $\Phi$ is in $\mathbb{R}^{N \times M}$, where the number of possible candidates $M$ is generally much larger than the signal dimension $N$. Minimizing the prediction error bears a great amount of resemblance with the sparse coding problem, which tries to represent a signal y by a sparse set of atoms in a dictionary $\Phi$, by finding the best sparse set of non-zero coefficients $x$ that minimizes the approximation error. Indeed, the predictive coding in hybrid coding scheme can be regarded as a special case of sparse coding problem, but with two major differences: (a) the number of allowable chosen candidates is pre-defined, and (b) the weights of chosen candidates are fixed. One may wonder whether it is feasible to lift these two constraints,
and this very question motivates the authors to develop the framework presented in this paper.

Specifically, for a given block $y$, we think of the matrix $\Phi$ as a redundant dictionary, where each column (corresponding to one prediction candidate) represents one dictionary atom, and we propose to directly find the most sparse set of coefficients $x$, with a squared prediction error less or equal than a threshold determined by the expected quantization level. We choose $L_0$-norm, i.e. the number of the non-zero coefficients in the solution vector $x$, as our sparsity measure.

The above formulation generalizes the predictive coding method by explicitly allowing adaptive number of prediction candidates and their corresponding weights in a per-block basis. We would also like to note that the redundant dictionary $\Phi$ has its own design flexibility. Indeed, we could incorporate both inter and intra prediction candidates in the dictionary, as well as any specially designed fixed basis.

B. Prior Works

Sparse representation of signals using redundant dictionaries has found wide use in many multimedia signal processing applications. It generally refers to represent a given signal with a sparse (few non-zero weights) linear combination of atoms in a dictionary that is overcomplete. The discussion of sparse coding (the process to determine the coefficients from a given signal) and dictionary learning is beyond the scope of this work. References [2] and [3] provide excellent exposition of various topics related to sparsity-based signal processing.

Several research groups have attempted using redundant dictionaries for block-based image and video coding, including [4]–[8]. In all reported dictionary-based video coders, the dictionary atoms are used to represent the prediction error block for intra/inter-frame video coding. Therefore, they are very different from what is proposed here. Instead of using a single dictionary, [7] uses multiple dictionaries, pre-designed for different residual energy levels. The work in [8] proposes a two-layered transform coding framework, which finds a sparse representation of the prediction error block using orthogonal matching pursuit with an overcomplete dictionary learned offline using K-SVD [9] and codes the resultant approximation error with the fixed DCT transform. The work in [10] codes each frame in the intra-mode, with a dictionary that is updated in real time based on the previously coded frames. Although such online adaptation can yield a dictionary that matches with the video content very well, it is computationally very demanding. Finally, one major challenge in applying sparse representation for compression is that the dictionary atoms are generally not orthogonal. Quantizing the coefficients associated with them directly and independently are not efficient. First of all, the quantization errors of the coefficients are related to the errors in the reconstructed samples in a complicated way. Secondly, these coefficients are likely to have high correlation. We believe that one reason that the dictionary-based coders have not been more successful is because they quantize the sparse coefficients associated with the chosen atoms directly. In the proposed framework, we propose to represent the subspace spanned by the chosen atoms by a set of orthonormal vectors, which are derived by orthonormalizing the chosen atoms. As we will show in the later sections, the coefficients corresponding to these orthonormal vectors are much less correlated and can be quantized and coded independently without losing coding efficiency.

The earlier versions of the proposed framework have been published in [11], [12]. Specifically, in [11], we proposed a one-stage framework that used the self-adaptive redundant dictionary to directly represent the frame blocks. This framework is extended to incorporate a second stage modified DCT transform in [12]. The work presented in this paper is based on our earlier works but extended significantly. Indeed, we propose a modified rate-distortion aware orthogonal least squares (OLS) algorithm that maximizes the bits-normalized residual norm reduction recursively when finding the sparse coefficients in the first stage. We also propose an adaptive switching algorithm for determining the optimal switching point from the self-adaptive dictionary to the second stage DCT dictionary. We further extend the proposed framework to accommodate variable block sizes, and propose a fast partition decision algorithm.

C. Paper outline

The remainder of the paper is presented as follows: in Section II, we give the formal problem formulation of using a self-adaptive redundant dictionary for video coding and demonstrate its efficiency in representing frame blocks; in Section III, we elaborate the rationale of bringing in an additional stage to the proposed framework, and present the complete two-stage video coding framework; in Section IV, we describe the rate-distortion aware design of the proposed framework, including an RD optimized OLS algorithm and an adaptive switching algorithm between self-adaptive and DCT dictionaries; in Section V, we extend the proposed framework to accommodate variable block sizes and present a fast partition decision algorithm; in Section VI, we describe the entropy coder designed specifically for coding the locations of chosen atoms and their corresponding coefficient values; in Section VII, we describe remaining implementation details of the proposed framework, compare the rate-distortion performance of the proposed coder with existing video coding standards, H.264 and HEVC, and conduct a complexity comparison analysis between the proposed coder and HEVC. We conclude our paper in Section VIII.

II. VIDEO CODING USING SELF-ADAPTIVE REDUNDANT DICTIONARY

In this section, we lay out the problem formulation on using self-adaptive redundant dictionary to represent frame blocks. We also show the efficiency of using self-adaptive redundant dictionary compared to the popular hybrid coding method.

A. Finding Sparse Representation using Self-adaptive Dictionary

In Section I, we motivate our work by generalizing the predictive coding as a sparse coding problem of minimizing
L0-norm with a redundant dictionary. Specifically, for every frame block, a dictionary is used to represent the block. The columns of this dictionary consists of atoms from two parts: (a) all the vectorized inter-frame motion estimation candidates, which include all possible overlapping blocks with different integer displacements from the current block location given a preset search range; and (b) all the vectorized intra-frame prediction candidates, each corresponding to one of the HEVC intra-frame prediction modes. The dictionary is self-adaptive because it depends on the location of the block in a frame. Also note that the number of possible inter/intra-prediction candidates is generally significantly higher than the block size (signal dimension), which implies the dictionary is redundant.

Indeed, in this paper, we use a displacement range of $[-23, 24]$ both horizontally and vertically, yielding a total of 2304 inter-frame candidates, plus additional 27 HEVC intra-prediction modes$^1$. Therefore we name this redundant dictionary hereafter as Self-Adaptive Dictionary, or SeAD for short. We now give the formulation of the proposed framework.

For a block in a video frame, we first perform mean subtraction, and we also make each atom in the corresponding SeAD to be zero mean and unit L2-norm. Note that both the encoder and decoder need to perform the same mean subtraction and normalization on the prediction candidates formed from previously decoded frame/area, to form the same dictionary. Denoting $y$ as the vectorized representation of the mean-removed block to be coded, and $\Phi$ as the SeAD. Note that $y \in \mathbb{R}^N$ and $\Phi \in \mathbb{R}^{N \times M}$, where $N$ is the block size and $M$ is the number of SeAD atoms. We derive the sparse representation for $y$ using $\Phi$ by solving the following constrained optimization problem:

$$\arg\min_x \|x\|_0 \quad s.t. \quad \|y - \Phi x\|^2_2 \leq N\epsilon^2 \quad (1)$$

Note that this is the L2-bounded noise variant of the classical “L0 sparse recovery” problem [13]. Typically, the L0-norm problem is solved by Orthogonal Matching Pursuit (OMP) [14], [15]. As an iterative greedy algorithm, OMP consists of three major steps in each iteration: (i) an atom is chosen to maximize the correlation between the atom and the current residual vector; (ii) the coefficients for the chosen atoms are derived/updated based on the orthogonal projection of the residual onto the chosen atoms; (iii) the residual is updated by incorporating the new atom in the solution. The iteration is terminated when the fidelity constraint is met.

Although OMP is shown to be competitive for L0-norm sparse coding, it is less desirable in solving (1). Specifically, the solution vector (coefficients, $x$) by OMP is associated with highly correlated, non-orthonormal atoms in SeAD. As mentioned in Section I-B, directly applying quantization on those coefficients is not efficient. Furthermore, the correlation maximization procedure in OMP is not guaranteed to pursuing the atom that can best represent the subspace that the current residual may reside (which is the null space of the chosen atoms). Intuitively, this is because the remaining atoms very likely still have a large energy on the subspace already spanned by the previous chosen atoms.

From the above perspective, we use Orthogonal Least Squares (OLS) for solving (1). The main difference between OLS and OMP is that at each iteration, OLS orthonormalizes the remaining atoms with respect to chosen atoms. More specifically, starting at the second iteration, OLS recursively performs a one-step orthonormalization of all remaining atoms with respect to the chosen atoms. Denoting the last orthonormalized atom in the chosen set $\mathbf{D}$ by $d$, the one-step orthonormalization operates on each remaining atom $\psi_i$ by:

$$\psi_i \leftarrow \psi_i - \langle d, \psi_i \rangle d$$

$$\psi_i \leftarrow \frac{\psi_i}{\|\psi_i\|_2} \quad (2)$$

Because all the remaining atoms are orthonormalized with respect to the previously chosen atoms, the next atom can be simply chosen to maximize the correlation of the current residue with the remaining atoms, and the coefficient corresponding to the chosen atom $z_t$ is simply the inner product of the residual with the chosen atom $\psi_t$:

$$z_t = \langle r, \psi_t \rangle \quad (3)$$

We refer to [16] and [17] for the details of the algorithm, and the proof that OLS maximizes the residual norm (distortion) reduction. Note that, by OLS, the resultant coefficients are with respect to the orthonormalized atoms. In Sec. IV, we further refine this algorithm by considering the bit cost for coding the chosen atom, and present the RD aware atom selection algorithm in Algorithm 1.

Given a set of chosen orthonormalized atoms $\mathbf{D}$, denote its corresponding coefficients by $z$, we apply uniform quantization with deadzone to all but the first coefficients in $z$ with the same stepsize $q$, and deadzone $\Delta = q/6$ following the HEVC standard recommendation for inter-coded blocks. For the first coefficient, we predict it using the L2-norm of the mean-removed prediction candidate corresponding to the first chosen SeAD atom, and the prediction error is then quantized using the same quantizer as for the remaining SeAD atoms. Such prediction is motivated by the fact that this prediction candidate is typically the one that best matches with the current block.

Finally, the mean of the block is predicted using the mean of the first chosen SeAD atom, and the mean prediction error is always quantized with stepsize $= 1$. We use the smallest quantization for the mean prediction error since the mean value quantization error significantly affects the final image reconstruction quality.

**B. Evaluating the Efficiency of Self-adaptive Dictionary**

In this section, we demonstrate the effectiveness of using SeAD to represent a block by comparing the sparsity and variance of the coefficients resulting from using SeAD and hybrid coding scheme. In all cases reported below, we use a fixed quantization stepsize $= 8$. Figure 1 compares the distributions of the number of non-zero coefficients needed using $16 \times 16$

---

$^1$HEVC has a total of 35 intra-prediction modes, but 8 angular prediction modes are not available because the difference of the causal region between HEVC and proposed framework.
A. Why Do We Need Two-stage?

We start our discussion with a plot demonstrating the motivation for incorporating the second stage. In Fig. 3, the continuous blue curve shows the distortion versus the total bits for coding chosen atoms for an exemplar block in sequence Football, when only SeAD is used. The beginning part of the curve shows a very fast decay of the distortion, which indicates the great RD efficiency of the SeAD. However, after a few iterations, the decaying rate significantly slows down for each newly chosen atom. This is in part because the latter atoms are generally placed in a random location and hence their locations need a large number of bits to code.

In addition to the continuous blue curve, we also show two red curves. They represent the distortion reductions using DCT basis had the first stage stopped after a few atoms were chosen. In the left red curve, it is apparent that stopping first stage too early yields sub-optimal result, i.e. at that point, SeAD is still more RD efficient (faster decay). Interestingly, the right red curve shows an opposite trend, that using DCT basis becomes more RD efficient by providing a faster decay slope compared to continuing with the SeAD. This observation also inspires us to develop an adaptive switching method from the first stage to the second stage, which switches to the second stage whenever the DCT basis gives faster decay rate in the distortion vs. rate profile.

B. Two-stage Framework with Adaptively Orthonormalized DCT

The proposed two-stage framework is illustrated in Fig. 4. We describe each component in more details below.

The first stage is inherited from the framework presented in the previous section. After applying quantization, let \( \hat{z} \) denote the size of the redundant dictionary, specifying the index of a chosen atom demands significantly more bits than the index of a transform coefficient, which impacts the final rate-distortion performance. In this section, we present a two-stage framework, which completes our proposed method. The two-stage framework incorporates a second stage transform, which is used for coding the residuals from SeAD representation. In this work, we choose an adaptively orthonormalized DCT transform in the second stage.
the quantized first stage coefficients, the residual \( r = y - D \hat{z} \) is the input of the second-stage.

Given the information about which atoms are chosen and the residual \( r \), the DCT basis vectors are first orthonormalized with respect to the first stage chosen atoms. The coefficients of the residual with respect to these altered DCT basis vectors are then derived simply by performing inner product of the residual vector with each altered DCT basis vector. These coefficients \( c \) are quantized with the same quantization stepsize \( q \) using a deadzone quantizer. The residual \( r \) is therefore reconstructed as \( T \hat{c} \), where \( T \) contains the altered DCT basis vectors and \( \hat{c} \) includes the quantized second stage coefficients. Finally the reconstructed block \( \hat{y} \) is given by

\[
\hat{y} = D \hat{z} + T \hat{c}
\]  

(4)

By orthonormalizing the DCT basis with respect to the chosen first-stage SeAD atoms, we force the resultant altered DCT bases represent the null space of the chosen SeAD atoms in the first stage, or the subspace that the residual \( r \) lies in. This will represent the residual \( r \) more efficiently than using the original DCT. In experiment, we observe this modification improves the PSNR by around 0.2dB at the same rate.

IV. RATE-DISTORTION OPTIMIZATION IN THE PROPOSED ENCODER

In this section, we propose two rate-distortion aware algorithms to be incorporated in the proposed framework. Specifically, an adaptive switching algorithm is implemented to determine when to switch from SeAD to DCT transform; and a modified OLS algorithm is proposed that factors the impact of bit-cost of choosing a specific SeAD atom. Both algorithms are based on the belief that better RD efficiency can be achieved by pursuing the steepest decay in the distortion vs. rate profile.

A. Rate-distortion Optimized Adaptive Switching

One important question for the two-stage framework is how and when to transition from SeAD to DCT. One heuristic approach is to evaluate the distortion reduction ratio at each OLS iteration, and switch to the second stage (DCT) under

\[
\frac{\Delta_{SeAD}}{b_{SeAD}} \geq \frac{\Delta_{DCT}}{b_{DCT}} \rightarrow \text{stay at first stage,}
\]

else switch to second stage.

In the proposed two-stage framework, the second stage will be skipped when the distortion from the first stage is less than a preset target determined by the expected quantization error. In the rare event where a switch does not happen even after many self-adaptive atoms have been chosen, the coder will terminate the first stage and switch to using DCT dictionary whenever the newly chosen self-adaptive atom incurs a dis-
One key ingredient in the proposed adaptive switching scheme is the introduction of bits-normalized distortion reduction. Ideally, the bits shall be estimated from the entropy coder for a chosen SeAD atom or DCT basis. However, doing this would require a significant increase of complexity for the proposed coder. Furthermore, our preliminary study found that in most cases, the bit-cost behavior is heavily influenced by the location of chosen SeAD atom. Therefore to simplify the process, we instead choose to design two sets of LUTs for the location-dependent estimated bit cost per SeAD atom and that per DCT atom, respectively, based on the offline training data obtained using the entropy coder we will describe in Section VI. Bits spent both on the location and coefficients are counted and different LUTs are derived for different quantization step sizes. We employ a tiered design for each LUT, which partitions all possible atoms into different zones. Three tiers are used for SeAD atoms and 4 tiers are used for DCT basis, as shown in Fig. 5 and 6. Besides, we use a separate LUT for the first SeAD atom only, as it is coded differently than later atoms and usually has much smaller bit cost than the remaining SeAD atoms. We would also like to note that the use of tiered design minimizes the storage overhead of loading a large LUT that contains one bit-cost estimate per location, and it also minimizes the bias from using the bit cost estimated derived from the training data.

Figure 5 shows the average bit cost per SeAD atom location and the resultant look-up table for quantization step size at 8. It is as expected that the actual bit cost is lower at the center location, which corresponds to the inter-prediction candidates with smaller displacement values. We also observe that the atoms located along the zero horizontal and vertical displacement axes have smaller bit cost. In our LUT design, we reflect this observation by having a low bit cost zone (shown blue in Fig. 5) along the vertical and horizontal axes. Figure 6 shows the aggregated bit cost per DCT atom location and the resultant look-up table for quantization step size at 8. Similarly, we observe a tiered bit cost loosely following the entropy coder scanning order. We would also like to note that the aggregated bit cost in the lower right region (corresponding to high frequency DCT coefficients) is shown blue to indicate the absence of reliable bit cost data. The resultant LUT is nevertheless designed by including that region in the 4th tier.

B. Rate-distortion Aware OLS Algorithm

The original OLS algorithm in the first stage selects SeAD atom by finding the one (orthonormalized with respect to previously chosen atoms) that best correlates with the residual at each iteration, without considering the bit-cost of the chosen atom. It is quite possible this chosen atom is not rate-distortion optimal, in the sense that it may require more bits to code (for both its location and magnitude) than some other atoms, even though it leads to the highest distortion reduction. Here we propose an improved OLS algorithm that selects the atom by maximizing the distortion reduction, which is equivalent to minimizing the bits-normalized correlation. This effectively is pursuing the steepest descend in the rate-distortion profile at each iteration. In our implementation, we use the same bit-cost LUTs described in the previous section to estimate the bits for coding a chosen atom. The proposed RD aware OLS algorithm is summarized in Algorithm 1.

Algorithm 1 Rate-distortion Aware Orthogonal Least Squares

Input: \( \Phi \in \mathbb{R}^{N \times M}, y \in \mathbb{R}^{N \times 1}, \epsilon, B. (\Phi = [\phi_1, t = 1, 2, \ldots, M]) \) is the SeAD dictionary, \( y \) has zero mean and unit L2-norm, and \( B \) is the LUT that specifies the estimated number of bits needed to code the location and coefficient of each atom.

1. \( r \leftarrow y, K = \emptyset, K^c = \{1, 2, \ldots, M\}, \Phi = \Phi, D = 0, t = 1. (t \) indicates current residual, \( \Psi \) contains remaining unchosen atoms, \( D \) includes all chosen orthonormalized atoms).
2. \( \text{while } ||r||_2 > \epsilon \text{ do} \)
3. \( \text{Update remaining atoms } \Psi \leftarrow \Psi_{\{., i\}}, \forall i \in K^c, \text{ by performing one-step orthonormalization of all columns in } \Psi \text{ w.r.t the last chosen atom } \psi_{t-1} \)
4. \( \psi_t \leftarrow \text{argmax}_{\psi_k \in \Phi} \frac{||y_k - r||_2}{||y_k||_2} \) and \( k \) - index of \( \psi_t \)
5. \( \text{Update chosen index set } K \leftarrow K \cup \{k\} \text{ and remaining index set } K^c \leftarrow K^c \setminus \{k\} \)
6. \( D \leftarrow [D; \psi_t] \)
7. \( z_t \leftarrow (r, \psi_t) \)
8. \( \text{Update } r \leftarrow r - z_t \psi_t \)
9. \( z \leftarrow [z/z_t] \)
10. \( t \leftarrow t + 1 \)
11. \( \text{end while} \)
12. \( \text{return the set of chosen atom index } K, \text{ chosen orthonormalized atoms } D, \text{ solution vector } z \text{ w.r.t } D. \)

C. Evaluating the Effectiveness of RD Aware Algorithms

We demonstrate the impact of RD-aware adaptive switching and bit-normalized OLS by comparing the RD performance with adaptive switching and improved OLS, and that with threshold-based switching without RD aware OLS. Figure 7 shows the RD plots for an inter-frame in sequence City and Football. We observe a significant RD improvement in the high-rate region by incorporating the RD-aware methods, but the low-rate region shows a mixed result. We suspect the low-rate region performance is undermined because the bit cost...
estimates in the LUTs and the distortion estimates for the quantizer are less accurate at the low rates.

We also demonstrate the rate-distortion efficiency of using the second-stage by comparing the PSNR vs. average number of non-zero coefficients at different quantization levels ($q$). Table I shows the numbers of chosen SeAD and DCT atoms in the two stage coder, and the number of chosen SeAD atoms in the one-stage coder at various $q$ for sequence Football. In order to get a fair comparison, for the two-stage framework, we use RD-aware adaptive switching and keep all non-zero DCT coefficients. In the one-stage framework, we keep choosing SeAD atoms until the PSNR matches that of the two-stage coder at the same $q$. For both cases, the RD-aware OLS implementation is used. From the table we can see that the two-stage framework significantly reduces the number of chosen SeAD atoms, even though the total number of chosen atoms (including DCT) is higher. This validates our early discussion that even later SeAD atoms are still more powerful at representing the frame blocks than DCT basis. However, because the total number of SeAD atoms is much larger than the total number of DCT bases, coding the location of each chosen SeAD atoms takes more bits than that of a DCT basis, the two-stage frame work leads to a better rate-distortion efficiency.

V. EXTENSION TO VARIABLE BLOCK SIZES

In all previous sections, the proposed framework works on a single $16 \times 16$ block size. We choose the $16 \times 16$ as the single block size primarily based on our early study that $16 \times 16$ finds a good balance between efficiency and complexity. However, it is straightforward to see that block size $16 \times 16$ may not be the best choice for all image regions. Indeed, as we will demonstrate in this section, better RD efficiency can be achieved for some blocks if smaller block sizes are considered. On the other hand, larger than $16 \times 16$ block sizes are excluded, mainly from the complexity and efficiency trade-off of the proposed framework. In this section, we propose an extension of the two-stage framework by enabling quad-tree partitioning structure within each $16 \times 16$ block, with possible partition to $8 \times 8$ and $4 \times 4$ sub-blocks. Similar to HEVC, we allow a mix-match selection of partition sizes within every $16 \times 16$ block. However, unlike HEVC, in this work we identify the best partitioning for the first stage and second stage jointly.

A. Allowing Variable Block Sizes Improves RD Efficiency

We start our discussion on variable block sizes by demonstrating the potential efficiency improvement by allowing variable block sizes. In Fig. 8, the black dash curve shows the distortion versus accumulated bits spent had we only used a fixed $16 \times 16$ block size. We also draw a red solid curve, which presents the distortion versus bits by coding the block using four $8 \times 8$ sub-blocks. The bits spent for sub-block case counts all four sub-blocks. The first star(•) marker on the red curve uses one atom per $8 \times 8$ block, resulting a total of 4 atoms used. The subsequent star markers depict additional $8 \times 8$ atoms used for sub-blocks. By evaluating the slope of the two curves, which corresponds to the bits-normalized distortion reduction, the red curve shows a steeper decay than the black curve, which translates to better RD efficiency.

Figure 8 also shows that coding the first atom at any block sizes requires smaller number of bits. Indeed, to code the four first $8 \times 8$ atoms, we only need less than 30 bits, comparing to later atoms that usually requires more than 11 bits per atom. This is due to the fact that the first atom is usually located around the center of the SeAD location map, and our entropy coder is able to predictively code the first atom location very efficiently.

The above observation also provides insight about how to decide the best partitioning for the proposed framework. One straight-forward approach is to code a block with all the possible partitions and choose one that requires least amount of bits with the same distortion reduction. However, it is very expensive to exhaustively evaluate the complete RD profile for all the possibilities. Instead, we propose a fast partition mode...
decision algorithm that approximates the rate-distortion slope by evaluating only first few atoms.

B. Fast Algorithm for Deciding Quad-tree Partitioning Structure

Now we give the description of our fast algorithm for deciding quad-tree partitioning. For every $16 \times 16$ block, the algorithm decides which partition mode shall be used by evaluating the bits-normalized distortion reduction on the first few atoms, and the best partition mode is given by the one with highest bits-normalized distortion reduction. LUTs for different block sizes are used for estimating the bits.

Specifically, the following steps are followed:

1) Partition the $16 \times 16$ block into four $8 \times 8$ sub-blocks. For each $8 \times 8$ sub-block, further partition it into four $4 \times 4$ sub-blocks. For each $4 \times 4$ block, choose the best SeAD atom by RD aware OLS. Note this yields a total of 4 SeAD atoms to present the $8 \times 8$ sub-block. Determine the distortion reduction $\Delta_d$ with these 4 atoms (at $4 \times 4$ level) and estimate the total number of bits $b_4$ needed to code the 4 atoms.

2) For the same $8 \times 8$ sub-block, run a “mini” two-stage framework with adaptive switch that only allows to select up to 4 atoms. Determine the distortion reduction $\Delta_b$ with these 4 atoms (at $8 \times 8$ level) and estimate the total number of bits $b_8$ needed to code the 4 atoms. The 4 atoms can be a combination of SeAD and DCT atoms. DCT atoms are included only if the adaptive switching algorithm decides to switch to DCT before the 4th SeAD atom is chosen.

3) Compare the bits-normalized distortion reductions for this $8 \times 8$ sub-block using different models, i.e. $\frac{\Delta_d}{b_4}$ and $\frac{\Delta_b}{b_8}$, choose the mode that gives larger bits-normalized distortion reduction.

4) Repeat steps 1)-3) for all four $8 \times 8$ sub-blocks. Record the overall distortion reduction $\Delta_p$ and corresponding bits $b_{np}$, where each $8 \times 8 \times 8 \times 8$ sub-block uses the best mode (with or without partition to $4 \times 4$) determined from steps 1)-3).

5) Run the two-stage framework for the $16 \times 16$ block to select up to 16 atoms (SeAD and DCT combined). Evaluate the bits-normalized distortion reduction $\frac{\Delta_{np}}{b_{np}}$.

6) Compare $\frac{\Delta_p}{b_{np}}$ and $\frac{\Delta_{np}}{b_{np}}$, choose the partition mode that has the larger value.

Note that it is crucial for the fast partition decision algorithm to facilitate fair comparison between partition and non-partition modes. We explicitly enforce that by matching the number of total atoms used between best partition mode and non-partition mode, and evaluating the resultant bit-normalized distortion reduction for making the decision. In addition, the fast mode decision algorithm is built on the belief that the rate-distortion profile estimated by the first few atoms provides sufficient insight on their respective RD efficiency, which we have observed through Fig. 8.

If a sub-block partition mode is chosen, the coder processes each sub-block in a raster scan order. We code each sub-block following the same two-stage framework procedure described in previous sections.

C. Partition Modes Statistics

Before the end of this section, we give some visualization on how the partition modes are chosen for different video contents. Figure 9 shows exemplar frames from two sequences, (a) City and (b) Crowd_Run, coded at quantization steps $= 8$ with variable block sizes and fast mode decision. We superimpose the partition modes used for every $16 \times 16$ frame block upon the frames. Three interesting observations can be made by inspecting their partition mode statistics. First of all, we see in both cases, a majority number of blocks are coded at the $16 \times 16$ block size, which justifies our early study on using $16 \times 16$ blocks. Second, except a few blocks in the fine detail region, $4 \times 4$ block sizes are seldom used. Finally, most of the partition modes are in regions with texture details and/or edges of the objects. This is as expected and demonstrates that the fast mode decision algorithm is effective in selecting the most efficient partitioning modes.

Table II lists the statistics of partition modes for City and Crowd_Run at different quantization levels. For each sequence at every quantization level, we give the number of $16 \times 16$ blocks that are coded without any partition (16 only), with all $8 \times 8$ partitions only (8 only), and with a mixture of $8 \times 8$ and $4 \times 4$ partitions (mixed 4 and 8). In both sequences, we see the trend that the coder increasingly prefers non-partitioning $16 \times 16$ block size as the quantization level increases. It is expected since at high quantization level, majority of the blocks only need very few atoms (one SeAD...
atom plus a few DCT atoms) at $16 \times 16$ block, and further partitioning is deemed not RD efficient because the need to code additional SeAD atoms for each sub-partitions.

D. Notes on Intra-frame Coding with Proposed Framework

Before we move on to the entropy coder design, we would like to point out that although we have described the proposed framework in the context of inter-frame coding, the proposed framework can also be adopted to code intra-frame with minimal changes. Specifically, in coding intra-frame, one could use the causal encoded region of the frame to assemble SeAD atoms based on the same principle, and all other components and RD aware algorithms can be directly used for intra-frame coding. However, we choose not to report the intra-frame coding result in this paper, because we found the proposed framework does not show competitive rate-distortion performance compared to HEVC for intra-frames. This is likely because our current entropy coder was not optimized for the intra-coding case.

VI. CONTEXT-ADAPTIVE BINARY ARITHMETIC CODER

This section describes how we code the relevant symbols of the proposed coder by a context-adaptive binary arithmetic coder, inspired from the CABAC design in current video coding standards [18]. We restrict ourselves here to provide the description of only the essential components of our entropy coder without diving into why and how each component is designed. We refer the readers to [19] for the complete design and discussion of the entropy coder used in the proposed framework.

The information we need to code for each block (at the chosen block size) consists of mainly three parts: a) the location of the chosen SeAD atoms; b) the levels of each quantized coefficients; and c) second stage DCT coefficients. We design the entropy coder for coding SeAD atom locations and levels, and we use HEVC’s entropy coder for coding DCT coefficients directly. Similar to HEVC’s CABAC coder, we define two probability update modes: the normal mode where the probability of coding each bin is updated based on the past data; and the bypass mode where the probability is fixed and uniformly distributed.

A. Coding SeAD Atom Locations

We index all inter-prediction candidates in the search range of $(-23, 24)$ in a $48 \times 48$ location map with horizontal index indicating the horizontal displacement, and the vertical index indicating the vertical displacement of the inter candidate. For the 27 atoms derived from HEVC intra-frame prediction modes, we assign them to the top row of the location map, with indices $(dx = (-13, \ldots, 13), dy = -24)$. We code the location of each chosen atom by its horizontal and vertical position in this location map (to be referred to as the motion vector).

For the first atom, we predictively code the location using the locations of the first chosen atoms of its neighboring blocks. Specifically, we use the median of the motion vectors defining the first chosen atoms of three neighboring blocks (i.e. to the top, left, and top-left blocks) of the current block as the predicted motion vector, and code the difference. Similar to HEVC, the difference in each direction is encoded by its absolute value and sign. Since the absolute value is observed to have a geometric distribution, truncated unary/k-th order Exponential-Golomb (UEG3) code with a cut-off value of 7 is used for binarizing the absolute value of the difference. The sign bit is sent directly in the entropy coder and coded using the bypass mode.

For the later atoms, we do not follow the predictive approach because the spatial location correlation among the corresponding atoms in neighboring blocks are generally weak for those atoms. Instead, we choose to code the locations of later atoms directly. The absolute value of each location component is binarized using truncated-unary code with an alphabet size of 25, and the sign is encoded in bypass mode. Since we observe different statistics on the absolute values of motion vectors for different atoms, five different sets of context models are assigned to the second, third, forth, fifth, and all remaining atoms. Finally, the EOB symbol (indicating no more SeAD atoms are coded) is treated as a special case of the absolute value of the x-position, and the actual absolute values of horizontal locations of the atoms (0 to 23) are increased by 1 to avoid the ambiguity of symbols.

B. Coding SeAD Atom Coefficients

The symbol for the first coefficient is the quantized predictive residual from the norm of the first chosen SeAD atom (before normalization). This symbol is directly coded by truncated unary/k-th order exponential-Golomb code. Remaining coefficient levels are encoded in a reversed atom chosen order, similar to the entropy coder for the DCT coefficient levels of the HEVC. We code a level sequence by its absolute levels and signs separately. For absolute values, we predict each absolute quantization level using that of the next chosen atom, except the last one in each block, which is always predicted by the value of 1. Consequently, we specify absolute levels using a sequence of prediction residuals, in the reversed chosen order, and binarize each of them by a truncated unary/k-th order exponential-Golomb code with a cut-off value of 7. Lastly, the prefix bins of binarized residuals, which corresponds to the truncated-unary parts of the symbol before cut-off value, are encoded using the normal mode; while the suffix bins, which corresponds to the k-th order exponential-Golomb code of remaining values larger than 7, along with the sign part, are encoded using the bypass mode. Table III illustrates an example of coding SeAD atom coefficients.

<table>
<thead>
<tr>
<th>City</th>
<th>Quantization Step Sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>4</td>
</tr>
<tr>
<td>16 only</td>
<td>1283</td>
</tr>
<tr>
<td>8 only</td>
<td>1943</td>
</tr>
<tr>
<td>mixed 4 and 8</td>
<td>574</td>
</tr>
<tr>
<td>CrowdRun</td>
<td>16 only</td>
</tr>
<tr>
<td></td>
<td>8 only</td>
</tr>
<tr>
<td>mixed 4 and 8</td>
<td>248</td>
</tr>
</tbody>
</table>
TABLE III
EXAMPLE OF CODING SEAD ATOM QUANTIZED COEFFICIENT LEVELS

<table>
<thead>
<tr>
<th>SeAD Chosen Order</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coding Order</td>
<td>←</td>
<td>←</td>
<td>←</td>
<td>←</td>
<td>←</td>
<td>←</td>
<td>←</td>
</tr>
<tr>
<td>Coefficient Levels</td>
<td>2</td>
<td>3</td>
<td>5</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Absolute Levels</td>
<td>2</td>
<td>1</td>
<td>5</td>
<td>3</td>
<td>2</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Predicted Residuals</td>
<td>2</td>
<td>2</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Sign of Coefficient Levels</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

TABLE IV
CONFIGURATION USED FOR H.264 AND HEVC

<table>
<thead>
<tr>
<th>Software version</th>
<th>HEVC bi-pred</th>
<th>HEVC 64×64</th>
<th>HEVC 16×16</th>
<th>H.264</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profile</td>
<td>Main</td>
<td>Main</td>
<td>Main</td>
<td>High</td>
</tr>
<tr>
<td>GOP structure</td>
<td>IPPPP</td>
<td>IPPPP</td>
<td>IPPPP</td>
<td>IPPPP</td>
</tr>
<tr>
<td>Max. CTU</td>
<td>64x64</td>
<td>64x64</td>
<td>64x64</td>
<td>64x64</td>
</tr>
<tr>
<td>Weighted prediction</td>
<td>Allowed for B- and P-frames</td>
<td>Allowed for B- and P-frames</td>
<td>Allowed for B- and P-frames</td>
<td>Allowed for B- and P-frames</td>
</tr>
<tr>
<td>Ref. pictures</td>
<td>Up to 4 frames</td>
<td>Up to 4 frames</td>
<td>Up to 4 frames</td>
<td>Up to 4 frames</td>
</tr>
<tr>
<td>In-loop filters</td>
<td>Deblocking and SAO</td>
<td>Deblocking and SAO</td>
<td>Deblocking and SAO</td>
<td>Deblocking and SAO</td>
</tr>
<tr>
<td>Entropy coder</td>
<td>CABAC</td>
<td>CABAC</td>
<td>CABAC</td>
<td>CABAC</td>
</tr>
</tbody>
</table>

VII. RATE-DISTORTION EXPERIMENTS

We conducted a series of rate-distortion experiments comparing the proposed coder with two video coding standards, H.264 and HEVC. In this section, we describe the codec configurations and experimental results.

A. Codec Configurations

We configure the proposed codec for coding inter-frames as follows: a total of 2331 atoms are used for SeAD atoms, when full search range is available. Out of them, there are 2304 atoms from inter-prediction candidates corresponding to a integer search range of \([-23, 24]\) in the immediate previous reconstructed frame: the remaining 27 atoms correspond to the available HEVC intra-prediction modes. The same base quantization stepsize with deadzone quantizer is used for both first stage SeAD atom coefficients and second stage DCT coefficients. For DCT coefficients, an additional quantization matrix from HEVC [20] is used to enable frequency-adaptive quantization. For entropy coding, uniform context model is always used as the initial state, and the context model is re-initialized after the completion of a GOP, which is set to 10 in all experiments. Each GOP is set to consists of 1 I-frame and 9 P-frames.

In Table IV, we summarize the configurations used for H.264 and HEVC coders. These configurations are chosen to provide a fair comparison with the proposed coder.

B. RD Experiments and Comparisons

We show the coding performance of the proposed coder with \(16 \times 16\) single block size, and with variable block sizes, and other three comparison coders for three test sequences, Football, City, and CrowdRun. The Football sequence has a frame size \(704 \times 576\) and the other two sequences (City and CrowdRun) have a frame size \(1280 \times 720\). For all sequences, a total of 100 frames are coded and we use a GOP size of 10. Within each GOP, the frames are encoded using the IPPPP structure. For the proposed framework, we use HEVC to generate intra-frame as the head frame of every GOP and the proposed framework is used to encode the subsequent inter-frames. For HEVC, reference software HM version 12.1 [21] is used and we present three configuration settings: with HEVC \(16 \times 16\), we restrict the prediction and transform unit up to \(16 \times 16\), but allow the smaller prediction and transform sizes for a fair comparison with the proposed coder, which also limits the block size to \(16 \times 16\); with HEVC \(64 \times 64\), we do not impose restriction on block size, allowing up to size of \(64 \times 64\). In both of these two cases, four previous frames are considered for motion compensation but the RDO process chooses only one candidate. Finally, with HEVC bi-pred, we allow weighted prediction from two references chosen from 2 previous frames and 2 future frames and up to size of \(64 \times 64\) block size, we collect RD results from all inter-coded frames. For H.264, we use open-source encoder x264 with CABAC and all other advanced coding tools and subpartitions enabled (referred as H264 CABAC). Finally, we convert the total bits spent on inter-coded frames to bit-rate, by assuming a frame rate of 30Hz.

Figure 10 shows the average PSNR versus rate plots for test sequences. For all three sequences, the proposed coder with variable block sizes outperforms H.264 significantly. More importantly, it provides noticeable gain over HEVC \(16 \times 16\) and \(64 \times 64\) over the entire large rate range for Football, and is slightly better than HEVC \(16 \times 16\) and \(64 \times 64\) at the lower rate range for City and CrowdRun. When using HEVC \(64 \times 64\), we see the proposed coder maintains their improvement at low rate range for all three sequences, and the proposed coder is still competitive at high rate range for Football. Note that for these three test sequences, HEVC \(64 \times 64\) is only slightly better than HEVC \(16 \times 16\). We think this is in part because these test sequences has complex texture, so that a block of \(64 \times 64\) is typically split into smaller blocks of sizes \(16 \times 16\) or smaller for RD optimized coding. Also the sequence football is an standard definition resolution, for which the gain from using a larger block size is limited. Besides, we see by using variable block sizes, the proposed framework achieves 0.2 - 0.6dB RD improvement over using single \(16 \times 16\) block size at the high rate region, which validates our hypothesis that using smaller block size for some image regions may improve the coding efficiency. Indeed, with the option of variable block size, the proposed coder is competitive with both HEVC even at the high rate for sequences with rich and complex details like City and CrowdRun, where using a fixed block size falls short.

Finally, comparing the proposed coder with HEVC bi-pred, the proposed coder is worse than HEVC bi-pred, but the difference is small for sequences Football and CrowdRun, the two sequencies with complex motion and texture. Recall that HEVC bi-pred has b-frame coding enabled and allows up to 4 reference pictures in prediction candidates; while the proposed coder only considers SeAD atoms in one reference frame. We expect that the proposed coding method will achieve a significant performance gain if it is adapted to consider atoms in more than one frame.

Figure 11 shows two sets of exemplar snapshots taken from sequences City and Football, using reconstructed P-
frames from HEVC $64 \times 64$ and proposed coder, respectively, at similar bit rates. To demonstrate the quality difference, we also overlay two zoomed-in portion of the snapshots to show the details more clearly. We observe that at similar bitrate, the proposed coder is able to provide perceptual quality improvement over HEVC, both in high textured content like "jersey in Football" and in smooth regions like the blue football jersey in "Football".

C. Complexity Analysis of Proposed Coder

In this section, we analyze the complexity of the proposed coder, and compare that with the HEVC reference software HM. We choose not to directly compare the running time of encoding and decoding with the HEVC reference software since the proposed coder is implemented on Matlab without explicit optimization.

In the encoding process, as shown in Fig. 4, the proposed coder contains a first-stage that solves an RD-aware OLS from SeAD dictionary, a second-stage that uses a modified DCT basis to transform the residual from the first-stage, and CABAC entropy coder to code the symbols before writing to the stream. Assuming the complexity for second-stage and entropy coder is similar to the HEVC encoder, the major difference in complexity between the proposed coder and conventional coder like HEVC, is the use of OLS in the first-stage as opposed to the motion estimation and motion-compensated prediction coding. We would like to note that the SeAD dictionary in the proposed coder only contains candidates from integer offset of a predefined search range in the reference frame, instead of motion estimation candidates, which are usually from fractional-pel offsets, obtained from the interpolated reference frame. Indeed, both HEVC and H.264 use quarter-pel accuracy motion estimation with a multi-tap interpolation filter.

We now give an analysis of the complexity of the OLS used in the first-stage (see Algorithm 1 for algorithm details). Without loss of generality, we assume the majority of the complexity come from additions and multiplications.

And we assume either operation takes a same complexity. The inner product of two vectors in $\mathbb{R}^N$ is of complexity $O(2N)$. The four ingredients described in Algorithm 1 are one-step orthonormalization, correlation between remaining SeAD atoms, finding the maximum correlation SeAD atom, and residual update. The one-step orthonormalization shown in Eq. 2 includes two inner products, one addition and two scaling, the total complexity is $O(7N)$ for each remaining atom at each iteration. The correlation calculation between residual and each SeAD has a complexity of $O(2N)$, and the residual update has a complexity of $O(2N)$. Finally, finding the maximum correlation value of a given size $M$ is assumed to be linear in time $O(M)$.

Consider the case that we run Algorithm 1 on an SeAD dictionary in $\mathbb{R}^{N \times M}$, with signal in $\mathbb{R}^N$ to choose $s$ SeAD atoms. Note that the number of remaining atoms is always decreasing, we have the following for the total complexity:

$$O\left((M + M - 1 + \ldots + M - s + 1)2N\right) +$$

$$O\left((M - 1 + \ldots + M - s + 1)7N\right) +$$

$$O\left(M + M - 1 + \ldots + M - s + 1\right) + O\left(2sN\right)$$

$$= O\left(\left((2M - s + 1)s + \frac{7(2M - s)(s - 1)}{2}\right)N\right)$$

$$+ O\left(\frac{(s(2M - s + 1))}{2}\right) + O\left(2sN\right)$$

$$\approx O\left(9NM^2\right)$$

Assuming $M \gg s$, Eq. 5 is approximately $O(9NM^2)$.

For HEVC coder, assume the same search range and only one reference frame is used, note that HEVC uses quarter-pel motion estimation, so the total number of possible candidates are approximately $16 \times$ the number of SeAD atoms. The majority of the computation comes from assessing the mean absolute difference between one prediction candidate and frame block. For a block containing $N$ pixels, each comparison has a complexity of $O(N)$. Assume a fast motion search algorithm like hexagon or diamond search is used, based on

\footnote{this excludes intra-prediction modes.}
(a) Seq. City, proposed coder (PSNR: 37.9014 db, Rate: 5.6382 Mbps).
(b) Seq. City, HEVC (PSNR: 36.8512 db, Rate: 5.1448 Mbps).
(c) Seq. Football, proposed coder (PSNR: 38.7112 db, Rate: 3.0872 Mbps).
(d) Seq. Football, HEVC (PSNR: 37.4780 db, Rate: 3.0995 Mbps).

Fig. 11. Snapshots of exemplar frames coded by the proposed coder and HEVC 64×64 with zoomed-in details overlay.

the rough estimated report in numerous fast motion search works [22] [23], around one-tenth of all motion prediction search candidates are visited by those fast motion search algorithm, or $16M/10$. The resultant complexity of finding one motion estimation from the same search range as SeAD would be $O(1.6NM)$.

Now, considering the fact that HEVC generally uses multiple references. The AMVP in HEVC allows up to 16 references per one of the two reference lists. In our proposed coder, we see the average number of chosen SeAD atoms is around 6 at low quantization levels for complex sequence like CrowdRun. Therefore, it is safe to assume that when using multiple and weighted references, HEVC would use similar amount of references. Assume $s$ multiple references are used, the final complexity of finding the motion compensation of a block containing $N$ pixels would be $O(1.6NMs)$. Comparing that to the proposed coder, we shall see around $5.5 \times$ complexity increase by replacing predictive coding with proposed SeAD dictionary at given block size and search range.

A significant amount of encoding cycles is used for the mode decision, that is to determine the best block sizes for the input frame block. To limit the complexity penalty, earlier coder like H.264 operates on a constraint that the residual block size for transform coding must be the same as the block size used for motion compensation. In HEVC however, one prominent change is the introduction of coding tree unit or CTU, and it allows transform coding size (TU) different from the prediction unit size (PU). Obviously, the flexibility of PU and TU sizes places a complexity penalty on the mode decision process, as the encoder needs to go through every possible combination of PU and its valid TU sizes. Whereas in the proposed coder, we decide the block size for both stages simultaneously, by the quad-tree partitioning algorithm presented in Sec. V-B. Given a block size of $16 \times 16$ and the minimum block size of $4 \times 4$, the proposed coder only needs to consider a total of 17 possible partition combinations (i.e. $16 \times 16$, all $8 \times 8$, 4 different configurations of one $4 \times 4$, 6 different configurations of two $4 \times 4$, 4 different configurations of three $4 \times 4$, and all $4 \times 4$), where HEVC contains significant larger number of possibilities. These additional complexity of HEVC is not considered in the previous analysis.

The decoder complexity of the proposed coder is similar to
that of a conventional coder. Indeed, note that by Eq. 4, the reconstruction of a block is merely the sum of chosen SeAD and DCT atoms weighted by the corresponding coefficients. The only added complexity is the need to orthonormalize chosen SeAD and DCT atoms.

VIII. CONCLUSION AND FUTURE WORK

In this paper, we proposed a novel video coding framework that uses a self-adaptive redundant dictionary consisting of inter and intra-prediction candidates as dictionary atoms, to directly find a sparse representation of video frame blocks. To solve the sparse coding problem in the context of video coding, a rate-distortion aware orthogonal least squares algorithm is proposed, which iteratively finds the orthonormalized atom with largest bits-normalized distortion reduction. We further propose a two-stage coding framework with a RD aware adaptive switching algorithm. The proposed coder is extended to incorporate variable block sizes and a fast partition mode decision algorithm is proposed. We show the proposed coder can outperform the latest video coding standard HEVC with P-frame coding in the lower bit rate range and is competitive with that at the higher rate range.

Although our proposed coder has only shown limited gain over HEVC P-frame, and falls short to HEVC with B-frame coding, additional gains are likely with further optimization of the various components. First of all, the rate-distortion behaviors of both the first stage for coding the SeAD atoms and the second stage for coding the DCT atoms exhibit a very smooth trend, indicating that some analytical models for the rate-distortion curves of these two stages may be established, which may be utilized to yield a better rate-distortion optimized decision on the adaptive switching between the two stages and on the block partition decision. Secondly, how to design a better SeAD dictionary and a better second stage dictionary can be further explored. Specifically, the proposed framework only uses atoms from the previous frame in SeAD, while one could investigate using multiple frames, with both previous and future frames. For the second stage dictionary, although our piloting study on designing a special fixed dictionary through dictionary learning returned similar result as with using the DCT, other new development along this line might be borrowed to design a better dictionary or to adapt the dictionary. Thirdly, we believe significant gains are possible with more careful design of the entropy coder for coding the first stage atom locations. Lastly, although our preliminary attempt at extending the two stage coding framework to intra-frame coding has not yielded promising results, further optimization of the entropy coder, especially for coding the intra-prediction candidate locations, is likely to make it more competitive than HEVC.

ACKNOWLEDGMENT

The authors would like to thank Mr. Yi Zhou for his contribution to the entropy coder design presented in this work. The authors would also like to thank Dr. Anthony Vetro for insightful discussions and suggestions at various stages during the development of the proposed framework.

REFERENCES

[14] J. A. Tropp, “Greed is good: Algorithmic results for sparse approxima-

2242, October 2004.
Yuanyi Xue  Yuanyi Xue received the B.Eng. in Electronic Engineering from Southeast University, Nanjing, China, in 2008, and M.S. and Ph.D. degrees in Electrical and Computer Engineering from NYU Tandon School of Engineering, in 2010 and 2016, respectively. He joined MLB Advanced Media as Research Engineer in 2016. His research interests include sparse signal processing and machine learning techniques for video coding, convolutional neural networks for object tracking and recognition, and perceptual video quality assessment and modeling.

Yao Wang  Yao Wang received the B.S. and M.S. degrees in Electronic Engineering from Tsinghua University, Beijing, China, in 1983 and 1985, respectively, and the Ph.D. degree in Electrical and Computer Engineering from University of California at Santa Barbara in 1990. Since 1990, she has been on the faculty of Electrical and Computer Engineering, Tandon School of Engineering of New York University (formerly Polytechnic University, Brooklyn, NY). Her research areas include video communications, multimedia signal processing, and medical imaging. She is the leading author of a textbook titled Video Processing and Communications, and has published over 250 papers in journals and conference proceedings. She has served as an Associate Editor for IEEE Transactions on Multimedia and IEEE Transactions on Circuits and Systems for Video Technology. She received New York City Mayor's Award for Excellence in Science and Technology in the Young Investigator Category in year 2000. She was elected Fellow of the IEEE in 2004 for contributions to video processing and communications. She received the IEEE Communications Society Leonard G. Abraham Prize Paper Award in the Field of Communications Systems in 2004, and the IEEE Communications Society Multimedia Communication Technical Committee Best Paper Award in 2011. She was a keynote speaker at the 2010 International Packet Video Workshop, and at INFOCOM Workshop on Contemporary Video in 2014. She received the NYU Tandon Distinguished Teacher Award in 2016.