Thèse de Doctorat

Saurabh PURI

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'Université de Nantes
sous le sceau de l'Université Bretagne Loire

École doctorale : Sciences et technologies de l'information, et mathématiques

Discipline : Traitement du signal et des images, section CNU 27
Unité de recherche : LABORATOIRE DES SCIENCES DU NUMÉRIQUE DE NANTES (LS2N)

Soutenance prévue le 09 November 2017

Learning, selection and coding of new block transforms in and for the optimization loop of video coders

JURY

Rapporteurs :	**M. Olivier DEFORGES**, Professeur, INSA Rennes
	M. Marco CAGNAZZO, Associate Professeur, TELECOM-ParisTech
Examinateurs :	**M^{me} Christine GUILLEMOT**, Directrice de Recherche, INRIA Rennes
	M. André KAUP, Professeur, FAU-Erlangen-Nürnberg Allemagne
Directeur de thèse :	**M. Patrick LE CALLET**, Polytech Nantes, Université de Nantes
Co-encadrant de thèse :	**M. Sébastien LASSERRE**, Principal Scientist, Technicolor France

Contents

List of Tables

List of Figures

1

General Introduction

1.1 Context

Digital videos and images have become a vital part of our lives with an exponential increase in the use for information exchange, marketing, entertainment and many other applications. Today video alone accounts for more than 70% of all the consumer internet traffic and is expected to only increase in the coming years. This is mostly driven by the use of video heavy social networks that allow easy uploading and sharing of high quality videos directly from mobile devices capable of capturing high definition (HD) and Ultra HD (UHD) videos.

Video compression standards developed in the past 30 years have played a vital role in enabling technologies related to videos such as live broadcasting, streaming, video-conferencing, and at the same time helped in achieving inter-operability among various devices. Most of the existing compression standards (MPEG-2, H.264/AVC, HEVC) are based on the principle of hybrid video coding where the spatial redundancies are removed using transform-based coding and temporal redundancies are removed using motion compensation prediction performed from the previously coded frames. Figure 1.1 illustrates the hybrid coding method using a block diagram.

The advent of the newest video coding standard, High Efficiency Video Coding (HEVC), in January 2013 has made it possible to broadcast the Ultra-HD content on the communication channel. The HEVC provides nearly 50% bit-rate gain in comparison to the H.264/MPEG-4 standard [16]. It is shown in [17] that the subjective testing of HEVC compression provides 60% gain compared to the previous standard at high resolutions. Therefore, HEVC is most likely to replace H.264 in the coming years. This is also evident from recent announcement by Apple on extending HEVC support in the MacOS and iOS 11 [18].

However, with more and more UHD content appearing in the market, the demand for video related applications is growing faster than the growing capacity of the channel. Also, the demand for videos with higher frame rate like 50/60 fps or even 100/120 fps, bigger color gamut and higher bit-depths requires further development in the video codecs with compression efficiency

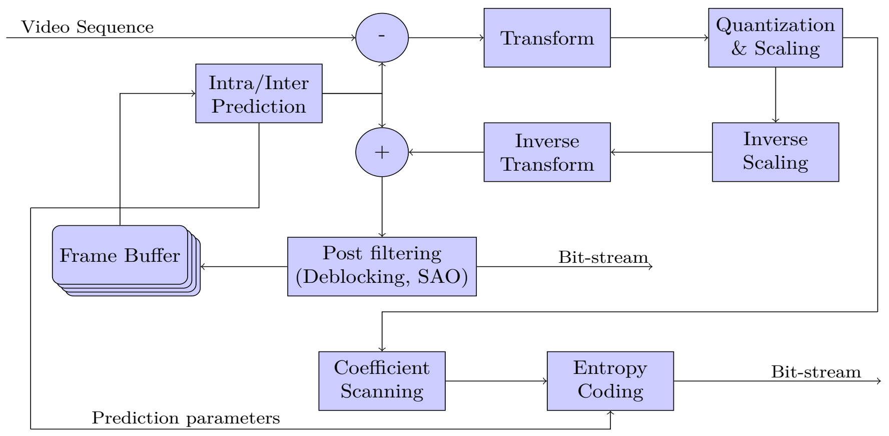


Figure 1.1: Typical compression engine of a hybrid block based codec

even higher than HEVC. Moreover, the shift of the video market from live broadcasting to video-on-demand or over-the-top (OTT) encourages an acceleration of the standardization process as new standards are more easily deployed in OTT infrastructure compared to rigid broadcasting ecosystems.

The efforts have already begun as the ITU-T VCEG and ISO/IEC MPEG had set up a Joint Video Exploration Team (JVET) for exploring the potential of video compression beyond HEVC and their first meeting was held in October 2015. Many new coding tools were added on top of HEVC which, in combination, showed a compression gain of more than 25% over HEVC but at the cost of significant encoder complexity addition. Since then, efforts are made to achieve higher gains but keeping the complexity under some limits.

Among various parts of a video codec, a significant interest has been shown in improving the transform coding in HEVC. This is mainly because of the sub-optimality of the Discrete Cosine Transform (DCT) which has been known in the literature since long time. From H.264/AVC to HEVC, the transform coding has not been changed much. The work carried in this thesis draws motivation from the fact that the DCT alone is not sufficient to model all the variations of the residuals present inside a video signal. This thesis proposes to use multiple transforms to achieve higher compression gains.

1.2 Objective and Scope of the Research

Most of codecs (MPEG-2, H.264, HEVC) have employed the integer based Discrete Cosine Transform (DCT) as the dominant block transform. The DCT is mostly used because it approximates the Karhunen-Loève transform (KLT) for Gaussian-Markov fields which are supposed to be a decent approximation of natural residual images. Another important reason behind the high affinity towards the DCT is that it can be efficiently implemented on hardware [19].

The main leverage that this research work will try to use, is based on the observation that DCT is not efficient for all types of signals. Regions with high amount of variations are usually difficult to predict and hence contains more pixel-domain energy in the predicted residuals. It is observed that there exist methods of learning new transforms that are capable of compacting the energy more than DCT. The work carried in [20] is one example of such methods.

The Karhunen-Loève Transform (KLT) is considered to be optimal for transform coding of stationary source [21]. Since the residual statistics are non-stationary, finding a single KLT which is capable of de-correlating the signal globally on an image is not possible. The KLT needs to be locally adapted to the source statistics. This is practically difficult as it requires the transmission of side information in the form of KLT eigen vectors and can be prohibitively expensive. Even KLT adapted to the local variation can be sub-optimal for transform coding of the whole image because it fails to cater the inter-block redundancies which remains [22].

This thesis focuses on designing an efficient set of transforms that is tested in competition with the existing HEVC core transforms (DCT/DST) to achieve higher compression efficiency. The research work presented in this thesis can be categorized into three main areas. First, transform learning techniques are explored under two flavors: online and offline. Second, new methods for efficient signaling of the side information in the bit-stream are presented in the context of multiple transforms. Third, methods on efficient selection of a transform candidate from a dictionary of transforms are investigated.

The main objective of this thesis is to significantly improve the coding efficiency of the HEVC while keeping the encoding complexity, decoding complexity and memory footprint as low as possible.

1.3 Outline

This thesis is composed of four parts: an introduction comprising chapter 2 and chapter 3, followed by first contribution detailed in chapter 4, 5 and 6 of Part II, followed by second and third contribution detailed in chapter 7 and 8 of Part III. Part IV concludes the thesis.

Part I
Chapter 2 covers the fundamental concepts and principles of video compression, focusing mainly on aspects that are relevant for the topic of this thesis.

Chapter 3 provides a detailed literature review on different advanced transform methods that have been proposed so far. The chapter is divided into four sections. The first section reviews the latest developments in the transform coding in context of future foreseen standard currently developed by JVET. The following three sections cover three different approaches in transform design, namely systematic known transforms, offline learned transforms, and online learned transforms.

Part II
Chapter 4 covers the analysis of the training process of a typical data-driven transform learning scheme. Different classification and transform optimization methods are discussed in terms of their mathematical formulations. Further, the existing transform learning methods are compared with each other on the basis of their training process (learning), signaling and selection inside a video codec.

Chapter 5 presents the first contribution of this thesis under the proposed online content-adaptive transform learning scheme that learns on-the-fly a set of transforms per frame and then use the learned transforms to encode the whole sequence. The performance of the proposed online learning scheme is compared to the existing offline learning scheme in terms of coding efficiency, complexity and memory requirements.

Chapter 6 is an extension of chapter 5 and presents several contributions to 1) improve the signaling of transform index, and 2) improve basis vector coding in the context of online learning scheme. It is shown that these contributions provide further gain over the existing online and offline methods on multiple transforms.

Part III
Chapter 7 shifts the focus from online learning to the offline learning methods and presents the second major contribution of this work. The first part of the chapter revisits the existing mode dependent transform learning schemes. After observing the limitations of the existing techniques, an improved mode dependent transform competition (IMDTC) scheme is proposed to achieve higher compression gains with much lower complexity. Moreover, a low-complexity IMDTC is proposed to further reduce the encoding/decoding time and memory requirements.

Chapter 8 then focuses on how to improve the content-adaptability using offline transform learning methods. A pool-based transform learning scheme is proposed and it is shown that it is beneficial to have multiple sets of transforms, also referred as pool of transform sets, in contrast to a single transform set. The generation of pools is investigated and effect of using different number of pools is shown on the final coding gain. This contribution shows that there is a potential to achieve higher gains when using several transform sets.

Part IV
Chapter 9 concludes the thesis with a discussion on the achievements made in this thesis. We summarize and analyze what we have learned. Finally, a brief perspective on the future work is provided.

1.4 Summary of The Contributions

Following is the list of contributions that are presented in this thesis.

– Detailed mathematical analysis of various data-driven transform learning schemes.
– Proposed an online content-adaptive transform learning scheme where a set of transforms are learned on-the-fly on a specific content by iterating over a single or a group of picture belonging to a specific content.
– Detailed comparison of the proposed online learning with the existing offline methods in terms of coding gains, complexity and memory requirement is carried in this work.
– In the online learning frame-work, methods to efficiently transmit the learned basis vectors to the decoder are proposed.
– In multiple transform learning framework, improved coefficient coding for adaptive multiple transforms are proposed to stabilize the learning scheme and to achieve higher compression gains.
– In multiple transform learning framework, transform index prediction method is proposed to improve the coding gains.
– Revisited the existing Mode Dependent Transform Competition (MDTC) scheme and extended it by introducing Improved MDTC (IMDTC) and low-complex IMDTC scheme

which provides significant gains over HEVC but at a considerably lower complexity compared to the MDTC scheme.

– A pool-based approach is proposed to provide higher content-adaptability using offline learned transforms.

I

Prior art on transform-based video compression

2

Video Compression, HEVC and Transform-based Coding

The goal of this chapter is to introduce the reader to the terminologies related to video compression in general, to the latest HEVC standard and its associated transform coding. The readers who are familiar with the terminologies in above topics may skip this chapter. First, the basic building blocks of a typical video compression system are briefly described. Then follows, a brief introduction to the latest HEVC video coding standard. Next, this chapter will cover some basics on transform coding which is an important part of a video compression system. Finally, some of the short-comings of the transform coding part in HEVC are discussed.

2.1 Video Compression: Basic Building Blocks

The basic goal of any video compression system is to represent a video signal in the most compact representation at an acceptable perceptive quality. The compression efficiency of a system is generally defined by the compression ratio which is defined as the ratio between the bit rate necessary for the storage of the uncompressed source and its compressed counterpart. For example, for a digital TV with SD resolution of 720×576, the uncompressed source requires nearly 165 Mbit/s, and the rate after compression is typically 4 Mbit/s. Therefore, the compression ratio is 165:4 = 41.25 [2]. For digital broadcast applications, the compression ratio is even higher than 100. Figure 2.1 shows the basic building blocks of video compression systems that are used to compress and de-compress video signals.

A video signal is **first** fed to a signal analysis step whose goal is to de-correlate the signal and exploite both spatial and temporal redundancies present in the video signal. Several linear and/or non-linear signal models may be for this purpose. Usually, linear prediction models and separable block transforms are used for de-correlating the signal. On the decoder side, the analysis block is replaced by the synthesis block which is used to reconstruct the signal by using inverse transforms and adding the prediction. The **second step** of encoding is quantization

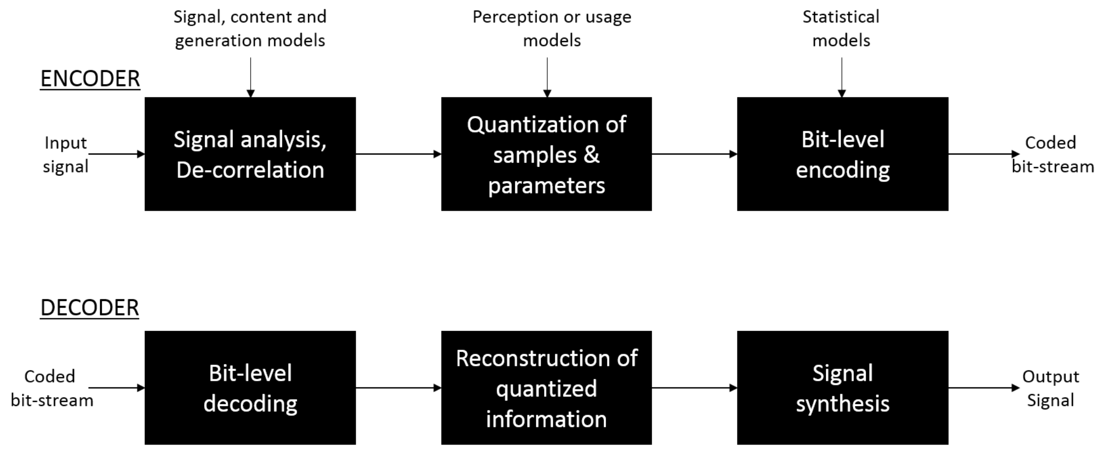

ENCODER
Signal, content and generation models
Perception or usage models
Statistical models
Input signal
Signal analysis, De-correlation
Quantization of samples & parameters
Bit-level encoding
Coded bit-stream
DECODER
Coded bit-stream
Bit-level decoding
Reconstruction of quantized information
Signal synthesis
Output Signal

Figure 2.1: Basic Building Blocks of a Video Compression System [2]

where distortion is introduced in order to achieve efficient compression. Perceptual models may also be employed to exploit the fact that human eyes are not able to perceive all the artifacts and more distortion can be accepted at certain frequencies which are less perceived by the human eye. The **final step** is a bit-level encoding which represents the discrete set of quantized values by the lowest possible bits using pure statistical models. These three blocks are the essence of any multimedia coding system.

2.2 High Efficiency Video Compression Standard

High Efficiency Video Coding or formally known as ITU-T H.265 / ISO/IEC 23008-2 is, to date, the latest standard developed by the Joint Collaboration Team on Video Coding (JCTVC) which is a team of experts from ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) [23]. The first version of HEVC standard was released in January, 2013. Initially, the new standard aimed at achieving 50 percent bit-rate saving for high quality video coding as compared to the state-of-the-art Advanced Video Coding (AVC) standard which accounts for more than 80 percent of all web videos. This goal has been reached particularly on high definition videos as shown in [16]. Thus, the HEVC codec is expected to ease the burden on global networks where HD and UHD video content is becoming more and more popular. HEVC is based on the basic hybrid structure as employed by previous standards since H.261. However, the standard contains a series of incremental improvements [24] over h.264 in order to achieve a better compression such as

1. flexible Block partitioning using a quad-tree structure,
2. enhanced Intra Prediction modes,
3. greater flexibility in Transform block sizes,
4. more sophisticated interpolation in Motion compensation and de-blocking filters,
5. adaptive coefficient scanning order and improved coefficient coding,
6. addition of parallel processing capabilities, and
7. new features like SAO, GOP structures, etc.

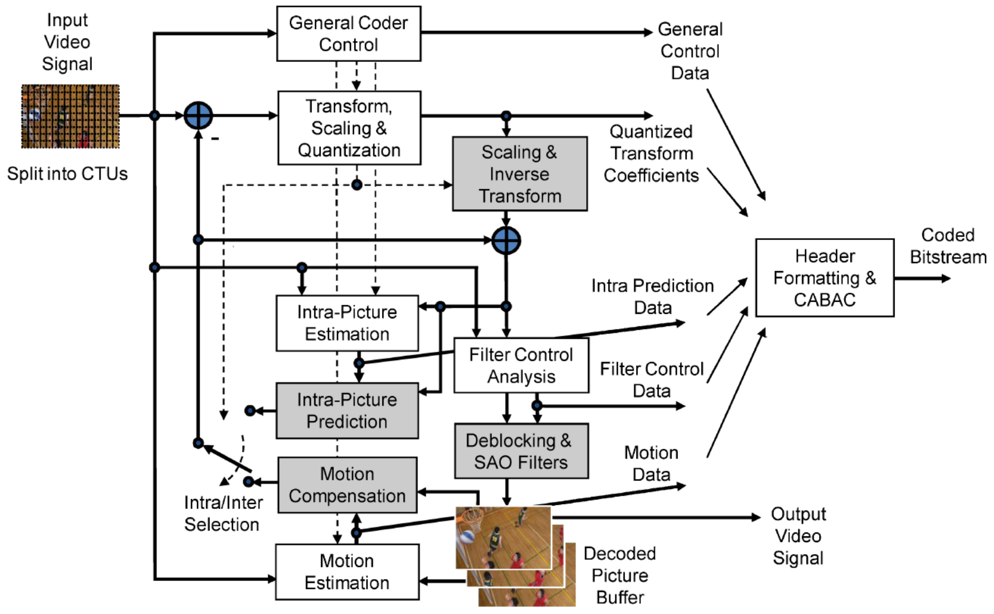


Figure 2.2: Basic Block Diagram of a HEVC Encoder [3]

Some of the above mentioned improvements are described hereafter in order to help the reader understand the proposed contributions that are presented in later chapters of this thesis.

2.2.1 Sketch of the codec

HEVC maintains the hybrid coding structure similar to its predecessors. The block diagram in Figure 2.2 shows the basic structure of a HEVC encoder that outputs a bit-stream conforming to the HEVC standard. Each picture is split into block-shaped regions, with the exact block partitioning conveyed to the decoder by using a Quad-Tree (QT) structure. The first picture in a video sequence is coded using intra-prediction as the first step to remove the spatial correlation present in the picture. This is done using the previously coded samples from the neighboring blocks of the same picture. The remaining pictures in the sequence utilize previously coded pictures to predict the blocks in the current picture. The reference picture index and the corresponding motion vectors are transmitted in the bit-stream.

The residual between the original and predicted picture is transformed in order to remove most of the pixel correlation and compact the signal energy in the first few transformed coefficients. These coefficients are quantized based on the Quantization Parameter (QP) value set at the encoder by the user. The quantized coefficients along with side information (including general control data, QT, intra prediction data, filter control data, motion data etc.) are finally coded using the Context Adaptive Binary Arithmetic Coding (CABAC) encoder which outputs an HEVC compatible bit-stream [25].

The decoder parses the bit-stream and performs inverse operations in order to reconstruct the decoded signal. The blocks common to both encoder and decoder are shaded in gray in Figure

2.2.

2.2.2 Quad-tree structure in HEVC

HEVC, in contrast to previous standards, employs a flexible partitioning of a picture in a video sequence. Each coded video frame is partitioned into tiles and/or slices, which are further split into Coding Tree Units (CTU). The CTU is the most basic unit of coding similar to Macroblock as used in the previous AVC standard. Partitioning of CTUs is based on a quad-tree approach in order to be split further into Coding Units (CU). The size of these CUs can vary from 64×64 to 8×8 [24]. Figure 2.3 depicts this structure.

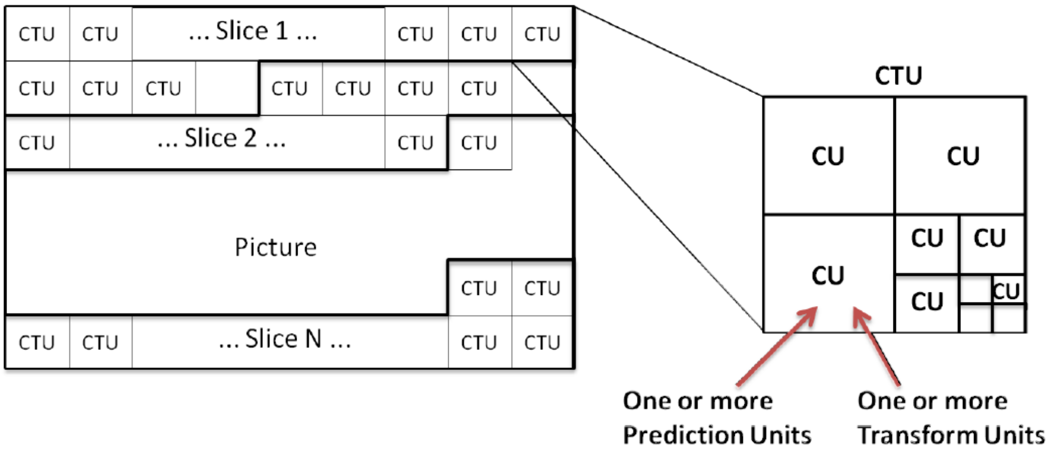


Figure 2.3: Picture, Slice, Coding Tree Unit (CTU), Coding Units (CUs) in HEVC [3]

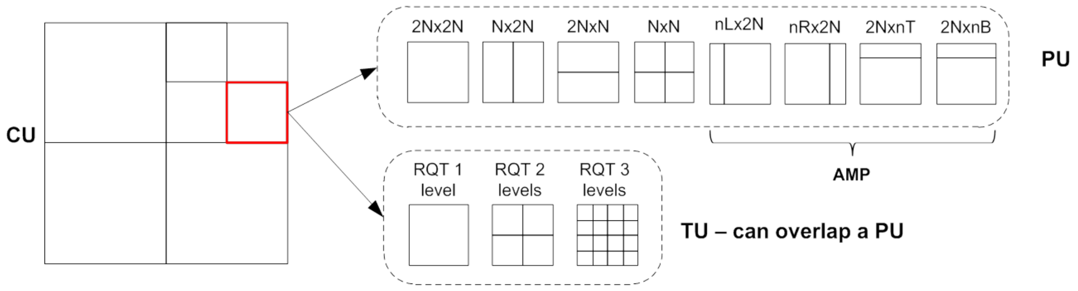


Figure 2.4: CU partitioning into Prediction Units (PUs) and Transform Units (TUs) [3]

Also, HEVC distinguishes between the blocks and units. A block is an array of only luma or only chroma samples whereas a unit is a structure that includes all the encoded color components (luma and chroma blocks) as well as the associated syntax elements and prediction data that is attached to the blocks (*e.g.*, motion vectors). For example, a CTU encapsulates the Coding Tree Blocks (CTB) of the color components and associated coding tree syntax structures. Similarly, a CU encapsulates the Coding Blocks (CB) of all the color components and the associated coding syntax structures.

Each CU is partitioned into one or more Prediction Units (PU) depending on the prediction mode (intra or inter prediction as explained in the next sub-section) selected at the CU level. Additionally, a CU is also partitioned into one or more Transform Units (TU) in a quad-tree fashion, and the associated quad-tree is named as residual quad-tree (RQT). Depending on the size of a PU, TU size may vary from 32×32 to 4×4 [25]. In general, the RQT is restricted to the depth of 2. For instance, a 32×32 CU cannot be split in TU smaller than 8×8. Figure 2.4 illustrates the splitting of CUs into PUs and TUs, where AMP denotes Asymmetric Motion Partitioning.

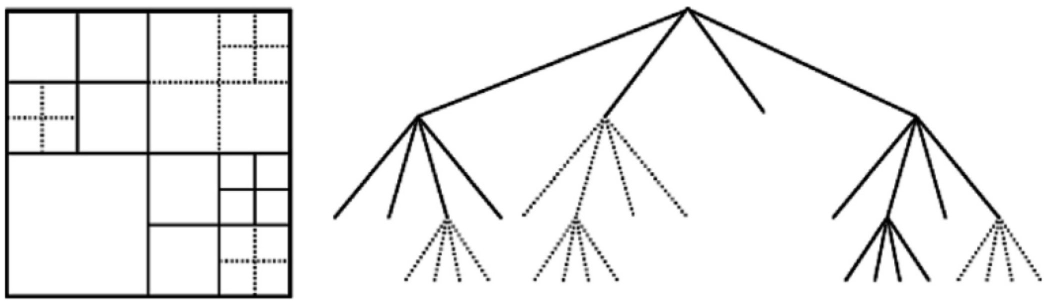


Figure 2.5: Subdivision of CTBs into CBs and TBs, with TB boundaries indicated by dotted lines [3]

Figure 2.5 shows the two quad-trees (QT) that represent the various possible CBs and TBs within a CTB. The QT is represented with hard lines and the RQT is represented with dotted lines. Usually, the chroma blocks are split in the same way as the luma blocks. However, the size of the chroma block depends on the chroma sub-sampling ratio employed in the video encoding scheme.

Chroma sub-sampling is used in video encoding systems to encode the chroma information at lower resolution compared to luma information. This is because the human visual system is more sensitive to the variations in brightness than color. For the 4:4:4 chroma sub-sampling ratio, the chroma is not sub-sampled and therefore, chroma block size remains same as luma block size. However, for the 4:2:0 chroma sub-sampling ratio, the chroma block is sub-sampled to half in both horizontal and vertical direction. Therefore, a block of size 16×16 consist of a luma sample of size 16×16 and two chroma samples of size 8×8.

2.2.3 Intra and Inter Prediction

In HEVC, as well as its predecessor AVC, two prediction modes, namely intra prediction and inter prediction modes, are employed. Intra prediction of the current block is done using the previously decoded boundary samples of the neighboring blocks of the same picture. HEVC employs 33 directional modes for intra prediction as shown in Figure 2.6 (left). Alternatively, planar prediction and DC prediction are also used. This is much more compared to AVC which employs only 8 directional intra modes and DC intra prediction for blocks less than 16×16 in size. Figure 2.6 (right) illustrates the intra prediction of the current PU from the boundary

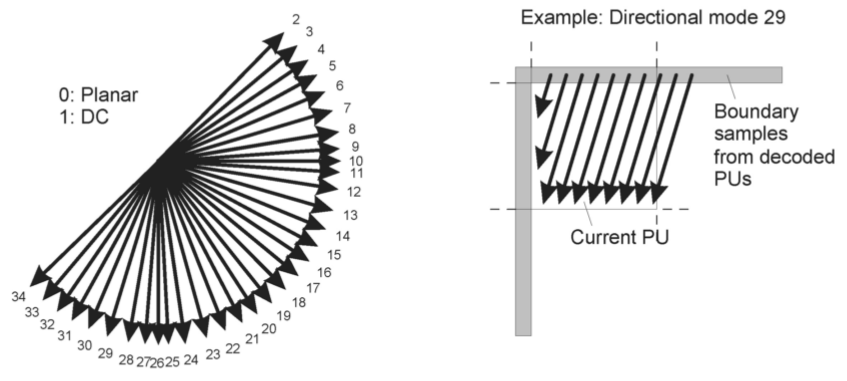


Figure 2.6: Modes and directional orientations for intrapicture prediction [3]

samples of the previously decoded PUs for the direction mode 29.

For intra prediction of the current block, the neighboring reference sample values are required depending on the prediction mode. At most 4N + 1 neighboring pixels are required for predicted a block of size N×N. Sample Padding is done in case of unavailability of some of the neighboring pixel values.

In case of inter prediction, the current block is predicted from the co-located blocks in the previously encoded pictures. Usually, a motion estimation process is employed by the encoder to determine the position of the suitable block in the previous picture or pictures (also referred as reference picture(s)). The motion vectors, indicating the 2-D offset of the block with respect to its reference block, are determined and are coded in the bit-stream. In HEVC, advanced motion vector prediction (AMVP) is proposed to improve the computation and coding of the motion vectors. Moreover, the design of interpolation filters in HEVC is also improved.

Finally, the prediction block computed using one of the above methods is pixel-wise subtracted from the original block to generate a residual block. This residual block is then fed to the residual coding part for further processing.

2.2.4 Transforms in HEVC

HEVC employs a 2-D Discrete Cosine Transform (DCT) similar to the previous standard in order to code the intra or inter predicted residual signal. HEVC, however, supports different transform block sizes of 4×4, 8×8, 16×16 and 32×32. These transforms are obtained from the integer approximated separable and scaled DCT basis functions. The design of these integer approximations is carefully carried as discussed in [26].

Apart from the HEVC core transforms, an alternative integer transform derived from the 2-D Discrete Sine Transform (DST) is applied to the luma intra-predicted 4×4 residual blocks. This is done to better fit the statistical property of such residual blocks, as explained in [3].

2.2.5 Quantization in HEVC

The coefficient values obtained after the transform step are scaled and quantized to predefined levels using a scalar quantizer in order to obtain the lossy compression. There is a certain advantage of transforming the image/video signal into coefficients before the quantization is applied. By carefully applying the quantization in the transformed domain, the subjective quality of the images and videos is preserved better than applying quantization on the original residual signal.

In HEVC and its predecessor AVC, the coefficients are quantized using a scalar quantization with a dead zone. Figure 2.7 depicts the scalar quantizer on the left and a dead zone quantizer with a dead zone of 3 steps. A dead zone quantizer zeroes out the coefficient values within the "dead zone" interval (see Figure 2.7) while the other intervals are equally spaced. The design of the dead zone is based on the statistics of the transformed coefficients. Moreover, the step-size of the quantization interval is controlled by the Quantization Parameter (QP). Smaller values of QP corresponds to finer quantization intervals following the formula

$$\Delta X = \Delta X_0(\text{QP} \bmod 6) \cdot 2^{\lfloor \frac{QP}{6} \rfloor}$$

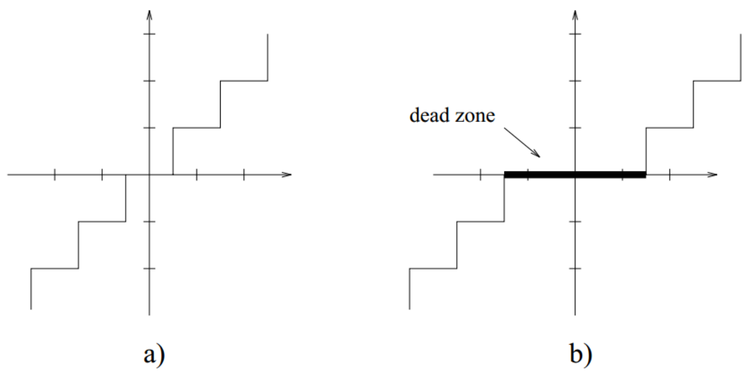


Figure 2.7: a) A Scalar Quantizer b) A Quantizer with a dead zone of length 3 [3]

2.2.6 Adaptive Coefficient Scanning and Coefficient Encoding

After quantization, the quantized coefficient block is fed to the coefficient encoding step. The number of non-zero quantized coefficients in a block are often small or even zero at low bit-rates. Therefore, coefficient encoding is designed to efficiently encode them and to also allow a high throughput rate as discussed in [15].

A hierarchical approach is employed for coding the transform coefficients inside a transform block (TB). At the first level, the coded block flag (cbf) is signaled for each TB to specify the significance of the entire TB. If the cbf flag is 1, the TB greater than 4×4 is organized into 4×4 size coding groups (CGs). Each of the CGs is associated with a flag indicating the existence of at least one significant coefficient in that particular CG. Figure 2.8 shows splitting of a 16×16

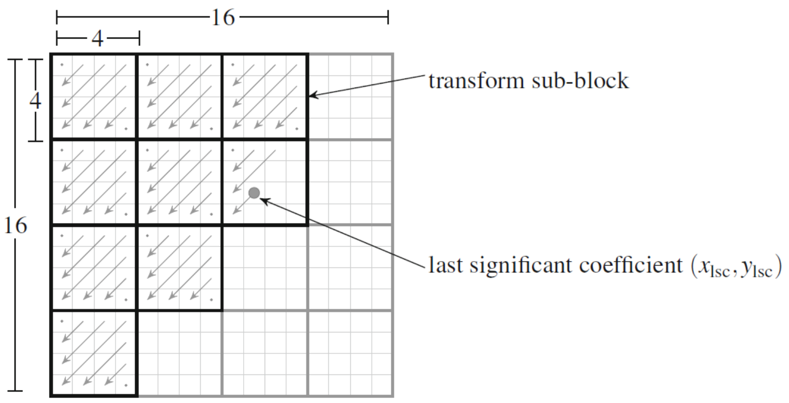

16
4
4
16
transform sub-block
last significant coefficient $(x_{\text{lsc}}, y_{\text{lsc}})$

Figure 2.8: Example sub-block scanning of a 16x16 transform block [4]

transform block into 4×4 CG.

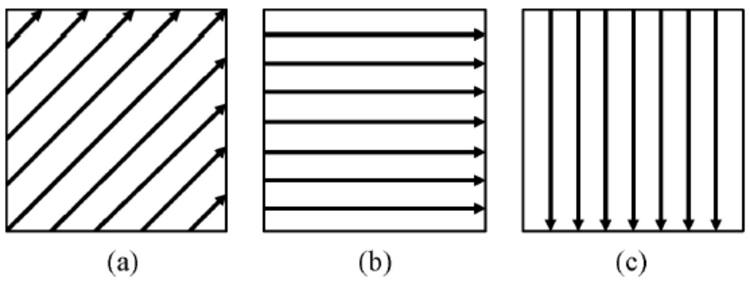


Figure 2.9: Different Coefficient Scanning methods in HEVC [3]

Next, coefficient scanning is used to convert the 2-D coding group to a 1-D array of values. HEVC incorporates an adaptive coefficient scanning method as compared to a single coefficient scanning mode in AVC. Three coefficient scanning methods, namely diagonal up-right, horizontal and vertical are employed at the block level as shown in Figure 2.9 (a), (b) and (c) respectively. The choice of the scanning method is done based on the intra-picture prediction direction. For example, the horizontal scan is used when the prediction mode is near to vertical, and vice-versa. For other prediction directions and inter prediction case, the diagonal up-right scan is used [4]. The goal is to scan from larger coefficient values to smaller coefficient values so that it assists the entropy coding scheme employed in HEVC.

Concerning the coding of the CGs, up to five scan passes are made on a single CG as follows:

- Significance Map - Indicates the position of the non-zero coefficients inside a CG.
- Level Greater than 1 - For the first eight significant coefficients, a flag is set if the level of the coefficients at a position is greater than 1.
- Level Greater than 2 - A flag indicates the first coefficient that is greater than 2. The flag

is not sent for any subsequent coefficients.

– Coefficient Sign - sign information of significant coefficients
– Remaining Absolute Level - the remaining values that are not coded in the above passes

Additionally, a mechanism of sign data hiding based on the parity sum of coefficient is applied to save bit-rate. The remaining absolute level is binarized using a Exponential-Golomb code. Finally, the binary stream of the CG is fed to the CABAC to obtain a compressed stream.

2.2.7 CABAC

Context-Based Adaptive Binary Arithmetic Coding (CABAC) is designed for a lossless compression of the syntax elements that need to be transmitted to the decoder. Figure 2.10 shows the basic building blocks of a CABAC encoder.

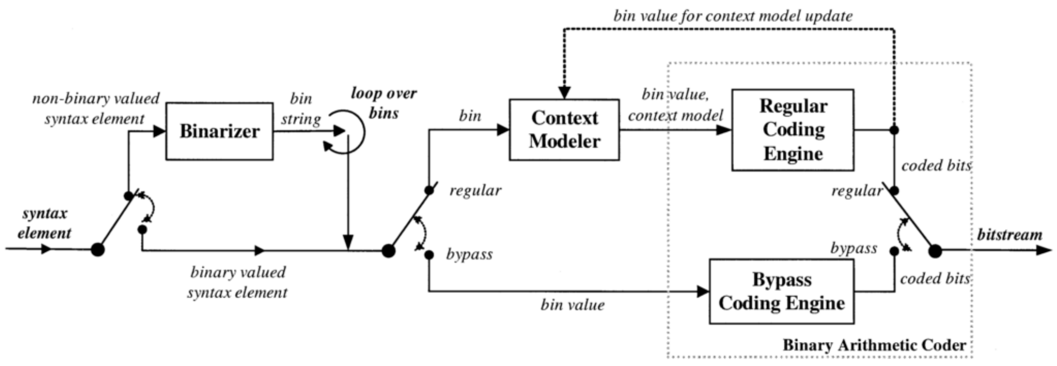


Figure 2.10: Block Diagram of the CABAC encoder [5]

A CABAC operation can be divided into three main parts:

– Binarization - CABAC requires the non-binary values to be transformed to the binary values, also known as bins. This is done to utilize a binary arithmetic coding engine which is computationally less complex compared to a m-ary arithmetic coder. This step is by-passed in case the input string is already a binary string.
– Context Modeling - Each bin can be attached to a specific probability model based on the previous encoded bins or symbols. The context modeling may be by-passed and in this case, the probability of occurrence of binary values 0 and 1 is equal to 50%. A context model may be used to determine the current bit based on the past samples.
– Binary Arithmetic Coding Engine - The two symbols (0,1) are characterized by the Most Probable Symbol (MPS) and Least Probable Symbol (LPS) based on the probability assigned by the context model. This probability is updated after each bin is encoded using the binary arithmetic coder. Each bin is specified by the probability estimate of the LBS ($\tilde{p}_{LBS}$) and the value of the MPS ($v \in \{0, 1\}$).

In contrast to AVC, several new context models have been introduced in HEVC for different syntax elements in order to better utilize the statistical correlation present among neighboring blocks.

2.2.8 Special Modes in HEVC

Apart from the regular coding modes which involves prediction followed by block transformation and quantization of the residual signal, HEVC employs the following three special modes:

Transform Skip

In this mode, the transform is by-passed. In HEVC, this mode is applied only on TBs of size 4×4 only. Moreover, it is proved in literature that having transform skip as one of the coding option provides better compression for screen-content sequences where the residual features may depict strong discontinuities and application of DCT may introduce ringing artifacts.

Transform and Quantization Bypass

Here, the predicted residual block is directly fed to the encoder in order to generate the bit-stream. This mode is also referred as a lossless mode because in addition to transform and quantization, the other processing such as SAO and deblocking filters are also by-passed. This mode is useful for loss-less coding.

PCM Coding

In PCM coding mode, no prediction, no transform and no quantization is applied to the signal inside a coding block and the samples are directly encoded into the bit-stream. This is also equivalent to a lossless encoding as it depicts the upper-bound of the bits required for encoding a block. Therefore, PCM is usually employed where the usual prediction and residual encoding cannot be handled properly and bits required to encode the block exceed the upper-limit (example, for highly noisy content).

2.2.9 Encoder Control

The encoder control is employed to ensure that the generated bit-stream conforms to the requirements imposed by the video coding specifications. Encoder control is configured to either provide a minimum bit-rate for a pre-defined reconstruction quality or provide a maximum reconstruction quality for a given bit-budget. The encoder control takes various encoder decisions such as choosing different modes and partitioning at the encoder. These decisions may be made locally at a block or sub-block level or may be made at a picture level. For example, a decision to choose intra or inter prediction is made at a CU block level.

During the encoding, several modes are tested in brute-force fashion and the decision to choose the best mode is generally taken based on a rate-distortion criteria which maintains the balance between the number of bits spent versus the distortion introduced in the reconstructed picture. This is known as the Rate Distortion Optimization (RDO) process. The RDO process depends on how the rate and the distortion are measured in the codec.

Rate may simply be calculated by counting the number of bits required to code a block in a given mode. A complete block along with all the syntax elements is coded to determine the rate which is used for the RDO decision. In some cases, only the bits required to encode the syntax elements is used as a criteria for making decision. Usually, an entropy coder such as CABAC is used to encode the block in order to determine the number of bits. For real time encoders, the entropy coders are often replaced by rate-models [27] which are easily implemented using look-up tables.

For measuring the distortion, the mean square error (MSE) is usually used, but in some professional encoders, this can be replaced by a human visual system oriented distortion metric in order to generate videos with better perceptual quality. In HEVC, the following distortion measures are taken into account for making the decision.

2.2.9.1 Sum of Squared Difference (SSD)

SSD is simply calculated as the sum of the square of pixel-wise difference between the two blocks that are being compared. Let O be the original block of size N×N, and R be the reconstructed block of the same size as O. Then SSD is calculated as:

$$D_{SSD} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} |o(n,m) - r(n,m)|^2$$

2.2.9.2 Sum of Absolute Difference (SAD)

In some cases, SAD is computed in place of SSD as it is faster to compute. It is similar to SSD except that the square term of omitted. Therefore, it is calculated as:

$$D_{SAD} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} |o(n,m) - r(n,m)|$$

2.2.9.3 Sum of Absolute Transformed Difference (SATD)

SATD is calculated by summing the transform of the absolute difference of the pixel values. The Hadamard transform is used for this purpose. The transform of residual provides a good estimate of the coding cost of the residual block in the subsequent transform coding stage. This metric is sometimes used during the analysis step in order to reduce the number of possible modes. For example, in case of intra prediction modes, 3 best modes out of 35 modes are chosen at the analysis step. SATD is used as a distortion metric in this case.

SATD is calculated as follows:

$$L = H \cdot (O - R) \cdot H^T$$

$$D_{SATD} = \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} |l(n,m)|$$

2.2.9.4 Peak Signal-to-Noise Ratio (PSNR)

PSNR is the most commonly used objective metric to measure the distortion of a video signal and it is computed as the ratio of the maximum possible amplitude of the original signal and the mean square error (MSE) between the original and the reconstructed signal. Also, the PSNR is computed independently for the luma and the two chroma components. It is measured in logarithmic in dB scale as follows:

$$PSNR = 10 \cdot \log_{10}(\frac{A_{\max}^2}{MSE})$$

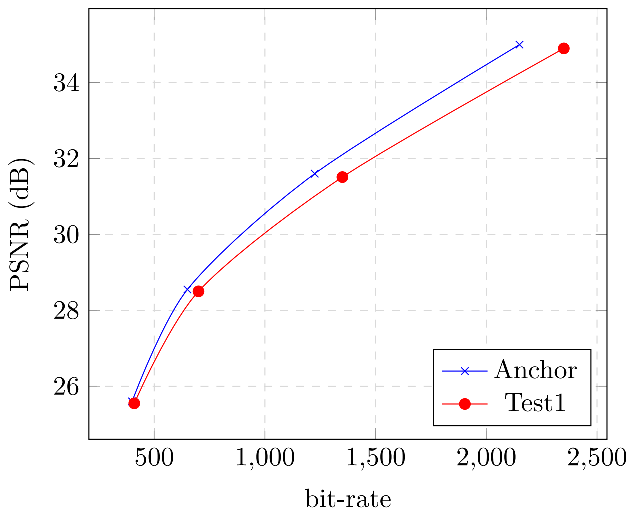


Figure 2.11: Example of a BD Plot [6].

with $A_{\max} = 255 = 2^8 - 1$ for a 8-bit signal, and $A_{\max} = 1023$ for a 10-bit signal.

2.2.9.5 Rate Distortion Optimization

The Rate distortion optimization (RDO) process is important to improve the balance between distortion and bit-rate of a lossy coded video sequence. It is possible to achieve less bit rate at the cost of higher distortion by changing the value of the Quantization Parameter (QP) that determines the target quality of a video sequence. In the rate distortion optimization process, the encoder is presented with different coding choices such as different prediction modes, transform block sizes etc. in order to encode the current block at a minimum cost. The RD optimization problem using the Lagrange function J is given by:

$$\text{minimize} J \text{where}, J = D + \lambda R$$

where D denotes the amount of distortion caused by selecting a particular mode, R denotes the number of bits required to encode a block with the selected mode and the “lagrange parameter” λ is deduced from the quantization parameter (QP). A high value of λ leads to greater penalization of the rate term and, therefore, the video quality is sacrificed to maintain lower rate. Similarly, a low value of λ favors distortion term and, therefore, leads to high bit-rate and low distortion.

2.2.10 Bjøntegaard Delta (BD) Rates

Bjøntegaard Delta (BD) plots are used for analyzing the performance in terms of the rate distortion (RD) coding gains using curves. It represents the average PSNR or average bit-rate difference between the RD curve of an anchor and the tested video sequence. These plots provide a good idea of the rate and/or quality gain with respect to the reference. The curves are found by a polynomial interpolation between at least four simulated points per curve.

As seen in the Figure 2.11, the PSNR versus rate data points are plotted and a curve using a third order polynomial is drawn through these four points. The difference between the curves is dominated by high bit rates. Hence sometimes, a logarithmic value of bit-rate is considered for calculating BD Rate as described in [6].

2.3 Block Transforms in general

Transforms are the key element in any block-based video coding system which are used for de-correlating the residual signal by exploiting the statistical dependencies in the pixel domain. A well-designed transform is able to efficiently represent the signal in the transform domain with 1) less number of non-zero coefficients, 2) less amplitude, 3) decreasing sequence of coefficients, thus leading to compression. The transforms are applied at a block level instead of a complete image. This leads to better adaptation of the local characteristics of the image. Thus, these transforms are often denoted as block transforms.

2.3.1 Properties of block transforms

Some properties of the block transforms, that are desired in video compression systems, are described below:

Separability

By definition, a separable transform is a cascade of a horizontal and a vertical 1-D transform.

$$T_{2D}(x,y) = T_{\text{hor}}(x)T_{\text{ver}}(y)$$

Separable transforms are computationally simpler and there exist fast and hardware friendly implementations. However, these transforms can only utilize correlation either within a column or within a row. They fail to model well the directional structures present in the signal. For example, if a residual block has a diagonal edge, the separable transforms cannot efficiently model its structure.

On the other hand, the non-separable transforms are able to adapt better to a given direction for which they are designed. In case of non-separable transforms, the block x of size N×N to be transformed is lexicographically ordered into a 1-D vector of size $\text{N}^2 \times 1$ and is matrix multiplied to the transform T of size $\text{N}^2 \times \text{N}^2$ to generate the coefficients c such that

$$c = T^T \cdot x$$

Therefore, a non-separable transform requires N^4 multiplications to compute the coefficients. Also, higher memory is required to store the non-separable transforms.

Reversibility

Transforms are said to be reversible if a signal can be recovered from its transform domain by simply applying an inverse transform. This is an essential requirement in a video coding application where the original image/video signal needs to be recovered at the decoder with high fidelity. Non-reversible transforms are therefore not suitable for these applications.

Orthogonality

An orthogonal transforms defines a set of orthogonal basis vectors. Given an orthogonal matrix $\mathbf{A}$ with its rows as the basis vectors, these rows will be orthogonal to each other. Moreover, the transform is said to be orthonormal if the row has a length of 1, i.e. $\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I}$. The advantage of using orthogonal transforms in compression comes from its property that the inverse of transforms A is given by its transpose i.e.

$$\mathbf{A}^{-1} = \mathbf{A}^T$$

On the other hand, a non-orthogonal transform does not possess the above property. However, it may possess higher energy compaction efficiency when compared to an orthogonal transform. This is illustrated in a paper by Onur G. [28]. The drawback of using a non-orthogonal transforms in video coding application is that they require more memory to store the non-orthogonal transform and its inverse at both encoder and decoder. Moreover, these transforms are not well suited for block based approaches that are currently used in the state-of-the-art codec.

Data Dependency

A transform can be categorized into a data-independent or data-dependent transform. The data-independent transforms do not depend on the input signal and are able to provide good compression efficiency over a large variety of signals. Data-dependent transforms are learned on an input signal and therefore, provide even better energy compaction. However, the complexity of computing the basis vectors for each signal type and sending the basis vectors to the decoder is not a trivial task. Consequently, data-independent transforms are mostly used up to now.

In this thesis, the practicality of data-dependent transforms in contrast to data-independent transforms is studied.

2.3.2 The Most Popular Transforms

Some of the most popular transforms that possess the above desired properties and have been extensively used in video coding are described below:

The Karhunen Loève Transform (KLT)

The Karhunen Loève Transform is an orthogonal transform that de-correlates the signal and has the highest energy compaction efficiency for stationary sources. The KLT of a stationary source can be interpreted as a Principal Component Analysis (PCA). It is found by the eigenvector decomposition of the covariance matrix, often obtained by SVD.

The autocorrelation matrix $\mathbf{R}_{xx}$ of an input vector signal x is defined as:

$$\mathbf{R}_{xx} = \mathrm{E}\langle xx^T \rangle$$

where, E is the expected value.

The KLT of $\mathbf{R}_{xx}$ is the matrix of eigenvectors v of $\mathbf{R}_{xx}$ and these vectors are always orthogonal since the correlation matrix is symmetric[15]. Therefore,

$$\mathbf{R}_{xx}v = \Lambda v$$

$$v^t \mathbf{R}_{xx} v = \Lambda$$

where, Λ is a diagonal matrix which contains the eigenvalues of $\mathbf{R}_{xx}$. Thus, this transform diagonalise the input signal covariance matrix which leads to a full de-correlation.

Due to the fact that the KLT is data-dependent transform, this transform is not used directly in the video and image related applications where the signal statistics are not known in advance. Some of the methods that utilize KLT transform for video coding application are detailed in Chapter 3.

In the paper [22], it is shown that the KLT is sub-optimal in a scalar quantized transform coding system and even in case of stationary signals, it may be sub-optimal. For example, in case of transformation of predicted residual signal, the KLT may be sub-optimal due to the correlation that exists between the residual blocks. Another drawback of KLT is that they cannot be implemented using fast algorithms.

Discrete Cosine Transforms

The discrete cosine transform (DCT) is the first and the most extensively used transform in the field of video and image coding. It was first introduced by Ahmed N. *et al.* in 1974 [29]. The author also introduced the algorithm to compute it. It has also been found that the DCT is a good approximation of the KLT in case of images and videos modeled by highly correlated Gaussian Markov fields. The orthogonal transform basis vectors T_k are defined as

$$T_k(n) = \sqrt{\frac{2}{N}} C_0 \cos\left(\frac{\pi}{N} k \left(n + \frac{1}{2}\right)\right) \tag{2.1}$$

$$C_0 = \begin{cases} \frac{1}{\sqrt{2}} & \text{for } k = 0, \\ 1 & \text{otherwise.} \end{cases}$$

where N is the length of the basis vector. It can be seen from the equation that the DCT basis vectors do not depend on the input signal (data-independent). Moreover, the DCT can be extended to two (or more) dimensions in the form of separable transforms.

Based on the boundary conditions, eight standard types of DCT can be derived from their basis vectors [30]. For video compression applications, the DCT-2 derived using equation 2.1 is the most used transform. Although due to the even-odd boundary conditions of the DCT-4, the overlapping transforms introduced by [31] was easier to develop on the DCT-4.

In recent works, other trigonometric transforms from the family of DCT are being used for video compression. These methods are described in Chapter 3.

Discrete Sine Transforms

The Discrete Sine Transform (DST) is defined by the following transform basis vectors

$$T_k(n) = \sqrt{\frac{2}{N+1}} C_0 \sin\left(\frac{\pi}{N+1} (k+1)(n+1)\right) \tag{2.2}$$

The DST was introduced in HEVC as an alternate to the DCT for 4×4 intra-luma prediction residual block. It fits the statistics better if the residual signal amplitude increases as function

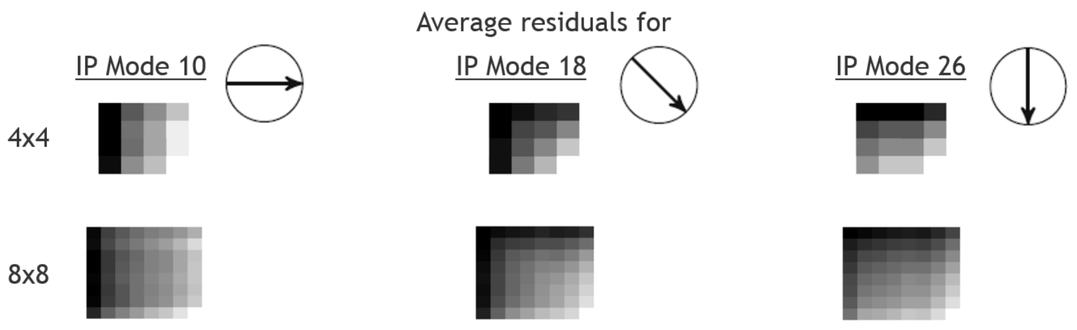


Figure 2.12: Average residual statistics for three different prediction modes

of the distance from the boundary samples, that are used for prediction [3].

Walsh-Hadamard Transform

The Walsh-Hadamard Transform, also known as the Hadamard Transform, is a variant of the Discrete Fourier Transform. It is simple to compute and can be efficiently implemented with fast algorithms. The Hadamard transform can be constructed recursively. Let the 1x1 Hadamard transform be $\mathrm{H}_0 = 1$, and H_m, where $\mathrm{m} > 0$, is defined as:

$$H_m = \frac{1}{\sqrt{2}} \begin{pmatrix} H_{m-1} & H_{m-1} \\ H_{m-1} & -H_{m-1} \end{pmatrix}$$

The Hadamard Transform of size 4×4 and 2×2 are used in AVC if a 16×16 Intra prediction residual is transformed using 4×4 transform. The 16 DC coefficients of each 4×4 transformed block are further transformed using a secondary Hadamard Transform of size 4×4. A 2×2 Hadarmard Transform is used for the corresponding chroma block [32].

2.4 Motivation to improve the transforms used in HEVC

The DCT is the dominant block transform in video coding standards such as HEVC, AVC and MPEG-2. The DCT is a good approximation of the KLT transform for images modeled by the first order Gaussian-Markov model [33]. Additionally, fast algorithms for hardware and software implementation of the integer 2-D DCT already exist and further favor the use of DCT as the core transform.

Residual statistics are known to be slightly different from the natural image statistics as the correlation among neighboring pixels is generally removed by the prediction step carried before the transform step. Therefore, the residuals do not always possess strong correlation among neighboring blocks. However, the DCT is still good for residual blocks that have less energy or are highly correlated especially in the horizontal and vertical directions.

Moreover, residual statistics depend on the intra or inter prediction mode and also, on the direction of prediction in case of intra prediction. Figure 2.12 shows the average residual statis-

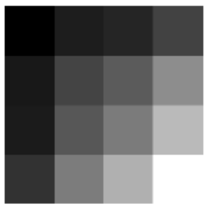


Figure 2.13: Average residual statistics for intra predicted luma residual of size 4×4

tics for three different intra-prediction modes where the arrow depicts the direction of the increase in the prediction error. Moroever, the author in [34] has shown that inter predicted residuals have more error energy scattered around the boundaries of the block which is not the case in intra predicted residuals.

In HEVC, the DCT is replaced by the DST for intra luma residual block of size 4×4. This is done mainly because the DST fits better to the average residuals statistics generated in case of intra luma 4×4 block. Figure 2.13 illustrates the average residual statistic of a 4×4 intra-predicted luma residual block for a video sequence. It is observed that the residual energy increase as one moves away from the boundary in both horizontal and vertical direction. Therefore, the DST better fits these statistics. In HEVC, using the DST instead of the DCT has brought a coding gain of 1% computed using the BD-rate metric.

Further, both the DCT and the DST have been proved to be sub-optimal for blocks with directional edges or with a high amount of texture. Figure 2.14 illustrates this with a help of an example. Two residual blocks with completely different statistics are shown and the corresponding unquantized coefficient values are presented. It is observed that the block with a strong edge is transformed into a coefficient block with dense coefficient values which are in turn expensive to encode by the entropy coder. However, for a residual signal with less energy, it can efficiently compress it to few coefficients, thus leading to less coding cost.

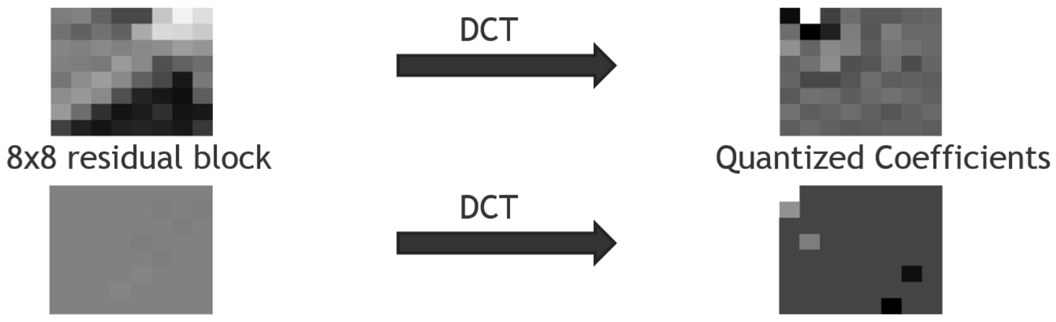


Figure 2.14: Illustration of sub-optimality of DCT for residuals with directional edges

One solution to efficiently encode blocks with directional edges or high texture is to learn a KLT by computing the eigenvector decomposition of the covariance matrix of the residual block. However, it is not practical to perform such task for each block as it is computationally demanding and also, the overhead of transmitting the basis vectors for each block is very high. Solutions that avoid transmitting the basis vectors exist in the literature where the past samples are used to generate the basis vectors at the decoder. However, these methods usually increase the decoder complexity which is not desirable.

But, it is fair to assume that the residuals with similar statistics may repeat across different regions of the same frame, within the same sequence and also, across different sequences. Therefore, transforms that are able to efficiently model different variations across different video sequences may perform better than the DCT. Many schemes have been proposed in the literature to find these **advanced transforms** that have better de-correlation properties than the DCT but are still computationally reasonable. In the next chapter, some of the most recent advanced transform schemes are presented in detail.

2.5 Conclusion

This chapter has presented the basic building blocks of a video compression system and a detailed description of the HEVC standard, which is the current state-of-the-art video coding standard, is provided. The tools in HEVC that provide higher compression gains over its predecessor AVC have been described.

Because this thesis focuses on the transform coding part of the video codec, several popular block transforms used in video codecs along with their properties have been briefly described. Finally, this chapter presents the motivation for improving the transform coding part in HEVC by describing the sub-optimality of the DCT for residual blocks with directional edges and high amount of texture.

This thesis will focus on improving the transform coding part in HEVC by proposing advanced transforms with better de-correlation capabilities.

Next chapter will detail the different advanced transform schemes that have been proposed in the literature along with their possible shortcomings.

3

Advanced Transforms

3.1 Introduction

Study of efficient transforms for image and video coding has been an active area of research for more than three decades. Considering the sub-optimality of the DCT and its integer approximation as discussed in the previous chapter, efforts were made to replace existing transforms or add new transforms in order to achieve higher compression efficiency while keeping the computational requirement under acceptable limits. The newly designed transforms are able to represent the directional edges and high texture area with less number of coefficients and thus producing a sparse representation. Many different approaches have been proposed to adapt the transforms to the video content. This has led to three main branches as shown in Figure 3.1.

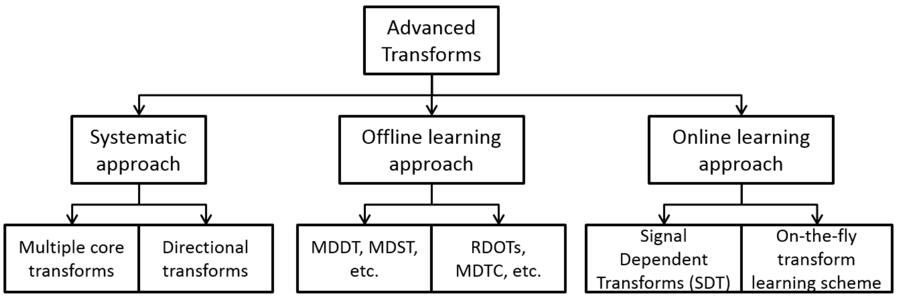


Figure 3.1: The three different approaches to obtain advanced transforms for video coding

The **first approach** involves using the systematic transforms, i.e. known fixed transforms not adapted to precise types of content, with better de-correlation of the residual signals compared to the DCT. These new transforms are either used in place of the conventional DCT or in addition to the conventional DCT. This approach is used in [8] and [10] by employing a set of directional

DCTs in order to exploit the directionality of the residuals. Zhao et. al. in [1] proposed Enhanced Multiple Transforms (EMT) that uses multiple 2D sinusoidal transforms, other than DCT, for the coding of intra and inter prediction residuals. This work has also been recently proposed in the ongoing standardization activity, led by the Joint Video Exploration Team (JVET), for the next generation video codec.

In the **second approach**, a dictionary of transforms is learned on a large offline training set which is then made available on both encoder and decoder side of a video codec. These transforms either replace the core DCT/DST transform inside the codec or are tested in competition with DCT/DST. The methods proposed under this approach fall into two categories depending on whether or not an additional syntax to indicate the choice of the transform is added to the bit stream. If not, this information is implicitly derived from the causal information available at the decoder side [12,35–37]. If yes, this information is explicitly signaled to the decoder, leading to extra bit-rate [38–41].

A completely different approach to adaptive transform learning schemes is the **third approach** where the transform basis is learned on-the-fly for a given frame or a sequence. These schemes lead to better content adaptation but suffer from high decoder side complexity as the decoder must compute the basis vectors from already decoded regions of a frame [13,14,42,43]. However, these schemes do provide an insight into the fact that there is still a huge correlation at residual level which can be exploited to improve the compression efficiency further.

This chapter will briefly describe some of the advanced transform schemes that have been proposed in the literature under the above three categories. In Section 3.2, the recent transform coding schemes proposed at JVET for the exploration of coding tools for the future video coding standard h.266 are detailed. Section 3.3 discusses other systematic transforms that have been studies in the literature. Section 3.4 and 3.5 briefly describe different offline and online transform learning schemes. Finally, Section 3.6 provides a perspective on the various proposed approaches and why the approach proposed in this work is different from the prior art. Also, possible modifications to already existing methods are also briefly described.

3.2 Future foreseen standard transforms in JVET

Many interesting works in the area of adaptive transforms have appeared recently. They have shown considerable improvement over the latest HEVC standard and are also proving to be a potential candidate in the standardization process of its successor, h.266. These adaptive transform schemes have been implemented in the Joint Exploration Model (JEM) software [44] which is the reference software developed by JVET and is based on the HEVC Model (HM) software. In the following, adaptive transform schemes currently explored by JVET are briefly described.

3.2.1 Enhanced Multiple Transforms (EMT)

The Enhanced Multiple Transform [1] scheme utilizes multiple trigonometric transform basis (sinusoidal family of DCT and DST) in order to compress an intra-predicted or inter-predicted residual block. This scheme has been adopted in the JVET test software JEM [44] for the development of the next generation video codec.

As described in previous chapter, in HEVC, the luma component of a 4×4 intra-predicted residual is transformed using DST-VII instead of DCT-II, because the statistics of 4×4 residuals are modeled better using the DST-VII. In EMT, three more transforms have been added from the family of sinusoidal transforms, namely DCT-V, DCT-VIII and DST-I. The basis function of the DCT-II/V/VIII and DST-I/VII for N-point input are shown below in Table 3.1.

Table 3.1: Transform basis functions for the DCT and the DST-type transforms [1]

Transform Type	Basis function $T_i(j)$, i, j=0, 1,…, N-1
DCT-II	$T_i(j) = \omega_0 \cdot \sqrt{\frac{2}{N}} \cdot \cos\left(\frac{\pi \cdot i \cdot (2j+1)}{2N}\right)$ where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$
DCT-V	$T_i(j) = \omega_0 \cdot \omega_1 \cdot \sqrt{\frac{2}{2N-1}} \cdot \cos\left(\frac{2\pi \cdot i \cdot j}{2N-1}\right)$, where $\omega_0 = \begin{cases} \sqrt{\frac{2}{N}} & i = 0 \\ 1 & i \neq 0 \end{cases}$, $\omega_1 = \begin{cases} \sqrt{\frac{2}{N}} & j = 0 \\ 1 & j \neq 0 \end{cases}$
DCT-VIII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \cos\left(\frac{\pi \cdot (2i+1) \cdot (2j+1)}{4N+2}\right)$
DST-I	$T_i(j) = \sqrt{\frac{2}{N+1}} \cdot \sin\left(\frac{\pi \cdot (i+1) \cdot (j+1)}{N+1}\right)$
DST-VII	$T_i(j) = \sqrt{\frac{4}{2N+1}} \cdot \sin\left(\frac{\pi \cdot (2i+1) \cdot (j+1)}{2N+1}\right)$

Additionally, Figure 3.2 illustrates the first two basis functions of DCT-II, DST-VII and DCT-VIII. It can be seen that the first basis vector of DCT-II, DST-VII and DCT-VIII models constant, gradually increasing and gradually decreasing magnitude distribution. Therefore, these three basis functions together are capable of capturing the residual statistics better compared to only using the single DCT-II.

As the residual statistics vary based on the intra-prediction mode as shown in Figure 2.12, a transform set consisting of two separable transform candidates is chosen based on the intra-prediction mode. Further, for a chosen transform set, a transform H is selected for the horizontal direction and a transform V is selected for the vertical direction based on the encoder mode decision and is explicitly signaled to the decoder. The separable transform T is therefore, represented as

$$T(x, y) = H(x)V(y)$$

Table 3.2 shows the 3 different transform sets being used and Table 3.3 shows the relationship between the chosen transform sets and the intra prediction mode. For inter-coding, only one transform set, the set 0 is applied for both horizontal and vertical transforms.

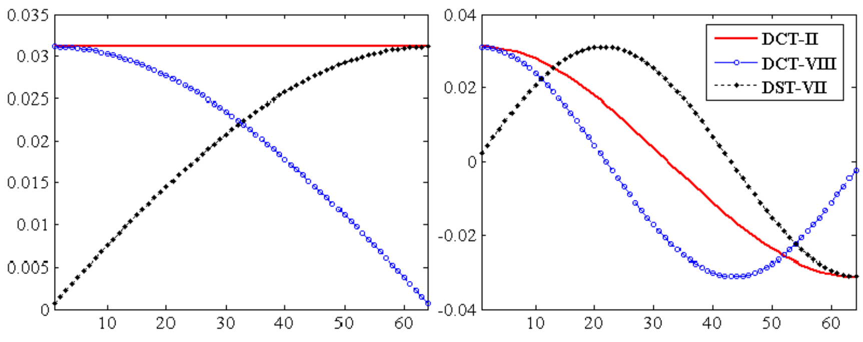


Figure 3.2: Illustration of first two basis functions of DCT-II, DST-VII and DCT-VIII [1]

Table 3.2: Transform set with corresponding candidates [1]

Transform Set	Transform Candidates
0	DST-VII, DCT-VIII
1	DST-VII, DST-I
2	DST-VII, DCT-V

Table 3.3: Transform set implicitly (no signaling) chosen for each IP Mode [1]

Intra Mode	**0**	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**	**14**	**15**	**16**	**17**
H	2	1	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1
V	2	1	0	1	0	1	0	1	2	2	2	2	2	1	0	1	0	1
Intra Mode	**18**	**19**	**20**	**21**	**22**	**23**	**24**	**25**	**26**	**27**	**28**	**29**	**30**	**31**	**32**	**33**	**34**	
H	0	1	0	1	0	1	2	2	2	2	2	1	0	1	0	1	0	
V	0	1	0	1	0	1	0	0	0	0	0	1	0	1	0	1	0	

For signaling the selected EMT mode, a joint CU level and TU level signaling scheme is employed. In this scheme, a CU index = 0 implies that all the TUs under the given CU will utilize DCT-II as the only transform and EMT is disabled for that CU block. However, when the CU index is set to 1, the chosen transform is indexed for each TU. This approach disables EMT for smooth regions where the benefit of applying multiple transforms is less and enables EMT for regions with high texture regions and, therefore, provides adaptive switching between the low cost and high cost transform signaling.

Further, even when EMT is enabled, some residual blocks might have less residual energy and signaling a transform may be costly for such residual blocks. Therefore, a default transform (DCT-II) is only used when the number of non-zero transform coefficients are less than the pre-

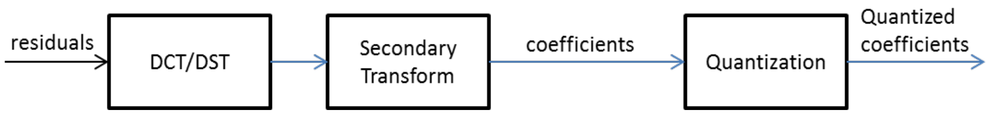


Figure 3.3: Secondary Transforms

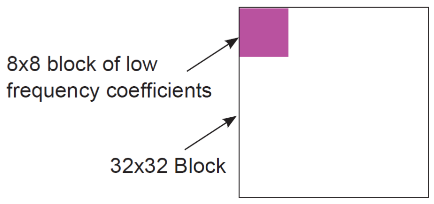


Figure 3.4: Illustration of top-left sub-block where ROT8 is applied

defined threshold and therefore, no index is encoded for this case and the decoder is able to detect this case from the number of decoded transform coefficients.

The scheme achieves 2.81% and 2.29% gain under All Intra (AI) and Random Access (RA) configuration with an increase of 172% and 60% in encoder run-time. The speed-ups proposed in their work leads to a gain of 2.76% and 2.25% gain with encoder complexity addition of 117% and 25% for AI and RA respectively.

3.2.2 Secondary Transforms

The concept of secondary transforms is to exploit the correlation in the transform coefficients after a primary transform has been applied (Figure 3.3). The rotational transforms (ROT) was proposed in [45] and was studied in HEVC [46,47]. Then, it has been replaced by mode dependent non-separable secondary transforms (MD-NSST) that lead to better gains. The two approaches are briefly described below.

3.2.2.1 Rotational Transforms

Rotational Transforms (ROT) is one of the prominent secondary transform studied extensively in the literature. In [46, 47], the two different ROTs are designed, namely 4×4 (ROT4) and 8×8 (ROT8). The ROT4 is applied only to 4×4 residual blocks. For blocks greater than or equal to 8×8, ROT8 is applied to the top-left sub-block of the transform coefficient block as shown in Figure 3.4.

The basic goal of ROT is to enable the exchange of energy between columns and rows of the transformed coefficient block in order to have better compaction and to minimize the quantization error. This exchange of energy is controlled by an angle parameter α_k which is tuned carefully by training on different video sequences. A dictionary consisting of 4 ROTs is learned offline and stored in the memory of both encoder and decoder.

During the encoding process, an R-D search is required to choose the best ROT among the

5 possible choices (4 ROTs + 1 skip ROT). These ROTs are applied only on the top-left frequency coefficients for block sizes greater than or equal to 8x8 (Figure 2.3) and the choice of best ROT is explicitly indexed to the decoder. A fast ROT decision has also been proposed in the same work to speed-up the R-D search operation by estimating the distortion in transformed domain and reducing the unnecessary computation of inverse scaling and inverse transformation.

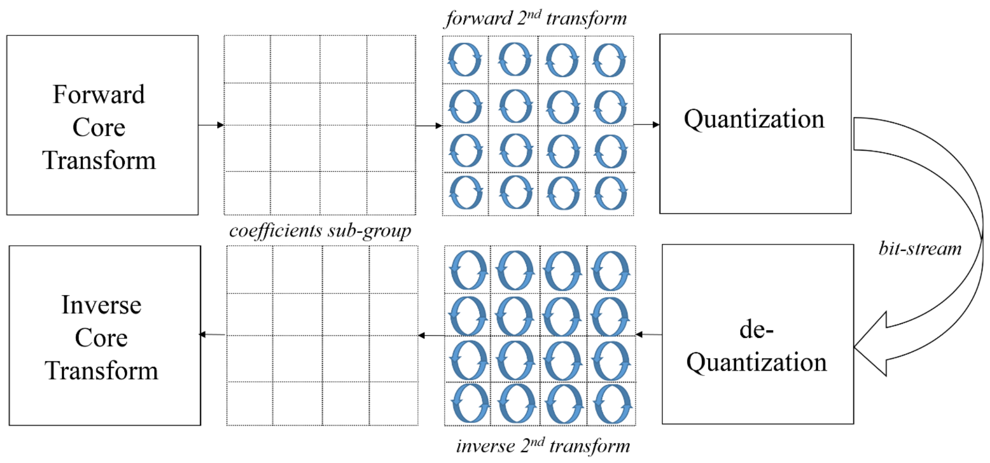


Figure 3.5: Illustration of ROTs applied on coefficient sub-group [7]

Saxena *et al.* in [48] further proposed a mode dependent ROT in order to avoid the expensive R-D search operation and signaling of the overhead bits. The already coded intra prediction mode information is used to determine the choice of secondary transform. This led to considerable reduction in encoder complexity.

Recently, rotation based secondary transforms [49] have been proposed in the next generation video codec standardization activity by JVET. It is proposed to apply the ROTs of size 4×4 on each of the coefficient group inside a coefficient block of size. Figure 3.5 illustrates the usage of secondary transforms. A gain of 1.9% and 1.0% is achieved on top of HEVC with a complexity addition of 230% and 22% under AI and RA configuration respectively.

3.2.2.2 Non-Separable Secondary Transforms (NSST)

A variant of the Rotation-based secondary transform known as Mode dependent non-separable secondary transform (NSST) has been recently proposed by Qualcomm for the future video coding standard. This method has been evaluated on the latest JEM software and the method has shown better gains compared to the ROT [50].

The NSST explores the potential of using non-separable transforms in place of separable transforms. Therefore, ROT of size 4×4 is replaced by a 16×16 non-separable transform T which is applied to a 4×4 coefficient block. Additionally, an 8×8 non-separable transform is applied on the low-frequency coefficients for a block size larger than or equal to 8×8.

Each coefficient block of size N×N is represented as a vector $\vec{X}$ of size N^2×1 and the final transformed coefficients are calculated as $F = T \cdot \vec{X}$ which are then re-organized into a N×N block using the scanning order for that block (*i.e.* horizontal, vertical or diagonal).

Also, a set of non-separable transform matrices is chosen based on the intra prediction mode where each set consists of 3 transform matrices. Due to the symmetry property of 33 directional intra-prediction modes, 17 NSST sets are used for directional modes and 2 sets for DC and planar mode. Therefore, in total 19×3 NSSTs are used. A zero is signaled when NSST is not used, otherwise an index is signaled to indicate the selected NSST matrix [51].

The NSST scheme achieves a gain of 3.5% over HEVC when only 4×4 NSSTs are used and a gain of 6% when both 4×4 and 8×8 size NSSTs are used under AI configuration. The complexity increase is 231% and 319% respectively [51].

3.3 Other systematic Transforms

In the previous section, it was shown that the EMT scheme proposed by Qualcomm uses multiple systematic trigonometric transforms with some local adaptivity in order to achieve higher compression. However, these transforms remain sub-optimal for directional structures that may exist in residual domain. Zhu *et al.* in [52] examined the sub-optimal performance of the 2-D DCT in the case of images with strong directional structures. The experiment conducted by Zhu *et al.* using the non-separable KLT transform learned on the artificially generated directional source shows that the upper bound of the performance improvement compared to the conventional DCT is 15 dB.

Due to the huge potential of improvement by using directional transforms, many related works have emerged in the past that apply systematic transforms in a directional manner to achieve better compression. Some of the related work is explained in this section.

3.3.1 Systematic Directional Transforms

This subsection describes the work where already-known transforms are applied in different directions in order to exploit the directional structure present in the image.

Directional Discrete Cosine Transforms (DDCT)

In order to tackle the limitations of conventional DCT to handle the diagonally oriented edges, a new set of directional DCTs was introduced in [8] and tested on AVC. The work was mainly motivated from the shape adaptive DCT ([53–55]) and six directional modes were introduced as shown in Figure 3.6. These modes were introduced in addition to the horizontal and vertical modes for which the conventional DCT was applied.

For each of these six directional modes, a transform direction is set and the first set of 1-D DCTs of different lengths are applied to the block along the corresponding direction. The second set of 1-D DCTs are then applied to each row and finally, a zig zag scan is applied to convert the 2-D coefficient block into a 1-D vector for subsequent coefficient coding steps. Because these directional DCTs suffer from the mean weighting effect [55], a DC separation and a ΔDC correction method were proposed in [55] to overcome this limitation.

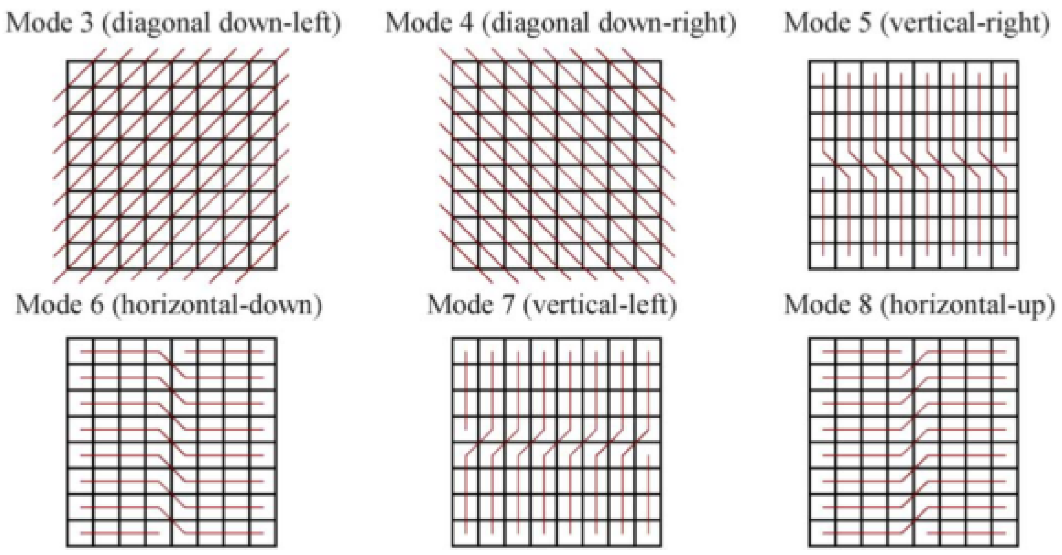


Figure 3.6: Six directional modes for a 8×8 block. [8]

Although the theoretical analysis of using the above proposed directional transforms shows better energy packing efficiency (EPE) and coding gains compared to the conventional DCT, the method suffers from certain drawbacks in terms of encoder complexity and lack of fast implementations for transforms that do not have length as a multiple of 2. Also, additional 3 bits are needed to indicate the mode selected for each image block. Some of these limitations were partially overcome in the schemes described in the next sub-section.

Directional Adaptive Block Partitioning Transform (DA-PBT)

The DA-PBT scheme proposed in [9] was designed to improve the coding efficiency of the above described DDCT scheme by proposing several modifications under two variants. The first variant, also known as Direction Adaptive Block Transform (DA-BT), employs a different scanning order, compared to a zig-zag scan in DDCT, to convert the 2-D coefficient block to a 1-D vector after the application of the directional DCTs. Also, for certain directions, the transform direction is modified compared to DDCT in order to cater the directionality of edges in a better way. The difference in scanning order and transform direction for DDCT and DA-BT is illustrated in the top and the middle section of the figure 3.7 respectively.

In order to avoid the use of DCTs of length (2N-1) in the second stage as done in both DDCT and DA-BT, the scheme was further modified by partitioning the block in the direction of the application of transform as shown in the bottom row of figure 3.7. This scheme is referred as DA-PBT. The first directional DCT (also referred as stage-1 DCT) is applied in the same manner as in DA-BT and the block is split into multiple partitions in the direction of the transform. The second directional DCTs (referred as Stage-2 DCT) do not extend across boundaries and therefore, the length of the DCT in stage-2 is reduced. Additionally, a third transform (referred as stage-3 DCT) is also applied (shown in the form of squares in Figure 3.7) to remove the inter partition correlation. As a consequence, a two-level DC separation/correction is applied which

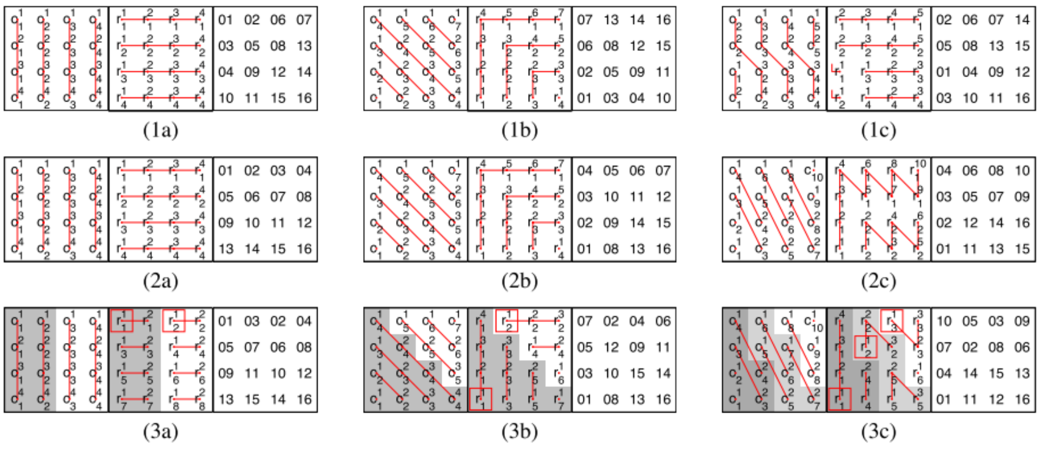


Figure 3.7: Transform directions and coefficient scanning order of DDCT (top), DA-BT (middle) and DA-PBT (bottom) [9]

is similar to the one-level procedure carried in DDCT and DA-BT.

This algorithm provides better compression efficiency than DDCT and DA-BT in case of image compression applications. However, the algorithm was also tested for video coding applications where the DC separation/correction and stage-3 transforms are omitted due to the low energy of the predicted residuals. Also, the direction was inferred from the prediction mode. The result shows a 2dB gain compared to 2-D DCT for image compression. For video compression, the algorithm performs better than the conventional DCT for AI coding in AVC. However, the gains in case of inter prediction were not sufficient when the advanced motion compensation is used.

Directional Adaptive Residual Transforms (DART)

The directional transform schemes for residual coding were further improved in the work proposed by Cohen *et al.* in [10]. The Directional Adaptive Residual Transforms (DART) was introduced for residuals having directional structures.

Similar to DDCT and DA-BT, DART comprise two stages where the stage-1 DCT is applied in a similar fashion as in the previous work using directional DCT. However, a limited stage-2 DCT is only applied along the DC coefficients generated from the primary transform. The focus is on coding prediction residuals that exhibit less correlation in directions orthogonal to the features in the block. Therefore, the stage-2 transforms bypass the AC coefficients. The coefficient output after the stage-1 and stage-2 transforms comprise of stage-2 DC coefficients, stage-2 AC coefficients and stage-1 AC coefficients.

The DDCT and DA-PBT have disadvantage of using a single point or two point 1-D DCTs which have very less coding gains. Therefore, in order to avoid that, the author in [10] proposed a transform path folding method (as shown in Fig. 3.8). The number of DC coefficients output by the primary transform pass is reduced by folding or reflecting the shortest 1-D transform paths such as those near the corners. For the center region, the directional transform mode is

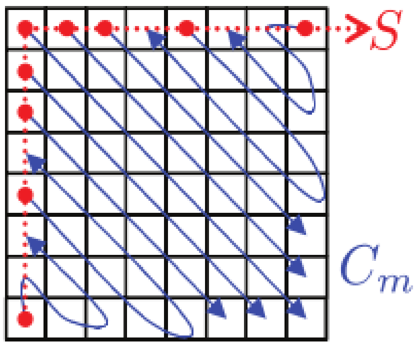


Figure 3.8: Transform Path Folding [10]

used to exploit the correlation in that particular direction. Since the number of DC coefficients is reduced, this shortens the length of the secondary transform. DART algorithm is advantageous in terms of reduced complexity for secondary transforms and elimination of the less efficient shorter transforms that were used in DDCT and DA-PBT.

Further, the correlation between the prediction mode and the directional mode is used to prepare a most probable list of transforms for each prediction mode. Then, CABAC is used in order to encode the transform direction chosen from the list for each block.

3.4 Offline learned Transforms

The schemes described above use data independent known transforms to achieve better content adaptation. A yet another approach is to learn a set of directional transforms offline on a training set which is representative of the different residual statistics present in natural images and video sequences. This transform set is then made available to both the encoder and the decoder. This section will discuss some related works in the category of offline learned transforms.

3.4.1 Adaptive Karhunen Louve Transform (KLT)

As described in section 2.3.2, the KLT has certain limitations when applied to image and video content due to the non-stationary nature of the signal. However, in the past, there have been many attempts to adapt the transforms to the local variations of the signal by employing classification methods that cluster signals with similar statistics in one class. Consequently, a KLT, which is capable of de-correlating the signal more efficiently, is learned for signals belonging to the same class.

The process of classification of signals (say image or residual blocks) is known to be a challenging problem and extensive work has been carried in order to find the right classification criteria and appropriate parameters to separate classes. Methods such as k-means for classification of the signal are presented in [56]. Another method, as proposed in [57,58] for image compression, employs a new unsupervised learning algorithm that combines neural network and competitive learning in a recursive manner to generate a set of adaptive linear transformations. The scheme was deemed to be optimal, both in terms of classification and transformation to minimize the mean square error between the input and the reconstruction. The authors also showed that the learning scheme generates a generalized set of transforms that can be applied to a completely different set of images and still provides less reconstruction error than learning the KLT on a whole image.

The authors in [58] further discussed that such learning methods may be used to design separate networks for different classes of images, where different classes could be regarded as head MRI, body CT, chest X-ray, etc., when dealing with medical images. These networks are made available at both encoder and decoder. This multi-class concept will be further explored in the later part of this thesis and specifically in Chapter 8.

In contrast to the above method which is based on the minimization of the MSE, Effros *et al.* proposed a Weighted Universal Transform Coding (WUTC) algorithm in [59] for joint optimization of transform and bit-allocation for image compression. The encoding process utilizes a collection of transform/bit-allocation pairs to obtain optimal compression, and a Generalized Lloyd Algorithm (GLA) [60] for designing a locally optimal collection of transform/bit-allocation pairs. A similar work has also been presented in [61, 62] which tries to minimize the distortion by recursively optimizing the signal space partition and the local transform coder. The Coding Optimal Transform (COT) further extends the algorithm by integrating the optimization of the signal space partition, the transform and the quantizers [63].

3.4.2 Mode Dependent Transforms

Mode-dependent transforms have been extensively studied in the literature in both AVC and HEVC where these new transforms are shown to have better compression efficiency than the conventional DCT. The key idea behind the design of mode-dependent transforms is to enable capturing different residual statistics for different prediction directions as the average residual statistics vary based on the prediction mode. This is also illustrated in Figure 2.12 of chapter 2.

Ye *et al.* in [35] first proposed a **Mode-Dependent Direction Transform (MDDT)** scheme to improve residual coding in AVC. It involves replacing the DCT by an offline learned 2-D separable transform inside the codec depending on the intra-prediction mode. This scheme requires no additional signaling of transform choice to the decoder that infers the transform choice from the already decoded intra-prediction mode direction. Further, the offline learning process involves collecting a large number of residual blocks per IP-mode from several video sequences, and then computing a KLT H_i per IP-mode i by minimizing the following equation:

$$\arg\min_{H_i} \sum_{j \epsilon S_i} \|r^j - H_i c^j\|_2^2 \qquad \text{s.t.} \quad H_i^T H_i = I \tag{3.1}$$

where, r^j is the j^{th} residual block in a class S_i corresponding to a prediction direction i, and c^j is the corresponding j^{th} coefficient block. Additionally, Ye *et al.* proposed an adaptive coefficient scanning method for each prediction mode in place of the standard zig-zag scan in AVC. The MDDT scheme along with adaptive coefficient scanning showed a gain of 4-4.5% over AVC and was further explored during the development of HEVC standard. It was a part of the test codec KTA [64], developed after the standard h.264/AVC was finalized.

In [36] and [37], the MDDT algorithm was modified by introducing a ℓ_0-norm regularized optimization in order to obtain robust learning algorithm and to enforce the sparsity constraint in the optimization process. Sezer *et al.* in [37] demonstrated that the MDDT scheme was prone to outliers in the residual space and therefore, proposed a **Mode Dependent Sparse Transform (MDST)** scheme by adding a regularization term to the eq. 3.1. This is shown in eq. 3.2 as follows:

$$\arg\min_{H_i} \sum_{j \epsilon S_i} \left(\arg\min_{c^j} \left\{ \|r^j - H_i c^j\|_2^2 + \lambda \|c^j\|_0 \right\} \right) \qquad \text{s.t.} \quad H_i^T H_i = I \tag{3.2}$$

where, r^j and c^j are same as in eq. 3.1, λ is a Lagrange multiplier, and $\|.\|_0$ is the ℓ_0-norm, which is equivalent to the number of nonzero coefficients. The MDST scheme showed a bit-rate reduction of up to 10.2% (and 3.9%) over the DCT (and the MDDT scheme) when tested in AVC.

Recently, these mode dependent transforms have been implemented in HEVC and their coding performance was studies in [65, 66]. It is shown in [65] that the separable MDDT transform fails to provide significant gain over HEVC. This is mainly because the residual statistics in HEVC are very different from residuals in AVC. Due to the significant improvement in the prediction part in HEVC, there is relatively less correlation present in residual blocks generated in HEVC. Further, Arrufat *et al.* in [66] compared the coding gains of separable and non-separable MDDT and MDST schemes over HEVC. It was shown that the non-separable transforms achieve higher compression gain than their separable counterpart but, at the same time, suffer from higher encoder complexity and memory requirement. However, the results in [66] demonstrated that there exists a lot of correlation among residuals generated in HEVC.

Odd type-3 Discrete Sine Transforms

Many works were dedicated to the simplification of the MDDT algorithm, and analysis of residual statistics for different directional modes in HEVC was made in order to model the statisticsc using known trigonometric transforms. Yeo *et al.* in [67] proposed to use an odd type-3 discrete sine transform (ODST-3) [33] in addition to the DCT transform. The gains achieved using this transform were equivalent to separable MDDT. Moreover, the method had very low computational cost and required a very less memory storage compared to the MDDT. Finally, a conceptually similar work carried in [68, 69] was accepted in the working draft-3 of HEVC [70].

Therefore, in the HEVC specification, the DST-7 is used as an alternative transform for intra predicted luma 4×4 blocks only and, for all other cases, the DCT-2 is used. This is mainly due to the lack of fast implementations for the DST of larger sizes, and also due to the lack of significant coding gains in these cases.

3.4.3 Rate Distortion Optimized Transforms (RDOT)

The MDDT and MDST methods described earlier were based on the assumption that there is a correlation between the direction of the predicted residual and the chosen directional prediction mode. However, in [11, 71], it was shown that the mode based classification may not lead to the optimal residual separation in the space. It is shown in [11], that even for the same prediction mode, the residual block may possess different directional statistics. For illustration (as presented in [11]), the blocks obtained below are from the same intra-prediction mode 0 but possess very different statistics. The left block contains a vertical edge whereas, the right block has an irregular structure.

$$\begin{bmatrix} 21 & -2 & -3 & -4 \\ 21 & -2 & -3 & -4 \\ 26 & -1 & -3 & -4 \\ 22 & -1 & -2 & -4 \end{bmatrix} \begin{bmatrix} 7 & -16 & -14 & 14 \\ 7 & -22 & -16 & 11 \\ 13 & 1 & 0 & 3 \\ 44 & 40 & 9 & -7 \end{bmatrix}$$

This shows that a single transform per IP-mode may not be the optimal solution for transforming the residual signals. Therefore, a rate distortion optimized transform (RDOT) learning approach was proposed in [11] where, a single residual block is tested with a set of separable transforms obtained from an offline learning. The chosen transform is signaled to the decoder. Figure 3.9 shows how the two different transform schemes (MDDT and RDOT) are applied to a residual block X.

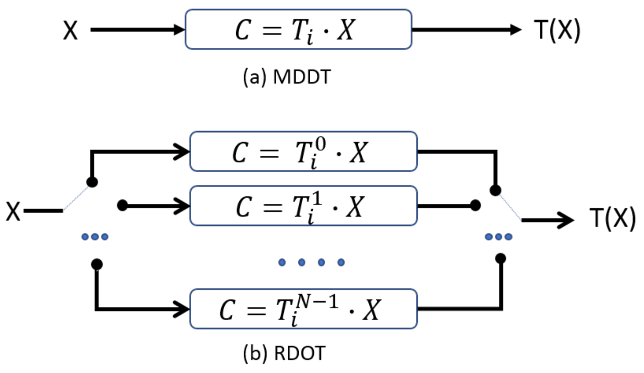


Figure 3.9: Transform schemes of a residual block X for intra prediction mode i in case of (a) MDDT and (b) RDOT [11]

Recently, Arrufat *et al.* in [39] proposed a Mode Dependent Transform Competition (MDTC) scheme in HEVC which, similar to RDOT, test multiple offline learned transforms per IP-mode in competition with the DCT. The transform with minimum rate-distortion (R-D) cost is chosen per TU block. The choice is then explicitly signaled to the decoder. Different offline learning schemes were proposed in [39] and [11].

MDTC has shown a considerable gain up to 7% for non-separable transforms and 4% for separable transforms over the conventional HEVC in the All-Intra (AI) configuration. However, the approach is expensive in terms of both storage and encoder complexity. A part of this thesis focuses on improving the existing MDTC scheme by addressing some of its shortcomings. Finally, in [40], a different approach was proposed where a set of transforms is learned on a huge training set irrespective of the prediction direction. During the test, these new transforms are tested in competition with the DCT and the transform with minimum rate-distortion (R-D) cost is chosen and this choice is explicitly signaled to the decoder. However, the scheme provides less coding gains as compared to MDTC scheme as the cost of signaling the transform choice significantly affects the coding gain. In this thesis, methods to reduce the transform index cost are proposed.

3.5 Online learned transforms

In contrast to the offline learning methods detailed in the last section, an online transform learning method is defined by the process of self-learning of a set of transforms on the content by the encoder. The learned transforms are then coded in the bit-stream as a side information. Another variant of the online learning method involves learning a set of transforms on the past coded regions or pictures. In this case, the decoder is able to compute the transforms from the decoded regions or pictures and therefore, saving the overhead of signaling the learned transforms. In this section, some of the most recent works in this category are detailed.

3.5.1 Content Adaptive Transforms (CAT)

A recent work under this category was carried by Wang *et al.* in [12] where a residual block is sub-sampled into four smaller blocks and adapted transform bases are obtained from the first sub-sampled block. This is then used for transforming the other three sub-sampled blocks. The algorithm exploits the correlation that exists between the sub-samples of the residual block. Figure 3.10 illustrates the sampling method for generating four sub-blocks.

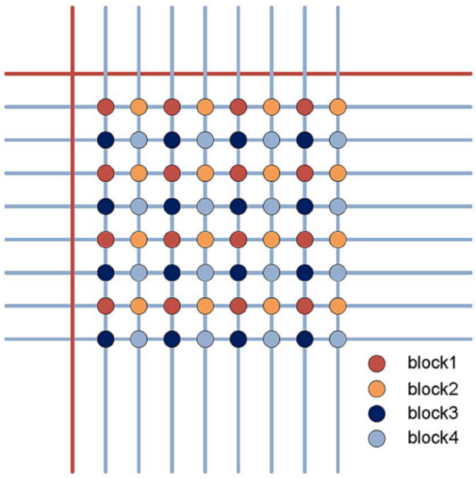


Figure 3.10: Illustration of the sampling operator of [12]

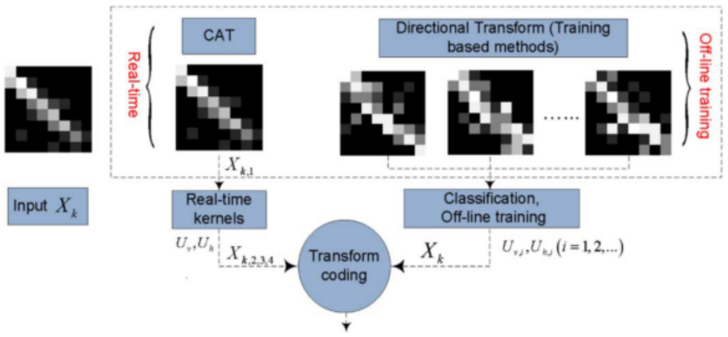

CAT
Directional Transform (Training based methods)
Real-time
Off-line training
Input X_k
$X_{k,1}$
Real-time kernels
Classification, Off-line training
U_v, U_h
$X_{k,2,3,4}$
Transform coding
X_k
$U_{v,i}, U_{h,i} (i = 1, 2, ...)$

Figure 3.11: Comparison of the training based adaptive transform framework and the proposed CAT framework [12]

When coding an input block, the transform generated above is tested in competition with the training based methods such as in MDDT and MDST, and the transform with better de-correlation property in selected as shown in Figure 3.11. The CAT scheme showed about 7.95% reduction of BD-rate over h.264/AVC. Although considerable gain is shown, the huge drawback is that the decoder complexity is increased very significantly.

3.5.2 Signal Dependent Transforms (SDT)

Signal Dependent Transforms [13,42], in contrast to CAT, employ a template matching technique to exploit the non-local correlation present in the reconstructed region of the same picture or previously decoded pictures. Figure 3.12 shows the basic flowchart of training a KLT using N similar patches.

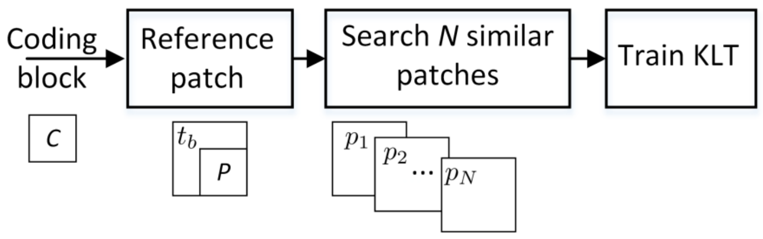


Figure 3.12: Block Diagram of training KLT on N similar patches [13]

For a given coding block shown as C in Figure 3.12, a reference patch is constructed where t_b is the template located top-left of the coding block C, and P is the intra predicted block constructed using t_b. In the next step, N patches similar to the reference patch are searched in the reconstructed region of the same frame or previously decoded frames. A KLT is learned on these N patches and the KLT is then used to compress the current block efficiently. Using this methods, not only the directional structures but also other complex gradient structures can be compressed in a better way.

The computation of the KLT requires an eigenvector decomposition on the decoder side which has a complexity of O(D^3) where, D is the dimension of the vector. Therefore, for a 4×4 residual block, the complexity is O(16^3) which is reasonable. But for higher dimension blocks such as 32×32, the complexity is extremely high and therefore, it cannot be applied directly. A method in [13] has been proposed to reduce the complexity of eigenvector decomposition for blocks bigger than 4×4.

The approach provides promising gains over the current state-of-the-art HEVC video codec with up to 23% bit-saving and has been considered as one of the potential candidate for future video coding technology in JVET [13]. One major drawback of this scheme is the high computational complexity requirement at the decoder. As the decoder needs to perform a template search operation to find the non-local correlated blocks, which are then used to compute KLT, this leads to heavy computation at the decoder end. This is likely to be too much, for the time being, for a practical decoder to be deployed in a mass market.

3.5.3 Other online transform learning schemes

Similar to SDT, the work carried in [72] highlights the use of reconstructed past samples to learn a new transform which can adapt to the correlation properties of the data. A memory factor m is defined as the number of past samples used to generate the new transform. There is no side information to be encoded as the same can be generated on the decoder side using the past samples. The algorithm is again computationally demanding to the decoder and also needs extra memory to store the past samples.

Yet another scheme by Biswas *et al.* in [14] proposed to adapt a transform to the inter predicted residual blocks inside a video codec. In this approach, the transform is obtained from a motion compensated prediction (MCP) block both at the encoder and decoder, hence eliminating the need of encoding the adapted transform bases into the bit-stream. A KLT is learned on blocks of motion compensated prediction error which are generated by subtracting the original block to many shifted and rotated versions of the prediction block. Figure 3.13 shows an example of a motion compensated prediction block and its corresponding residual block. Figure 3.14 shows that the residual statistics of shifted and rotated version are correlated to the actual residual statistics.

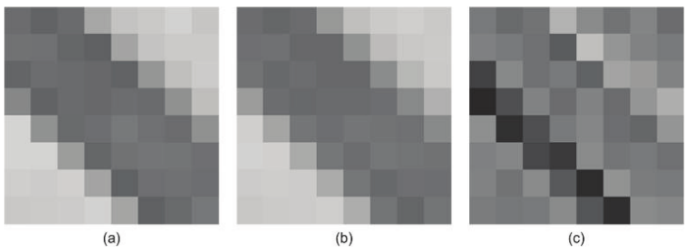


Figure 3.13: (a) Current block (b) Motion Compensated Prediction block (c) MCP error block [14]

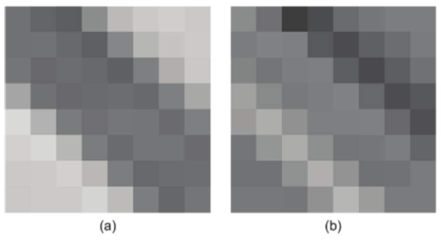


Figure 3.14: (a) MCP block with shift (0,-0.25) and rotation -0.5 degree, (b) MCP block minus shifted-rotated block [14]

Similar to other online approaches discussed above, the downside of this algorithm is that the decoder complexity is very high as it requires the computation of KLT from the motion compensated prediction block. Only 1 bit is required to indicate whether the DCT or a learned KLT is used for a particular block. Results showed an average improvement of 0.9 dB compared to h.264.

Compared to the above approaches mentioned so far, a different approach is presented in [20] and uses a Singular Value Decomposition (SVD) based compression to adapt to the content.

This work proposes to partition the signal space by computing a simple standard deviation on an image block where the blocks with abrupt changes in intensity are coded using SVD and the blocks with smooth variations are coded using DCT. The algorithm involves signaling of a reduced set of eigen basis to the decoder. Further, [73] proposed a method to efficiently encode eigen vectors using vector quantization. The method is computationally less demanding at the decoder but requires heavy computation at the encoder and, also requires signaling of the basis vectors per block as a side information.

The work in this thesis will further explore the online learning scheme where the learned basis vectors are signaled to the decoder in order to keep the decoder complexity low.

3.6 Discussion

The work in the literature that has focused on developing algorithms for h.264/AVC or image compression applications has shown considerable improvements. In HEVC, the residual energy has been further reduced by employing additional directional prediction modes for intra-prediction, higher sub-pixel accurate filters for motion prediction and finer topology using QT. Ma *et al.* demonstrated in [65] that the coding gains of different mode-dependent tools, as described in this chapter, are considerably less when compared to the gains in AVC. Similarly, the gains from directional transform schemes are less compared to the gains obtained on the previous standards [65].

A general conclusion can be drawn here that HEVC de-correlates the input signal better than h.264 leaving less correlation to exploit in the residual blocks. Therefore, the adaptive transform schemes explored in AVC are less efficient in HEVC. For this reason, recently, many new approaches were proposed in HEVC as were described in this chapter. Even though the correlation in the residual blocks in HEVC is less compared to h.264, these new methods have clearly shown that there is still a great potential for further compaction of the residual energy and for achieving higher compression efficiency.

Most of the methods proposed in HEVC use multiple transform candidates instead of a single transform candidate to capture the statistical variations of the residual signal better. These multiple transforms are used as a coding mode and the choice between the transforms is either explicitly signaled (EMT, ROT, NSST and MDTC) or implicitly derived (MDDT and MDST). Additionally, a hybrid signaling is employed which involves indexing the choice of transforms at both CU-level and TU-level. At CU-level, the index is coded to indicate the use of multiple transform candidates or not. If the flag is set, the multiple transform candidates are tested at TU-level and their index is encoded for each TU. This helps the encoder to automatically switch between the expensive TU-level indexing and the CU-level indexing, depending on the region of the video frame.

In summary, the most popular techniques employed to improve the residual coding in HEVC are:

- using multiple transforms instead of a single transform,
- offline learning of transform candidates on a large training set,
- non-separable transforms instead of separable transforms,
- exploiting mode dependency by inferring the coding index from the intra-prediction mode, and

– employing hierarchical indexing of transform index at CU and TU level.

Although the offline transform learning schemes described in section 3.4 have shown significant gains over HEVC, it fails to capture the correlation that may exist within a specific content or region of a sequence. This has been explored in the literature using online learning schemes such as CAT, SDT, etc. as described in section 3.5. These techniques demonstrate much higher coding gains but at the same time suffer from a high decoder complexity. This is mainly because the decoder is required to compute the transform from the past decoded signal. However, this is avoided by using learning schemes that learn a set of content-adaptive transforms at the encoder and signal the learned basis vectors to the decoder. By carefully handling the overhead of signaling the basis vectors, one can achieve coding gains while keeping the decoder complexity low. The work in this thesis will explore the possibility of using an on-line learning scheme for better content adaptation and its performance is compared to the already existing offline learning methods.

3.7 Conclusion

This chapter covered some of the related work carried in developing an adaptive approach on transform coding in a video coding scheme. Some of the prior work on advanced transforms have shown considerable gain in AVC and image coding, but failed to provide equivalent gain in HEVC. This is mainly because the residuals in HEVC have less correlation than in AVC. In order to improve the residual coding in HEVC, some new approaches have been proposed recently that test multiple transforms in competition with core DCT when compressing a residual block. These approaches have shown considerable gain over HEVC which proves that there is still some correlation that still exists in the residual domain. Also, non-separable transforms have proved to be more effective in HEVC in order to achieve higher compression gains. Finally, explicit signaling of transform index is used to indicate the choice of transform to the decoder.

II

The machinery on transform learning and coding

4

Data-Driven Transform Learning

Last chapter covered the different advanced transform methods proposed in the literature to improve the residual coding in both AVC and HEVC. These methods can be divided into two categories based on whether the transforms used for video coding are either known transforms (i.e. model-based transforms) or data-driven transforms. The data-driven transforms are dependent on the underlying data set on which they are learned, hence their name. Depending on whether the data set is generated on-the-fly from the coding of a given sequence or is generated offline from various sequences, the methods are named online learning or offline learning respectively. The data-driven transform based methods have shown significant gains over AVC [35,37,74] and recently over HEVC [39, 43, 66, 75] as already discussed in Chapter 3. The present chapter will focus on the learning part and describe how to learn a transform set using different learning methods.

The goals of this chapter are to firstly describe a typical data-driven transform learning scheme, then analyze parts of the training process in detail and compare different learning schemes from literature on the basis of their learning methods, and finally briefly describe the shortcomings of the current approaches, hence motivating the work carried in this thesis.

The chapter is organized as follows. The first section describes the training process in terms of different elements of the training process and the mathematical conception of the learning scheme. Next section compares the learning scheme of separable and non-separable transforms. Finally, a detailed discussion and conclusion are provided in the last section.

4.1 Analysis of the training process of data-driven adaptive transforms

The performance of data-driven adaptive transforms heavily depends on the training process employed to generate the multiple transform candidates. In this section, design of the training process is analyzed in detail.

4.1.1 Elements of the training process

Most of the data-driven adaptive transform learning schemes proposed in the literature have the following common elements which are important in the design of the training process.

1. **Training set**
   It is a large collection of residuals obtained from various source sequences.
2. **Initialization of the training process**
   Some initial set of transforms is generally required to start the training process.
3. **Classification criteria**
   The residuals are classified into several classes based on a classification criteria.
4. **Transform optimization**
   For each class, an optimal transform candidate is computed for residuals belonging to the class.

Ideally, the training set is designed such that it spans the space of all relevant residual statistics and, therefore, is a good representation of the residuals present in natural videos. For this purpose, these residuals are in general obtained from a diverse collection of video sequences.

Deciding a good initialization of the training process is challenging. In the literature, several methods of initialization have been explored. The initial transforms are either obtained by randomly splitting the training set into K classes and learning a KLT on each class or chosen as K oriented DCT-like transforms that are designed to capture the residual statistics in certain direction. In [76], a method to produce oriented DCT-like transforms is provided.

Apart from the choice of the training set and the initialization, an offline training process is split into two steps. The first step, known as the classification step, is used to classify the residuals into clusters based on some heuristics such as the morphology of the residuals, direction of edges, cost of encoding the residual using initial transforms, etc. Once a good classification is achieved, the next step is transform optimization which is further split into two sub-parts. First, for a given transform, optimal coefficients are obtained for each residual inside a given class. Second, for a given set of optimized coefficients, an optimal orthogonal transform is generated. These two sub-parts are iterated until some convergence is achieved. Once the optimal transforms are obtained for all the classes, the first step is repeated for re-classification. Thus, the two steps are also repeated, until convergence is achieved.

The following two sub-sections present various classification and transform optimization techniques that are known in the literature. Additionally, some new potential strategies that will be explored in this thesis are discussed.

4.1.2 Classification of residual blocks

As the residuals are known to have varying statistical behavior, the general idea is to classify those with similar statistics into one class so that a set of optimal transforms for each class of residuals can be learned. The first challenge here is to find the best criterion to classify the residuals in the residual vector space. Finding the best criterion for classification has always been seen as a challenging problem and has been extensively addressed in the literature. Some of the methods that are commonly used to cluster the residuals in the signal space are presented in the following.

k-means clustering

In [56], a method known as k-means has been described to classify the signals and has also been used in [61] for classification of the image blocks. However, an Euclidean distance-based classification such as k-means may not be suitable for the classification. This is because any two vectors that differ by a scalar multiple have a same set of transformed coefficients but the Euclidean distance-based classification might index these two vectors into different classes as was observed in [58].

Prediction-mode-based clustering

Methods such as the MDDT and the MDST (described in section 3.4.2) cluster the residual blocks based on the the intra-prediction mode. As it is well known in the literature that the residual blocks derived from the same prediction mode depict similar statistics, the prediction mode is usually considered as a good classification criteria. Many recent methods rely on this criteria to drive the classification. Also, the prediction mode is decoded before the computation of residual blocks and, therefore, this classification can be inferred directly from the decoded prediction mode and does not require any explicit signaling.

However, it is further shown in the literature that statistics of the residual signals obtained using the same prediction mode can be very different [38]. Moreover, the residual statistics and the prediction mode are further decoupled in HEVC relative to AVC, and this has led to minimal gains registered by the MDDT and the MDST in HEVC compared to what was obtained on AVC [65].

Energy Packing Efficiency Based (EPE) Clustering

Energy Packing Efficiency (EPE) of a transform T on a given residual block r is the ratio of the sum of the energies of the first M coefficients to the total energy computed using all N coefficients.

$$EPE_M = \sum_{m=0}^{M-1} c_m^2 / \sum_{n=0}^{N-1} c_n^2$$

Here c is the transform coefficient vector obtained by $c = T \cdot r$ where T is the transform and r is the residual vector. It is a measure to compute the effectiveness of the transform T in de-correlating the signal. Therefore, in case of K transforms ($K > 1$), EPE can be used as a criterion to cluster the residual space into K classes. In general, a j-th residual block r^j is assigned to a cluster k that maximize the EPE as expressed in the following objective function:

$$k = \arg\max_{k'} \left(\sum_{m=0}^{M-1} (c_m^j)^2 / \sum_{n=0}^{N-1} (c_n^j)^2 | c^j = T_{k'} \cdot r^j \right) \tag{4.1}$$

The RDOT method described in section 3.4.3 utilizes the above cost function to cluster the residuals into multiple classes.

Rate-Distortion (R-D) cost-based clustering

The mode decisions inside a video codec are generally taken based on the R-D cost where, the distortion D is computed as the mean-square error (MSE) between the original and the reconstructed residual, and the rate R is computed as the number of bits required to encode the transform coefficients and the associated syntax. The mode with least R-D cost is selected inside the R-D optimization (RDO) loop of the video codec.

Classification methods discussed so far do not take into account the R-D cost when clustering the residual blocks. However, some of the recent methods as described in section 3.4.3 utilize some R-D cost as a metric for classification where the objective function used is expressed as below:

$$\text{Cost } C = \|r - T \cdot c\|_2^2 + \lambda \|c\|_0 \tag{4.2}$$

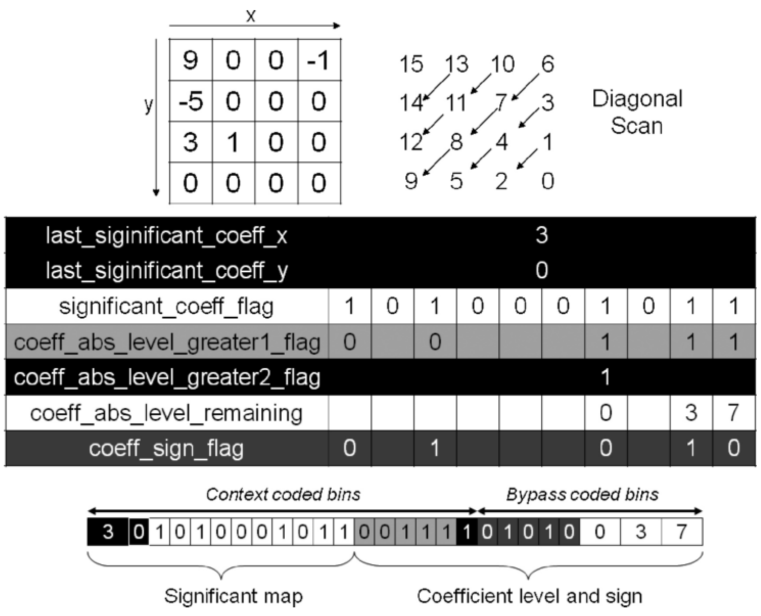
Here, the first term denotes the MSE between the original residual block r and the reconstructed residual block computed as $r' = T \cdot c$ where T is a non-separable transform and c is the quantized coefficient. The second term is an approximation of the real rate of coding the coefficients and is computed as the number of non-zero coefficients c. This approximation is useful in case of offline methods where it is difficult to determine the real rate of coding the coefficients.

In this thesis, the R-D cost is utilized as well for performing the clustering of residual blocks. However, in the online learning framework, the rate term is replaced by the true rate computed using CABAC in HEVC.

Coefficient Group based Clustering

As explained in the section 2.2.6, the transformed coefficient coding is performed by several passes on a single coefficient group. Figure 4.1 illustrates the five level coding of the coefficient with the help of an example.

These levels provide a good representation of the distribution of the coefficients inside a coefficient group and forms a kind of sparsity measure of the coefficients. The clustering can be done based on the levels needed to encode a particular TU block.



last_siginificant_coeff_x	3									
last_siginificant_coeff_y	0									
significant_coeff_flag	1	0	1	0	0	0	1	0	1	1
coeff_abs_level_greater1_flag	0		0				1		1	1
coeff_abs_level_greater2_flag							1			
coeff_abs_level_remaining							0		3	7
coeff_sign_flag	0		1				0		1	0

Figure 4.1: Example of transformed coefficient coding for a 4×4 TU in HEVC [15]

In addition to the above methods, clustering can be performed based on the energy in the pixel domain or transform domain. Further, a single criterion might not be able to efficiently separate the residual vectors into distinct classes. Thus, the methods combining two or three criterion to classify a residual signal may provide better results. For example, in RDOT [11], the residual blocks are first classified based on the intra-prediction mode and then, within each prediction mode class, EPE-based clustering is used to further classify each class into several sub-classes.

4.1.3 Computing optimized transform for each class

Given a classification, an optimal transform T_k for a class S_k is generally computed by minimizing the following equation

$$T_k = \arg\min_{T} \sum_{j \epsilon S_k} \left(\min_{c^j} \left[\underbrace{\|r^j - Tc^j\|_2^2}_{\text{I}} + \lambda \underbrace{P(c^j)}_{\text{II}} \right] \right), \quad \text{s.t.} \quad T^T T = I \tag{4.3}$$

where r^j is the j-th residual block in the class S_k and $c^j = Q(T^T r^j) \in \mathbb{Z}^{\mathrm{N}}$ is the corresponding j-th coefficient obtained after the transform and quantization step Q. The term I in the above equation is the distortion term and the term II, denoted by a penalization function $P(\cdot)$, is modeling the rate. The parameter λ provides a balance between the rate and the distortion term and is usually dependent on the quantization Q. Depending on the value of the penalization function $P(\cdot)$ in term II of equation 4.3, the transform optimization process lead to different methods that have been proposed in the literature.

When P = 0

In case P = 0, the equation (4.3) is reduced to the following:

$$T_k = \arg\min_{T} \sum_{j \epsilon S_k} \|r^j - Tc^j\|_2^2 \quad \text{s.t.} \quad T^T T = I \tag{4.4}$$

where the an optimal transform is simply the KLT on the residuals inside a class S_k and the method is the so-called MDDT.

The KLT basis is computed using the eigen-vectors decomposition of the co-variance matrix on the mean shifted residual data. However, it is shown in [37] that the above computed KLTs are often prone to the outliers and the overall performance of the representation gets affected. Also, it is shown that the performance of the transform obtained using above equation is even worse than the standard DCT for the case of HEVC [65].

When P = $\|\cdot\|_0$

In case P is the $\mathcal{L}_0$ norm $\|\cdot\|_0$, the method becomes the sparsity-based transform optimization schemes such as [77] and the MDST scheme [37] as described in 3.4.2. The equation 4.3 reduces to the following:

$$T_k = \arg\min_{T} \sum_{j \epsilon S_k} \left(\min_{c^j} \left[\|r^j - Tc^j\|_2^2 + \lambda \|c^j\|_0 \right] \right), \quad \text{s.t.} \quad T^T T = I \tag{4.5}$$

where, $\|c^j\|_0$ is simply the number of non-zero coefficients c^j. As discussed in the previous chapter, the MDST scheme tackles the problem of outliers in equation 4.4 and proposes a robust way to compute the KLTs that are regularized by the sparsity constraint. The sparsity term $\|c^j\|_0$ penalizes the residuals that are outliers for the class S_k. Therefore, the optimal transforms, learned by solving the above equation, not only minimize the MSE but also the rate.

The solution of equation 4.5 is not straightforward due to the presence of the $\mathcal{L}_0$-norm term. However, a two-step iterative process as proposed in [77] can solve the minimization problem and involves first finding the optimal coefficients for a given transform and then, computing optimal transforms for the given coefficients. Below are the details of these two steps carried for the optimization.

Step 1: optimal coefficients for a given transform - In the case of $\mathcal{L}_0$-norm, the optimal coefficients c_{opt} for a given transform are easy to compute and are found by hard-thresholding the transform coefficient c^j where the threshold value is related to the λ as follows [77]:

$$c_{\text{opt}}[n] = \begin{cases} c^j[n], & \text{if } |c^j[n]| \geq \sqrt{\lambda} \\ 0, & \text{otherwise} \end{cases} \quad \text{and} \quad n = 1 \cdots N^2 \tag{4.6}$$

Step 2: optimal transform for given coefficients - Once the optimal coefficients are found for the whole class S_k, the optimal transform T_k is computed by minimizing the reconstruction error as follows, as explained later,

$$T_{\text{opt,k}} = \arg\min_T \left(\sum_{j\epsilon S_k} \|r^j - Tc^j_{\text{opt}})\|_2^2 \right) \quad \text{s.t.} \quad T^T T = I \tag{4.7}$$

where, c_{opt} are the coefficients computed in Step 1. The solution to the above minimization equation can be obtained by a SVD decomposition of an intermediate matrix Y_k defined by

$$Y_k = \sum_{j \in S_k} c^j_{\text{opt}} (r^j)^T$$

$$Y_k = UDV^T \quad \text{with} \quad U^U = UU^T = I \quad \text{and} \quad V^T V = VV^T = I$$

The solution to the minimization problem is given by

$$T_{\text{opt,k}} = VU^T$$

where, U and V matrices are obtained from the SVD of Y_k [77]. In other words, the optimal transform is computed by eigen vector decomposition of the cross co-variance between the residuals and the optimal coefficients found in the Step 1.

After the new set of $T_{\text{opt,k}}$ is obtained, the optimal coefficients are again computed using equation 4.6. The above two steps are iteratively solved until convergence or is stopped after a fixed number of iterations.

The transform optimization process described above approximates the rate of coding the coefficients by a simple $\mathcal{L}_0$-norm which in supposed to model the significant flag in HEVC as described in 2.2.6. However, this model does not efficiently fit the real rate of coefficient coding computed in HEVC using CABAC which practically depends on coding of other syntax information such as last significant coefficient position coding, level greater than 1, level greater than 2, etc. as described in section 2.2.6. Also, the rate depends on the position of coefficients inside a coefficient block and the scanning order. Each of the syntax information is further influenced by the probabilities of the past encoded syntax from already coded regions. However, in order to simplify the optimization process, the actual rate is approximated by the rate of coding a significant flag, and the effect of other syntax elements as well as the past probabilities is ignored.

Another challenge is to find a relevant value of λ. The λ parameter provides a balance between the rate and the distortion term. A high value of λ penalizes the rate term more, thus favoring the low bit-rate coding. In HEVC, the value of λ is tied to the quantization parameter (QP) value which is set by the user and has been empirically computed. However, due to the approximation of the rate term in above optimization equation, it is unavoidable to re-compute the optimal values of λ per QP. There are several methods proposed in the literature that have addressed this issue and are detailed later in this chapter.

When P = CABAC

In case P = CABAC, the equation can be represented as:

$$T_k = \arg\min_{T} \sum_{j \epsilon S_k} \min_{c^j} \left(\|r^j - Tc^j\|_2^2 + \lambda R(c^j) \right) \quad \text{s.t.} \quad T^T T = I \tag{4.8}$$

where, R is the true rate of encoding the coefficients using CABAC. Similar to the previous case, the solution to equation 4.8 can be achieved using a two step iterative process as follows:

1. Optimize the coefficients for a given transform

$$\forall j \quad c^j = \arg\min_{c} \left[\|r^j - T_k c\|_2^2 + \lambda R(c) \right] \quad , T_k \text{ fixed}$$

2. Optimize the transform for given coefficients by solving

$$T_k = \arg\min_{T} \sum_{j \epsilon S_k} \|r^j - Tc^j\|_2^2 \quad , c^j \text{ fixed}$$

The most challenging task here is to find the optimal set of coefficients for a given transform set by minimizing the equation shown above in step 1. Within a given class S_k of residual blocks of size $N \times N$, the corresponding transformed coefficient blocks can be represented as in vector form of length N^2 and c_i^j denotes the i-th coefficient channel inside the vector c^j where $i \in \{0 \cdots (N^2 - 1)\}$. The equation in step 1 can be divided into two independent problems to solve the overall optimization equation. Because T_k is unitary, the first part can be reformulated as follows,

$$c^j = \arg\min_{c} \|T_k^T r^j - c\|_2^2 \quad \forall j$$

This can be rewritten as

$$c^j = \arg\min_{c} \|c_r^j - c\|_2^2 \quad \forall j$$

where, $c_r^j = T_k^T r^j$. As both the terms inside this $\mathcal{L}_2$ norm are vectors, it can be simplified as follows:

$$c^j = \arg\min_{c} \sum_{i} |c_{r,i}^j - c_i|^2 \quad \forall j$$

The second part of the equation in step 1 can be solved by assuming the following hypothesis

$$R(c^j) = \sum_i b(c_i^j)$$

In HEVC, the coefficient blocks (CBs) are encoded using the CABAC as detailed in section 2.2.6. The multipass approach employed by CABAC can be roughly modeled where $b(\cdot)$ is the model which converts the coefficient value into the number of bins (or bits after binarization but before arithmetic coding) required to encode said coefficient value. The overall optimization equation can be written as follows:

$$c^j = \arg\min_c \sum_i \left(|c_{r,i}^j - c_i|^2 + \lambda b(c_i) \right) \quad \forall j \tag{4.9}$$

and can also be expressed as a function of c_i^j as:

$$\forall j \forall i \quad c_i^j = \arg\min_{c_i} f_j(c_i)$$

where

$$f_j(c_i) = |c_{r,i}^j - c_i|^2 + \lambda b(c_i)$$

These functions are minimized one by one for each of the coefficient c_i^j in order to find the best coefficient vector c^j. In case the second term is approximated by an $\mathcal{L}_0$-norm, the result of the minimization of the function f$(\cdot)$ is a simple conditional thresholding of the c_i^j as in equation 4.6.

The above optimization process is repeated for all residual blocks inside a given class S_k, and the newly obtained coefficients are fed to the step 2 that computes an optimal transform by minimizing the reconstruction error between the residuals and the newly computed coefficients in a similar way as done for equation 4.7. The two steps are iterated until the convergence or for a fixed number of iterations.

The minimal solution to the equation 4.8 can be approximated by using the above described two-step iterative approach where a model $b(\cdot)$ is defined for coefficient coding and provides rate closer to the true rate of coding coefficients using CABAC. The goal of approximating the penalization function P closer to CABAC is to generate transforms that are optimal in the true rate-distortion fashion instead of the approximation using ℓ_0-norm. Finally, with a penalization function P closer to CABAC, it can be expected that the relationship between the λ parameter as derived for HEVC will hold true.

Even though the above process is designed to optimize the transform in terms of true R-D cost, the computation required to perform the above iterative process is very intensive. However, in case of learning where the computation of the optimal transforms is done offline, the complexity of the learning scheme is generally neglected as this is performed once for all in order to generate a dictionary of transforms. On the contrary, in case of online learning where the optimal transforms are learned on a given sequence by performing self-learning, the above optimization process is largely impractical.

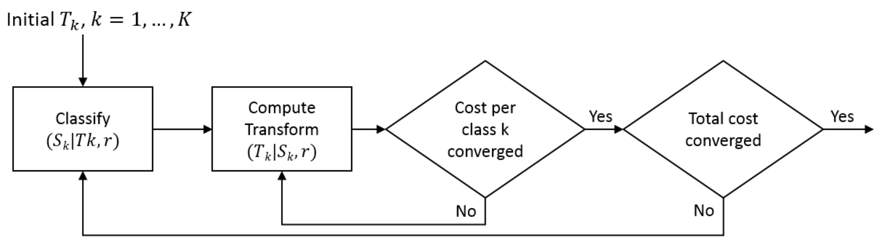

Initial $T_k, k = 1, \dots, K$
Classify $(S_k|Tk, r)$
Compute Transform $(T_k|S_k, r)$
Cost per class k converged
Yes
No
Total cost converged
Yes
No

Figure 4.2: Block Diagram of a typical data-driven transform learning process

4.1.4 Re-classification

As described in this chapter, a data-driven transform learning process first classifies a given training set into multiple classes based on a classification criteria or hypothesis and then, for each class, an optimal transform is learned in a transform optimization process by minimizing equation 4.3. Once an optimal transform is learned on all the classes, a new classification of the residual space can be obtained depending on the type of classification criteria used. For example, in case of EPE-based clustering, the residual classification can be improved by again computing the EPE of each new transform on residual blocks. However, in case of intra-prediction mode based clustering, the new classification does not depend on the transform and therefore, there is no need to perform such a new classification. Figure 4.2 illustrates the iterative learning process with the help of a block diagram.

4.1.5 Finding a consistent value for λ

The parameter λ drives the balance between the distortion term and the rate term in the optimization process described above. Due to the approximation of the rate by a $\mathcal{L}_0$-norm, the value of λ needs to be adjusted accordingly. Several methods have been proposed in the literature to compute the value of λ and can be divided into two main approaches as detailed below.

Annealing λ

Simulated annealing is a technique to find a solution close to a global minima for any optimization process. The naming has been inspired from annealing process in metallurgy where a material is heated to a certain initial temperature Te_{init} and is then cooled down slowly at a rate γ till a final temperature Te_{final} is reached. Similarly, in simulated annealing, the randomness of the solution is high at the beginning and is slowly reduced in order to make the solution converge to a global minima instead of a local minima. Randomness helps escaping from local minima.

The works carried in [37] and [36] utilize annealing techniques to determine the best value of λ. In [36], the value of λ is initially set to zero which makes the equation 4.3 convex and the rate function is slowly introduced by increasing the value of λ in order to find a better minima. In contrast to this approach, Sezer *et al.* in [37] propose to start with a large value of λ which favors the sparsity term in the optimization equation 4.3 and reduce the value of λ until the solution converges. Moreover, different values of λ are computed for different frequency components in this case. It is shown in their work that for the lower frequency components, the solution converges towards larger value of λ, and for the higher frequency components, the solution is observed to converge towards smaller values of λ. Both methods have provided gains over using a fixed value of λ. However, the comparison of these two methods has not been made in the literature.

Computing λ using a close-form solution

In a video codec, the value of λ depends on the QP value set by the user. Motivated from this fact, [36] drives a relationship between the λ and the quantization step size s as $\lambda = (sz)^2$ where, z is the parameter that controls the width of the dead-zone in a video codec. Therefore, given the value of z and s, one can compute the value of λ using the above relationship.

Zou *et al.* in [75] derives another close-form solution by comparing the term $\lambda \cdot \|c^j\|_0$ from equation 4.5 with the $\lambda_{\text{codec}} \cdot R(c^j)$ where λ_{codec} is the regularization term used in the codec and depends on the QP value. $R(c^j)$ is the actual rate of coefficient coding inside the codec. Further, the author approximates the actual rate $R(c^j)$ with the sum of entropies $H(c^j_{m,n})$ computed at each frequency position (m,n). Finally, the equation to compute the λ term is written as:

$$\sum_{m,n=0}^{N-1} \lambda_{m,n} E[\|c^j_{m,n}\|_0] \simeq \sum_{m,n=0}^{N-1} \lambda_{\text{codec}} H(c^j_{m,n})$$

The above equation is solved by assuming a laplacian distribution of the coefficients with zero mean and variance v and a uniform quantization with a dead-zone. The final derived relationship is express as [75]

$$\begin{aligned} \lambda_{m,n} = \frac{\lambda_{\text{codec}}}{\ln 2} \Big[&-(e^{(1-z)\Lambda_{m,n}Q} - 1) ln(1 - (e^{-(1-z)\Lambda_{m,n}Q}) \\ &+ ln2 \ + \ ln(1 - e^{-\Lambda_{m,n}Q}) \ - \ z\Lambda_{m,n}Q \ + \ \frac{\Lambda_{m,n}Q}{1 - e^{-\Lambda_{m,n}Q}} \Big] \end{aligned} \tag{4.10}$$

where $\Lambda_{m,n}$ is inversely proportional to the variance v of the coefficients, Q is the quantization step size and z is related to the width of the dead-zone. The above expression shows that the value of λ for each channel is dependent on the variance of the coefficients in each channel. As the variance of higher frequency components is generally lower than the variance of the low frequency components, the value of λ is therefore, lower for high frequencies, and vice-versa. This is similar to the observation made by Sezer *et al.* in [37].

4.2 Separable versus non-separable transform learning

The above methods illustrate the training process to learn a non-separable transform set T_k with K transforms where $k \in \{0 \cdots K-1\}$ is a class index. However, in the literature, non-separable transforms are deemed as computationally complex. Therefore, separable transforms, which consist of pair of horizontal H and vertical V 1D-transforms are advantageous for practical use.

The optimization equation 4.3 in case of separable transform learning is re-written as

$$(H_k, V_k) = \underset{H,V}{\arg\min} \sum_{j \epsilon S_k} \left(\min_{c^j} \left[\|r^j - Vc^jH^T\|_2^2 + \lambda P(c^j) \right] \right), \quad \text{s.t.} \quad V^TV = I \text{ and } H^TH = I \tag{4.11}$$

The above equation is then solved following the steps:

1. **Optimal coefficients for a given transform**

$$c_{\text{opt}} = \arg\min_{c^j}(\|r^j - V_k c^j H_k^T + \lambda P(c^j)).$$

For a penalization function P approximating the $\mathcal{L}_0$-norm, the optimal coefficients are obtained by thresholding the coefficients $V_k^T c^j H_k$ with $\sqrt{\lambda}$.

2. **Optimal vertical transform for given coefficients** The optimal coefficients computed in the last step are used to compute an optimal vertical transform by solving the following equation:

$$V_{\text{opt,k}} = \arg\min_{V_k}\left(\sum_{j\in S_k} \|r^j - V_k c^j_{\text{opt}} H_k\|_2^2\right) \quad \text{s.t.} \quad V_k^T V_k = I.$$

The solution of this equation is found by SVD decomposition of the intermediate matrix Y which is computed in a similar manner as described above in Step 2 for equation 4.7 and is detailed in [37]. The transform H_k is kept constant during this optimization.

3. **Optimal coefficients with given newly computed vertical transform** The optimal coefficients found in step 1 are updated by using the newly computed vertical transforms obtained in the previous step.

4. **Optimal horizontal transform for a given coefficients and vertical transform** An optimization is performed similar to step 2 above where the vertical transform is kept constant and an optimal horizontal transform $H_{\text{opt,k}}$ is computed as follows:

$$H_{\text{opt,k}} = \arg\min_{H_k}\left(\sum_{j\in S_k} \|r^j - V_{\text{opt},k} c^j_{\text{opt}} H_k\|_2^2\right) \quad \text{s.t.} \quad H_k^T H_k = I.$$

These steps are iterated until convergence is achieved and the final optimized pair of transforms $(H_{\text{opt,k}}, V_{\text{opt,k}})$ is either stored as a dictionary of transforms or used to perform the classification step in case of re-classification during the training process.

4.3 Discussion and Conclusion

This chapter has discussed various classification and transform optimization strategies that can be used inside the training process of data-driven transform learning schemes. Most of recent works have focused on learning an offline set of transforms, also referred as dictionary of transforms, on a large training set. The sparsity based transform learning scheme proposed by Sezer *et al.* in [77] is the basis of the different adaptive transform learning schemes proposed so far. The transforms learned using this technique have better convergence than the transforms obtained using the KLT on a set of residuals in a given class.

Inside a video codec, a dictionary of transforms is stored at both encoder and decoder sides. At the encoder side, these candidate transforms are tested either in competition with the conventional DCT/DST inside an RDO process or replace the DCT/DST. The index is either explicitly coded in the bit-stream or inferred from other syntax elements such as prediction mode in case of the MDDT and the MDST to indicate the chosen transform. In the decoder, before the inverse transformation of the decoded coefficients, a transform index is either explicitly decoded from the bit-stream or implicitly derived from the causal information that indicates the transform to be utilized from the dictionary to successfully decode the residual block.

Table 4.1: Summary of different offline transform learning schemes

Transform learning methods	Separability	Classification criteria	Transform optimization	Codec	Signaling method of transform choice
MDDT (Ye et. al.)	Separable	Prediction mode based only	KLT (equation 4.4)	AVC	No signaling (inferred from prediction mode)
MDST (Sole et. al., Sezer et. al.)	Separable	Prediction mode based only	Sparse Transform i.e. $\mathcal{L}_0$-norm based (equation 4.5)	AVC	No signaling (inferred from prediction mode)
MDST (Adria et. al.)	Non-separable	Prediction mode based only	Sparse Transform i.e. $\mathcal{L}_0$-norm based (equation 4.5)	HEVC	No signaling (inferred from prediction mode)
RDOT (Zhao et. al.)	Separable	Prediction mode + Energy Packing Efficiency (EPE) based (equation 4.1)	KLT (equation 4.4)	AVC	Explicit signaling in the bit-stream
RDOT – Lloyd-type (Zou et. al.)	Separable	Prediction mode + R-D cost based (equation 4.2)	Sparse Transform i.e. $\mathcal{L}_0$-norm based (equation 4.5)	AVC/HEVC	Explicit signaling in the bit-stream
MDTC (Adria et. al.)	Non-separable	Prediction mode + R-D cost based (equation 4.2)	Sparse Transform i.e. $\mathcal{L}_0$-norm based (equation 4.5)	HEVC	Explicit signaling in the bit-stream

4.3.1 Comparison of state-of-the-art offline learning schemes

In the literature, several schemes on offline transform learning have been proposed and differ in various ways such as classification methods or transform optimization techniques, and are summarized in Table 4.1. As observed in the table, the Rate-Distortion (R-D) metric as proposed by [77] is often used in these methods for transform optimization and is expressed by equation 4.5 in order to learn a dictionary of separable or non-separable transforms. The R-D metric is an approximation of the actual R-D balance as it is difficult to estimate the actual bit-rate in offline learning. Also, table 4.1 shows that separable transforms are mostly used due to their lower complexity. Almost all the methods rely on the classification based on the prediction mode. However, in some methods, a second classification criteria is used to further classify the residuals within each prediction-mode-based class. Using this classification, several transform candidates are learned per prediction mode and therefore, during the encoding inside a video codec, an index is coded explicitly to indicate the transform choice as described before.

In general, all the techniques proposed so far rely on an offline training set to learn a set of transforms. As mentioned before, the training set is chosen such that it captures the complete space of possible residual statistics that may exist in real videos. However, these methods fail to exploit the content specific correlation that may exist within the residuals belonging to a frame or a group of frames and consequently are not fully content adaptive. Exploiting this content specific correlation could possibly produce better gains. Online learning schemes proposed in the literature use methods where the already encoded regions of a frame or already encoded frames are used to learn adaptive transforms. These methods however, require heavy computation at the decoder to learn the same set of adaptive transforms from the past decoded information, hence complexifying the decoder.

Based on the above observations, a new online adaptive transform learning scheme has been proposed in this thesis in order to exploit the content specific correlation and to obtain an optimal set of transforms, at encoder side, which are then transmitted to the decoder. The method proposed in this thesis does not rely on the past frames but performs a self learning through a multi-pass approach. The next chapter will present the proposed online learning scheme in detail and compares its R-D performance with the R-D performance of different offline learning schemes discussed above.

5 Online and Offline Learning

5.1 Introduction

Chapter 4 has covered the general principles of data-driven transforms and has discussed the learning aspect of the data-dependent transforms which is essential for the good design of optimal transforms for video coding.

This chapter explores the data-driven learning scheme under two flavors: 1) online learning and 2) offline learning. A novel online learning scheme proposed in this work is described in detail in the first part of this chapter. A set of non-separable transforms is learned on a single frame of a sequence, and then the newly computed transforms are utilized for encoding the same sequence. Therefore, the complete encoder of the proposed online learning scheme optimizes three aspects: learning, selecting and signaling of a set of optimal transforms. This chapter describes the several methods that have been proposed to improve each of the above three aspects of the learning scheme.

The second part of this chapter focuses on the offline learning of data-driven transforms. A simple offline learning scheme based on one of the classification and optimization methods described in Chapter 4 is detailed. Finally, both the online and the offline learning are compared in the last part of this chapter, and numerical results will induce the discussion on the merits and demerits of both learning.

5.2 Online Adaptive Transform Learning Scheme

The proposed online learning is described in detail hereafter. First, the optimization algorithm used for transform adaptation is described. An iterative approach is proposed for learning the transforms on the content. The implementation details of this algorithm and the results are discussed in the subsequent sections.

5.2.1 Principles of the Algorithm

The algorithm is inspired from the work described in [77]. In this work, the proposed algorithm adapts the offline transform learning scheme presented in [77] to the current block-based coding engine (HEVC) to obtain an online learning scheme. Further, in contract to learning schemes developed in the literature (Table 4.1) that utilize the ℓ_0-norm as an approximation of the rate, our proposed method utilizes the accurate rate of coding a block computed using an entropy coder (CABAC) for the classification purposes and to learn the sparse transforms.

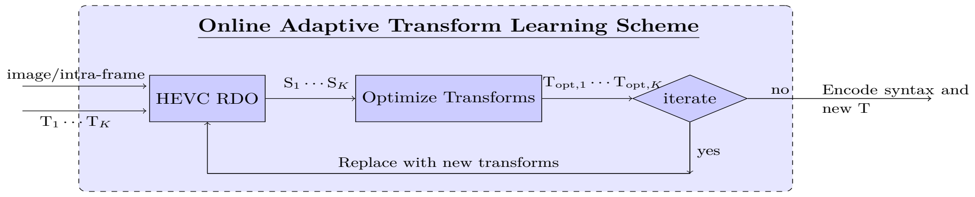


Figure 5.1: On-the-Fly Learning Algorithm

Figure 5.1 shows the block diagram of the learning scheme employed using HEVC as the rate function. The algorithm is split into a two-step optimization process where first the residual blocks generated during the encoding of a frame are classified on-the-fly into K classes $(S_1 \cdots S_K)$ using the HEVC RDO, and second, given a class S_k, an optimal non-separable transform $T_{\text{opt},k}$ is computed by minimizing the reconstruction error between the residuals and their corresponding quantized coefficients generated by the codec. These two steps are iterated until convergence. The various steps related to the training process are detailed below.

Initialization

The codec is initialized by a set of K non-separable transforms $T_1 \cdots T_K$ of size $N^2 \times N^2$ and are referred as non-DCTs hereafter. Having an appropriate initial set of transforms as

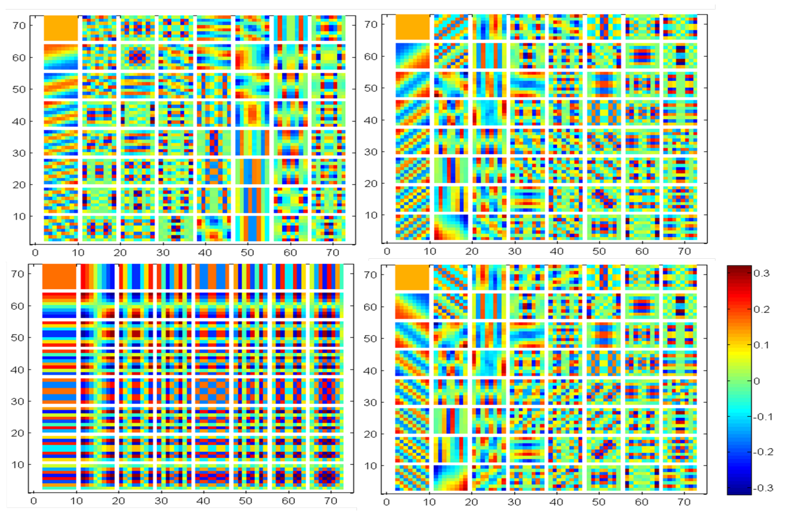


Figure 5.2: Figure illustrating Diagonally Oriented DCT like transforms

a starting point is a crucial step in the optimization process. Therefore, two different methods have been employed for initialization. In the first method, the K initial transforms are computed by randomly splitting the residuals generated from the coding of a frame or set of frames into K classes and then learning the KLT on each class. This is similar to the initialization methods used in the literature. In the second method, the codec is initialized using K non-separable DCT-like oriented transforms aligned in the directions ranging from $0°$ to $180°$. The method to compute these oriented transforms is detailed in [76]. Figure 5.2 shows an example of these oriented transforms for four different directions.

Classification of Residuals

The initial set of non-DCTs T_k where $k \in \{0 \cdots K\}$ are tested on each residual block r^j inside the HEVC Rate Distortion Optimization (RDO) loop in competition with conventional HEVC transforms (DCT or DST), and residuals are classified into K classes $S_1 \cdots S_K$ based on the cost function shown in the equation 5.1.

$$\text{label}\{r^j\} = \arg\min_k \left\{ \underbrace{\|r^j - T_k Q^{-1}(c_k^j)\|_2^2}_{\text{I}} + \lambda(\underbrace{R(c_k^j)}_{\text{II}} + \underbrace{R_T}_{\text{III}}) \right\} \tag{5.1}$$

In the above equation,
- term I denotes the distortion due to the quanziation Q,
- term II is the actual rate of encoding for the transformed and quantized coefficients c_k^j, where $c_k^j = Q(T_k^T r^k)$, using CABAC and the HEVC coding engine, and
- term III denotes the rate of indexing the chosen transform T_k, i.e. of signaling k.

Practically, the label is found by an extended RDO loop during an HEVC encoding.

Optimization of transforms

Based on the above classification, an optimal transforms $T_{\text{opt},k}$ is computed per class S_k using equation 5.2 which minimizes the reconstruction error for each class S_k.

$$T_{\text{opt},k} = \arg\min_H \sum_{j \epsilon S_k} \|r^j - HQ^{-1}(c_k^j)\|^2 \quad \text{s.t.} \quad H^T H = I \tag{5.2}$$

where the c_k^j are fixed and are obtained from the transform coding of the residuals inside the HEVC encoder. The solution of the above equation is obtained by a SVD decomposition of the cross-correlation between the residuals and the quantized coefficients within each class. The solution is detailed in [77] and has been briefly described in Chapter 4 section 4.1.3.

In contrast to the transform optimization method detailed in [77], where the above computed transforms are used to generate an optimal coefficients, the computed transforms are fed back to the HEVC RDO for re-classification step as shown in Figure 5.1. The step involving computation of optimal coefficients for a given transform is skipped in case of the proposed online learning scheme in order to reduce the complexity of the learning scheme.

Thus, we have defined an iterative process of classification using HEVC encoding and transform optimization using SVD-based minimization of equation 5.2 that converges or is stopped after a certain number of iterations. Figure 5.3 illustrates the convergence of the learning scheme

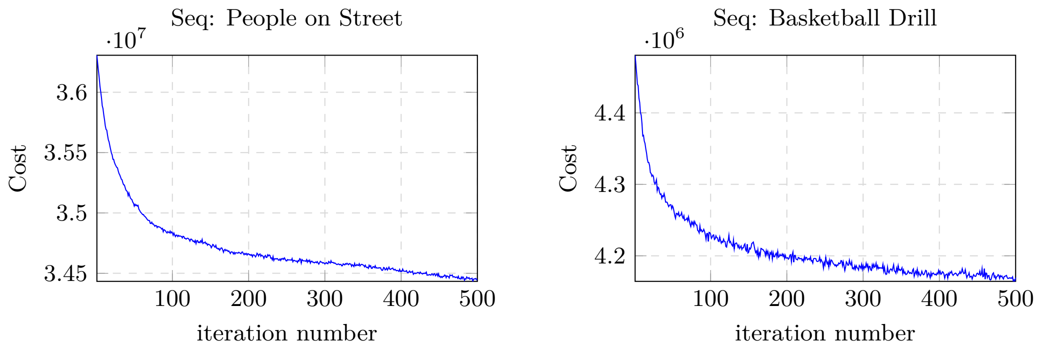


Figure 5.3: Cost Convergence vs Iteration

for two different sequences in terms of coding cost of a given frame. After convergence, the frame is finally coded in the bit-stream using the optimized non-DCTs. The choice of the best transform per residual block is coded by using a transform index. Also, the basis vectors of the learned non-DCTs must be signaled to the decoder, thus adding extra information to the stream.

The proposed algorithm differs from the state-of-the-art methods in many ways. First, an online learning approach is employed to leverage the true rate of encoding the residuals and the associated transform index in order to perform the classification step, whereas methods such as [38] perform a K-means clustering based on the EPE and other methods [36, 39, 40] approximate the rate of coefficient coding using an ℓ_0-norm. Further, the transform optimization step, in contrast to sparsity-based transform optimization methods, is simplified for online learning and utilizes the quantized coefficients generated by the video codec during the encoding process. This simplification helps in keeping the complexity of the proposed scheme under a practical limit.

5.2.2 Proposed Adaptive Directional Transforms in HEVC

This section provides the detailed implementation of the algorithm described in the previous section in the HEVC test software (version HM15.0). A total of K non-separable transforms $(\mathbf{T}_k^{8\times8})$ of size 64×64 and L non-separable transforms $(\mathbf{T}_l^{4\times4})$ of size 16×16 are used as initial transforms found using one of the initialization methods described before.

The HEVC encoder (HM) is modified such that, for encoding a frame, the initial non-DCTs are tested in competition with the core HEVC transforms on 8×8 and 4×4 residuals in order to classify the residuals in terms of R-D cost and then, for each class, an optimal transform is learned. This is repeated for a fixed number of iterations. The whole sequence is finally coded using the transform competition between the final learned set of non-DCTs and the core DCT/DST transforms. All the changes are applied only on the intra luma component. The chroma components are transformed using the conventional DCT. Also, inter residual blocks are coded using the conventional DCT. However, the intra blocks appearing in the inter-picture are coded using the transform competition between the learned non-DCTs and the core DCT.

A transform index per TU block is coded in the bit-stream to indicate the selected best candidate transform. Additionally, the basis vectors of the learned non-DCTs are also coded in the bit-stream for successful decoding of the coded coefficients. The methods used to efficiently encode this additional side information are detailed in the subsequent sub-sections.

5.2.2.1 Smart re-ordering of quantized transform coefficients

In HEVC, a mode dependent adaptive scanning as described in 2.2.6 is used to convert the 2-D coefficient block into a 1-D coefficient vector for entropy coding using CABAC. The aim of adaptive scanning is to provide coefficients in a CABAC friendly order which expects the entropy of the coefficients to decrease monotonically from lower frequency channel to the higher frequency channel. In order to represent the coefficients in a good order, the following two modifications are performed.

First, the basis vectors of the generated non-DCTs are shuffled such that they generate coefficients sorted from higher value to lower value in the diagonal-up direction. During entropy coding, a diagonal only scan is used for the non-DCT generated coefficients in order to align the average entropy in a good order.

Second, based on the observation that the HEVC generated quantized coefficients do not necessarily have coefficient energy monotonically decreasing from low frequency to higher frequency, a smart re-ordering of quantized transform coefficients is proposed in this work. The method first, computes an order O_k per class S_k and is based on the average coefficient entropy for each channel. Then, coefficients within this given class S_k are re-ordered using the above computed order O_k, so that the average entropy of the coefficients within this class is sorted in decreasing order. This re-ordered coefficients and the original residuals are then used to compute the non-DCTs from equation 5.2.

The proposed smart re-ordering method is described in detail in Appendix A, but is already used in this chapter when generating the non-DCTs and the results presented here take into account this re-ordering. The unitary gains due to this modification are presented later in Appendix A for completeness.

5.2.2.2 Associated Syntax Encoding

The implementation on the encoder side involves selection of the best transform candidate for each of the residual blocks of size 8×8 and 4×4. This makes it mandatory to encode the transform index for each transform block (TU) having non-zero values. A 1-bit flag is encoded to indicate the usage of DCT or non-DCT for a particular residual block. In case a non-DCT flag is set, the non-DCT is indexed using additional $\log_2(K)$ bits where K is the number of non-DCT transforms. A context is attached to the flag indicating DCT and non-DCT. No index is required for TU size greater than 8×8.

It is further observed that all the transforms in the set are not equally used. This is illustrated in Figure 5.4 which shows the distribution of transform usage for K =17 transforms. Therefore, the rate associated with the encoding of transform index can be further reduced by assigning a probability model. A probability table ($\mathbf{P(k)} = [p_1, p_2, \cdots, p_K]$) is defined for the online learning starting with an initial uniform probability. After the first iteration, each element of $\mathbf{P(k)}$ is defined as

$$p_k = \frac{m_k}{M_{N\times N}}$$

where m_k is the number of occurrences of the transform index k and $M_{N\times N}$ defines the total number of blocks of size N×N. Finally, the rate of the transform index is estimated as

$$R_T = -\log_2(\mathbf{p}_k).$$

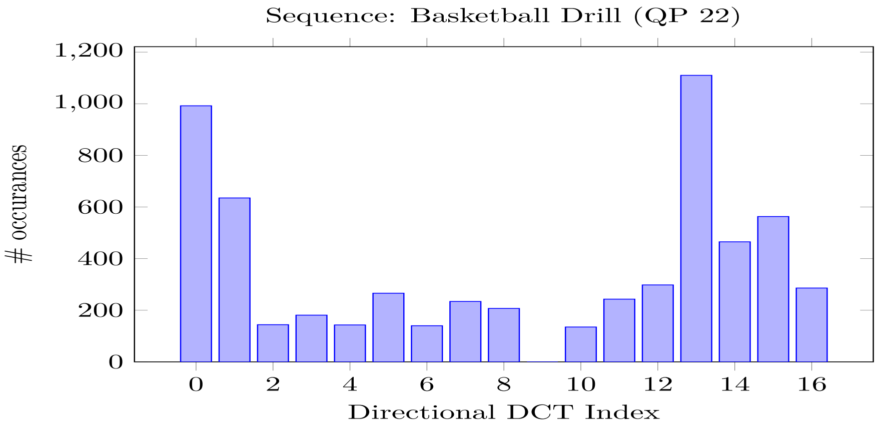


Figure 5.4: Selection of different transform bases in a training set

After each iteration, the probability table is updated and is used for the selection of the best transform candidate inside the rate-distortion search operation of the encoder. This approach is advantageous in rejecting the transforms that are not significantly used during the optimization process and therefore, saving the cost of signaling these transform basis vectors to the decoder. However, the downside of this approach is that the least used transforms are penalized with a high index cost right from first iteration without giving a chance to these least used transforms to learn on the data.

A novel **annealing-like approach** is introduced in this work to avoid the early rejection of least used transforms from the iterative learning process. In order to accomplish this, a parameter ϵ (referred as annealing parameter in this work) is introduced inside the classification step of the above described transform learning scheme and the equation 5.1 is modified and is shown in equation 5.3 where the term III of the equation 5.1 is multiplied with the annealing parameter ϵ.

$$\text{label}\{r^j\} = \arg\min_k \left\{ \underbrace{\|r^j - T_k Q^{-1}(c_k^j)\|_2^2}_{\text{I}} + \lambda(\underbrace{R(c_k^j)}_{\text{II}} + \underbrace{\epsilon R_T}_{\text{III}}) \right\} \tag{5.3}$$

The value of ϵ is increased gradually from the initial value 0 to the final value 1 at each iteration. At start, the rate of index coding R_T is strongly penalized by the annealing parameter ϵ in order to allow the transforms to learn without constraint of index coding and, gradually, the rate of index is introduced into the optimization process. This provides stabilization in the learning process, thus leading to better performance. The effect of the annealing is demonstrated in section on simulation results.

5.2.2.3 Transform Coding

The online learning algorithm introduced above requires transmitting the basis vectors of the newly learned non-DCTs to the decoder side. This accounts for an additional overhead which affects the final performance gain achieved by utilizing these adaptive transforms. In a scenario where all the basis vectors of a non-DCT transform need to be encoded, the overhead of transmitting K 64×64 transforms and L 16×16 transforms at precision of 10 bits can be estimated as $16 \times 16 \times 10 \times L + 64 \times 64 \times 10 \times K$ bits. For K = L = 4, this is estimated to be around 170 kilo-bits. In this work, two methods are proposed to significantly reduce this overhead.

The first method, referred as **Adaptive Incomplete Transforms (AIT)** hereafter, exploits the energy compaction properties of Singular Value Decomposition (SVD) and retains only the first n basis vectors of each transform. The value of n is adaptively computed per sequence and QP and is related to the energy compaction of a given transform. The details of this method are described in Chapter 6.

The second proposed method, referred as **Adaptive Incomplete Quantized Transform (AIQT)** hereafter, quantizes the n chosen basis vectors of a given transform where the quantization step size depends on the quantization parameter (QP) of the frame and the content itself. A statistical model is derived to determine the optimal value of precision b for each transform of each sequence such that

$$b < log_2(a_1(\Delta_q)^{a_2})$$

where Δ_q is the quantization step and a_1, a_2 are the model parameters derived for each sequence during the learning process. The details of this method are also described in Chapter 6.

Finally, the AIQTs are entropy coded in the bit-stream. The overhead of signaling AIQTs is considered while presenting the results.

5.2.3 Simulation Results

In this section, the results of several simulations are presented to demonstrate the performance of adaptive transform learning scheme in terms of overall BD-rate gain. Additionally, the impact of transform coding on the overall BD-rate performance is also illustrated. We implemented the proposed online transform learning scheme in HEVC test model (HM15.0) with the following encoder modifications:

1. $K = 4$ non-DCTs of size 64×64 and $L = 4$ non-DCTs of size 16×16 are learned on the first frame of each sequence for 200 iterations and are tested in competition with conventional HEVC transforms inside the RDO loop,
2. 1-bit flag is encoded to indicate the usage of DCT or non-DCT for a particular residual block. In case a non-DCT flag is set, the best transform candidate is indexed using additional $log_2(K)$ bits,
3. changes are applied only to the LUMA component, and
4. coefficients obtained from non-DCTs are reordered such that they are always scanned in the diagonal direction irrespective of the HEVC adaptive scanning in order to utilize HEVC CABAC context for coefficient coding.

In the first set of experiments, the BD-rate gains are reported only on the first frame of various HEVC sequences as defined in the Common Test Conditions (CTC) of HEVC [78]. Also, the overhead of signaling the basis vectors is not included, at first, when presenting the results in order to illustrate the effect of different improvements proposed in this work. The affect of the overhead will be presented later.

The first experiment is designed to illustrate the effect of different initializations of the learning scheme on the final BD-rate gain. Table 5.1 illustrate this effect in case of the KLT-based and the oriented DCT-based initialization as discussed in sub-section 5.2.1. It is observed that the KLT-based initialization shows better final BD-rate gains. This is expected as KLT-based initial

Table 5.1: BD-rate gain on first frames for different initializations (without basis vector signaling overhead)

Class	Sequences	KLT-based	Oriented DCT-based
A	Nebuta	-4.47	-2.65
	PeopleOnStreet	-5.05	-3.80
	SteamLocomotive	-4.18	-3.05
	Traffic	-3.35	-3.03
	Overall	-4.26	-3.13
B	BasketballDrive	-2.77	-2.01
	BQTerrace	-2.66	-2.26
	Cactus	-3.01	-2.15
	Kimono	-0.33	-0.42
	ParkScene	-3.92	-2.73
	Overall	-2.54	-1.92
C	BasketballDrill	-9.51	-9.44
	BQMall	-4.12	-3.15
	PartyScene	-4.37	-3.76
	RaceHorses	-6.14	-3.49
	Overall	-6.04	-4.96
D	BasketballPass	-7.15	-5.13
	BlowingBubbles	-7.06	-5.57
	BQSquare	-6.07	-5.61
	RaceHorses	-7.60	-4.05
	Overall	-6.97	-5.09
E	FourPeople	-3.15	-2.45
	Johnny	-1.75	-1.19
	KristenAndSara	-2.66	-2.15
	Overall	-2.52	-1.93
Overall		-4.47	-3.41

transforms are learned on the frame itself and proves to be a better starting point. This also proves that the initialization can significantly affect the final gains.

The second experiment is performed to illustrate the result in three cases: 1) without annealing (denoted as AT), 2) with annealing as proposed in 5.2.2.2 (denoted as AAT) and, 3) with annealing and smart-reordering as proposed in 5.2.2.1 (denoted as AAT + re-order). The KLT-based initialization is used for illustrating the gains.

Table 5.2 shows the BD-rate gain for all three cases. In case of AT, the cost of transform index (R_T) in equation 5.1 is kept zero during the learning process in order to avoid any implicit restriction on transform learning. For AAT, the annealing parameter ϵ is applied to the R_T as shown in equation 5.3 where the value of ϵ is increased from 0 to 1 uniformly for 100 iterations. Results show that a simple annealing strategy can provide better gain (up to 0.7% in case of RaceHorses as shown in 5.2). This proves the merits of jointly optimizing the coefficient encoding cost and the index encoding cost.

The results further show that the smart-reordering method improves the overall BD-rate gain by

Table 5.2: BD-rate gain on first frames to illustrate the effect of annealing and smart-reordering (without overhead)

Class	Sequences	AT	AAT	AAT+re-order
A	Nebuta	-4.47	-4.41	-4.39
	PeopleOnStreet	-5.05	-5.14	-5.10
	SteamLocomotive	-4.18	-4.07	-4.04
	Traffic	-3.35	-3.32	-3.34
	Overall	-4.26	-4.23	-4.22
B	BasketballDrive	-2.77	-2.88	-3.56
	BQTerrace	-2.66	-2.73	-2.87
	Cactus	-3.01	-3.05	-3.16
	Kimono	-0.33	-0.12	-0.43
	ParkScene	-3.92	-4.34	-4.32
	Overall	-2.54	-2.62	-2.87
C	BasketballDrill	-9.51	-9.53	-11.12
	BQMall	-4.12	-4.39	-4.62
	PartyScene	-4.37	-4.28	-4.80
	RaceHorses	-6.14	-6.64	-6.79
	Overall	-6.04	-6.21	-6.83
D	BasketballPass	-7.15	-7.36	-8.22
	BlowingBubbles	-7.06	-7.64	-8.16
	BQSquare	-6.07	-6.03	-6.38
	RaceHorses	-7.60	-8.30	-8.73
	Overall	-6.97	-7.33	-7.87
E	FourPeople	-3.15	-3.10	-3.38
	Johnny	-1.75	-2.41	-2.27
	KristenAndSara	-2.66	-3.42	-3.10
	Overall	-2.52	-2.98	-2.92
Overall		-4.47	-4.68	-4.94

0.3% over the annealing based method (AAT) with a maximum gain of 1.6% for the BasketballDrill sequence. Here, the results of an entropy-based re-ordering are presented. However, other methods of re-ordering are proposed in this work and their details are covered in Chapter 6.

It is also observed that the sequences with high directional structure (such as Basketball Drill) show significant gain of over 11% which proves that carefully adapting to the content of a sequence using the proposed online learning approach can provide much better BD-rate gains. The transforms adapt to the local directional structures present in the residual domain of a frame and if these structures occur quite often in a frame or a sequence, these learned transforms are able to model these statistics in a better way.

So far, the full overhead of signaling the basis vectors was not considered when presenting the BD-rate gains. As the overhead takes a significant portion of the bits required to encode a single frame, the gain of around 5% achieved using above methods is not enough to compensate for the overhead of signaling the basis vectors per frame. Actually, the on-line method performs worse than HEVC. Therefore, this is resolved by using the same set of transforms, that are learned on the first frame, to encode the rest of the frames of a given sequence. This considerably reduces

Table 5.3: BD rate gain with and without overhead for AI configuration.

Class	Sequences	With Overhead AAT+reorder (AI)	Without overhead AAT+reorder (AI)
A	Nebuta	-2.87	-2.89
	PeopleOnStreet	-4.77	-4.87
	SteamLocomotive	-1.18	-1.31
	Traffic	-2.92	-3.03
	Overall	-2.94	-3.02
B	BasketballDrive	-0.52	-0.62
	BQTerrace	-1.63	-1.66
	Cactus	-1.67	-1.73
	Kimono	0.21	-0.06
	ParkScene	-2.58	-2.70
	Overall	-1.24	-1.36
C	BasketballDrill	-8.37	-8.63
	BQMall	-0.14	-0.36
	PartyScene	-0.96	-1.07
	RaceHorses	-0.43	-0.79
	Overall	-2.48	-2.71
D	BasketballPass	0.38	-0.58
	BlowingBubbles	-0.51	-0.94
	BQSquare	-0.92	-1.25
	RaceHorses	0.63	-0.54
	Overall	-0.10	-0.83
E	FourPeople	-2.07	-2.23
	Johnny	-1.26	-1.54
	KristenAndSara	-1.74	-1.97
	Overall	-1.69	-1.91
	Overall	**-1.69**	**-1.97**

the effect of the overhead of signaling the transforms. Table 5.3 shows the BD-rate gain under All-Intra (AI) configuration where all the frames of a sequence are intra-coded using the newly learned transforms and the basis vectors are coded only once per sequence. The sequence length is set in accordance to the CTC [78] of HEVC. A coding gain is observed for most of the sequences with an average gain of 1.69% as shown in Table 5.3.

Table 5.3 further demonstrates the gain achieved without the overhead for AI configuration. It is observed that the overhead has a slight impact of 0.3% on the coding performance. However, as expected, the impact on the smaller resolution sequences (Class C and D) is higher compared to the higher resolution sequences (Class A and B) as seen in Table 5.3.

The comparison of coding gain on the first frame without overhead (Table 5.2) and on All-Intra without overhead (Table 5.3) shows that for some sequences (such as PeopleOnStreet, Traffic, Basketball-Drill etc.) the gain is comparable but for others, there is a significant drop in the gain. In general, the drop is higher in case of lower resolution sequences than higher resolution sequences. Figure 5.5 shows the BD-rate gain for two different sequences measured at subsequent intra-frames.

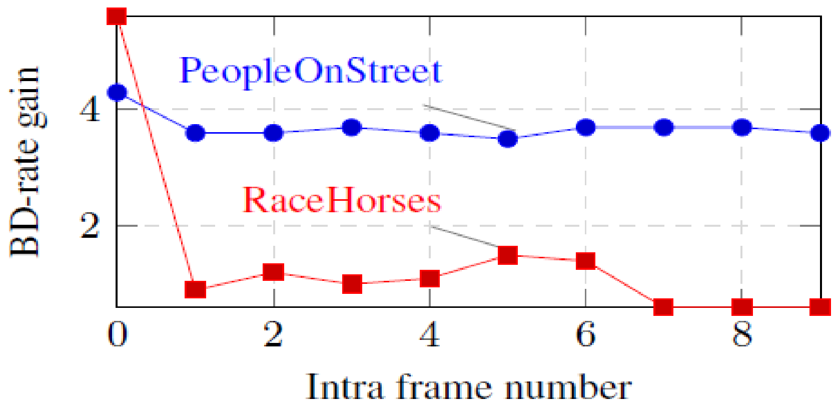


Figure 5.5: BD-rate gain (in %, relatively to HEVC) at each intra frame

The reason behind this could be the over-fitting of the transforms on the residuals obtained from the first frame. The small number of residual examples to learn on may lead to over-learning which is sub-optimal when considering the full sequence. This over-fitting does not occur in case of higher resolution sequences. The over-fitting may be avoided by learning on multiple frames instead of a single frame, but this would lead to a high latency and an unpractical complexity.

The second reason is linked to the correlation between the residual statistics in the first frame and the residual statistics in the remaining frames. The motivation behind using the same set of transforms for the whole sequence is that a high amount of correlation exists within the frames of the same sequence. However, if the correlation does not exist or if there is a scene change, the gain may be significantly lower. For sequences with low correlation, performance drops comparatively faster when this set of adaptive transforms are applied on the subsequent intra-frames of a sequence. However, for sequences such as PeopleonStreets and Basketball-Drill, the different frames have higher correlation and therefore, the BD-rate gain is retained across different frames.

Finally, Table 5.4 compares the BD-rate gains when the initial KLTs are directly used to code the whole sequence without learning to the the BD-rate gain when learned KLTs are used. It is clear that the gain using initial KLTs is much less compared to the gain achieved using learned KLTs. This proves that the learning of KLTs even on the first frame has an effect on the final BD-rate gain. This is also true for sequences where the residuals of the first frame and the remaining frames are not much correlated.

5.2.4 Discussion and Conclusions

As the conventional DCT is not optimal to fully exploit the directional structure present in the residuals of an intra-picture, a framework has been developed in order to exploit the residual signal statistics present within a given sequence and to adapt the transforms to the content using a block-based learning algorithm. It is observed that these new optimized transforms obtained using an iterative cost-based classification and optimization scheme can provide an average BD-rate gains of around 1.7% over standard HEVC under All-Intra configuration.

Table 5.4: Comparison of BD-rate gains with learning and no learning from initial KLT transforms

Class	Sequences	No learning init=KLT	Learning init=KLT
	Nebuta	-2.07	-2.87
	PeopleOnStreet	-2.00	-4.77
A	SteamLocomotive	-0.64	-1.18
	Traffic	-0.84	-2.92
	Overall	-1.39	-2.94
	BasketballDrive	0.04	-0.52
	BQTerrace	-0.48	-1.63
B	Cactus	-0.61	-1.67
	Kimono	0.26	0.21
	ParkScene	-0.96	-2.58
	Overall	-0.35	-1.24
	BasketballDrill	-0.52	-8.37
	BQMall	0.17	-0.14
C	PartyScene	-0.34	-0.96
	RaceHorses	0.01	-0.43
	Overall	-0.17	-2.48
	BasketballPass	0.89	0.38
	BlowingBubbles	-0.11	-0.51
D	BQSquare	-0.33	-0.92
	RaceHorses	1.05	0.63
	Overall	0.38	-0.10
	FourPeople	-0.33	-2.07
E	Johnny	0.12	-1.26
	KristenAndSara	-0.25	-1.74
	Overall	-0.15	-1.69
Overall		**-0.34**	**-1.69**

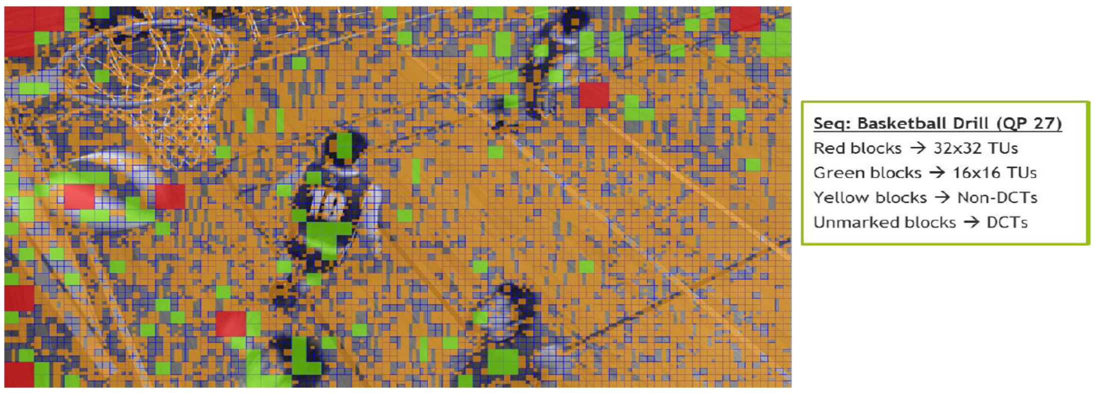


Figure 5.6: Usage of non-DCTs (yellow blocks) for sequence Basketball drill (QP 27)

Table 5.5: Usage Statistics of non-DCTs vs core HEVC transforms for Class C sequences

BasketballDrill		BQMall		PartyScene		RaceHorses	
4×4	8×8	4×4	8×8	4×4	8×8	4×4	8×8
65%	75%	58%	20%	63%	8%	72%	26%

Further it is shown that for some sequences such as Basketball-Drive, a significant gain of 8.4% is achieved over conventional HEVC. Figure 5.6 illustrates the areas of the sequence Basketball Drill, coded at QP 27, where non-DCTs (in yellow) are chosen in place of conventional DCT/DST. It clearly shows that the non-DCTs are extensively chosen across the whole frame. Also, the average usage statistics of non-DCTs vs core HEVC transforms for class C sequences is illustrated in Table 5.5. It shows that for 75% of the time, the newly learned transforms for residuals of size 8×8 are chosen compared to the conventional DCT in case of Basketball-Drill. However, for PartyScene, only 8% of the 8×8 size residuals are coded using non-DCTs.

5.3 Offline Adaptive Transform Learning scheme

This chapter has so far discussed the online learning where the transforms are learned on-the-fly and are required to be signaled to the decoder. It was further shown above that the overhead of signaling the learned transforms is huge and, therefore, cannot be performed per frame. In this section, an offline learning scheme is briefly described and the BD-rate gains of the offline learned transforms is compared to the online learned transforms (with and without the overhead).

In the literature, as described in Chapter 3 and 4, several offline learning methods have been proposed to learn a dictionary of transforms offline on a large training set, and this dictionary is then made available at both encoder and decoder for the coding of the video sequences. This avoids the signaling of basis vectors to the decoder and only the transform index is coded. In order to compare the proposed online learning method against the existing offline learning methods, an offline learning method is described and utilizes the same number of transforms as above in order to make a fair comparison of the gains achieved in both cases.

In order to compare the online learning scheme proposed above, an offline learning scheme, which utilizes an offline learned dictionary of transforms to encode the whole frame, is presented. This set of transforms is learned once for all and is made available to both encoder and decoder. Therefore, the overhead of signaling the transform basis vectors is removed.

The offline learning, in contrast to online learning, relies on a training set which is formed by a large collection of residual blocks taken from various sequences. Therefore, the learned transforms are generic in nature in contrast to online learning where specific transforms are learned per sequence. Chapter 4 has presented different ways of learning data-driven transforms. In this section, first, the chosen learning scheme is described. Then, the implementation details and results on the HEVC codec (HM15.0) are presented. Finally, a detailed analysis on both offline and online learning methods is presented in the consecutive section of this chapter.

5.3.1 Learning scheme

This sub-section presents the learning scheme used to determine an offline set of transforms. Table 5.6 shows the list of training sequences of various resolutions and content types that are

Resolution	Training Sequences (First 40 frames)
CIF	Foreman, Paris
720p	city, crew, panslow, raven, spincalendar
1080p	Mitsubishi Traffic, Pedestrian area, station2, rolling tomatoes, Rush hour, sunflower, table_setting, Vintage car, toy and Calendar, walking couple.

Table 5.6: Training sequences used for off-line learning of transforms

chosen to generate a training set. These training sequences are chosen such that they are independent from the HEVC CTC sequences. As described in chapter 4, the performance of the learning scheme greatly depends on different initialization, classification and transform optimization steps performed to generate the final set of transforms. For the offline learning, the following choices have been made:

1. **Initialization:** KLT-based initialization is chosen where the training set is first randomly split into K different classes and a KLT is learned per class.
2. **Classification:** Rate-Distortion (R-D) cost-based classification as described in 4.1.2 is used and is based on the R-D metric described in equation 4.2 where the rate is approximated by the number of non-zero quantized coefficients.
3. **Transform optimization:** For transform optimization, P in equation 4.3 is set as the $\mathcal{L}_0$-norm and the optimization equation is solved as described in 4.1.3.
4. **Choice of λ:** The value of λ is computed using the close-form solution as described in [36].

The classification and transform optimization steps are repeated until convergence or for a fixed number of iterations. The final transforms are stored at both encoder and decoder as a dictionary.

5.3.2 Implementation in HEVC

Similar to the online learning scheme proposed in the first part of this chapter, the offline learning scheme also learns K non-separable transforms ($T_k^{8\times8}$) of size 64×64 and L non-separable transforms ($T_l^{4\times4}$) of size 16×16 on the offline training set defined in Table 5.6 using the learning scheme described above. The HEVC encoder (HM) is modified such that the offline learned transforms are tested in competition with the core DCT/DST transforms on 8×8 and 4×4 luma residuals only. The chroma residuals and the inter residual blocks, as in online learning, are transformed using the conventional DCT.

Due to the transform competition at the encoder, the best chosen transform per residual block is signaled to the decoder using a transform index. However, in contrast to online learning, the basis vectors of the offline learned transforms are not signaled and are already available at the decoder as fixed transforms. This provides an advantage over online learning in terms of saving the heavy overhead due to transform basis signaling.

The transform index is coded in a similar fashion as done before for online learning where a flag is coded to indicate the use of DCT and non-DCT. If the flag is set, an index is further coded to indicate the chosen non-DCT from a set. A CABAC context is attached to the flag to

Table 5.7: BD-rate gain on the first frame for offline and online case (without overhead).

Class	Sequences	Offline case	Online case
A	Nebuta	-0.81	-4.39
	PeopleOnStreet	-1.64	-5.10
	SteamLocomotive	-0.28	-4.04
	Traffic	-1.88	-3.34
	Overall	-1.15	-4.22
B	BasketballDrive	-0.09	-3.56
	BQTerrace	-1.34	-2.87
	Cactus	-1.34	-3.16
	Kimono	-0.10	-0.43
	ParkScene	-1.75	-4.32
	Overall	-0.92	-2.87
C	BasketballDrill	-0.32	-11.12
	BQMall	-1.12	-4.62
	PartyScene	-1.55	-4.80
	RaceHorses	-1.80	-6.79
	Overall	-1.19	-6.83
D	BasketballPass	-0.48	-8.22
	BlowingBubbles	-0.71	-8.16
	BQSquare	-1.52	-6.38
	RaceHorses	-0.68	-8.73
	Overall	-0.85	-7.87
E	FourPeople	-0.86	-3.38
	Johnny	-0.35	-2.27
	KristenAndSara	-0.81	-3.10
	Overall	-0.67	-2.92
	Overall	-0.96	-4.94

improve its coding.

Similar to online learning, the basis vectors of the learned transforms are shuffled to align the energies of the coefficients from higher value to lower value in a diagonal up direction. Only diagonal scan is utilized when using the non-DCTs.

5.3.3 Simulation Results

The offline learning scheme is implemented in the HEVC test model (HM15.0) as describes above where K=L=4 non-DCTs are learned offline and are made available at encoder and decoder. The number of transforms are kept same as for online learning in order to compare the overall coding gains.

In a first experiment, the results of coding a single frame of a sequence is presented and is compared with the coding gains of online learning (without overhead). Table 5.7 shows the BD-rate gain on the first frame of the HEVC CTC sequences. An overall gain of 0.96% is observed in offline case. It is observed that the coding gains achieved using the offline learning are much lower compared to the online case when using the same number of transform candidates. This is expected as in the online learning scheme, the transforms are learned on a given frame and

Table 5.8: BD-rate gain (AI) for offline and online case (overhead included).

Class	Sequences	Offline (AI)	EncT (offline)	Online (AI)	EncT (online)
A	Nebuta	-0.58	254%	-2.87	624%
	PeopleOnStreet	-1.62		-4.77	
	SteamLocomotive	-0.26		-1.18	
	Traffic	-1.86		-2.92	
	Overall	-1.08		-2.94	
B	BasketballDrive	-0.49	253%	-0.52	436%
	BQTerrace	-1.25		-1.63	
	Cactus	-1.22		-1.67	
	Kimono	-0.44		0.21	
	ParkScene	-1.88		-2.58	
	Overall	-1.06		-1.24	
C	BasketballDrill	-0.63	289%	-8.37	419%
	BQMall	-1.22		-0.14	
	PartyScene	-1.62		-0.96	
	RaceHorses	-1.37		-0.43	
	Overall	-1.21		-2.48	
D	BasketballPass	-1.06	296%	0.38	410%
	BlowingBubbles	-1.24		-0.51	
	BQSquare	-1.62		-0.92	
	RaceHorses	-1.05		0.63	
	Overall	-1.24		-0.10	
E	FourPeople	-0.64	270%	-2.07	376%
	Johnny	-0.43		-1.26	
	KristenAndSara	-0.43		-1.74	
	Overall	-0.50		-1.69	
	Overall	**-1.02**	**271%**	**-1.69**	**449%**

therefore, are able to exploit the content specific correlation in a better way compared to the offline transforms. However, the coding gains of online learning presented here do not take into account the overhead of transmitting the basis vector.

In a next experiment, the BD-rate gain on the whole sequence under All-Intra (AI) encoder configuration is computed for the offline case. Table 5.8 shows the BD-rate gains for both offline and online learning where the gain presented for the online case takes into account the overhead of signaling the basis vectors. The offline learning provides an overall gain of 1% which is less than the gain achieved using online learning for the whole sequence under AI configuration in case the same number of transforms are used. This shows the merits of online learning in providing better adaptation to the content and therefore, achieving better compression efficiency. Table 5.8 also compares the encoder complexity (EncT) of both offline and online learning. It is shown that the online learning scheme is 180% more complex than the offline learning scheme. This is expected as in the online case, the learning is performed on the first frame which leads to the higher complexity. For other frames, the complexity of offline and online is comparable as the same number of transforms is being used.

Finally, it is observed that the BD-rate gains in case of offline learning are somewhat more stable than the gains achieved in case of online learning. Table 5.8 shows that for offline case, gains are observed for all the sequences whereas in online case, significant gains are observed for some sequences whereas for others a slight loss is observed. The variation of the BD-rate in online case is linked to the correlation between the frames and also the over-fitting on the first frame as described in section 5.2.3. However, in case of offline learning, the learned transforms are generic enough to provide consistent gains for all the sequences and do not favor a particular sequence.

Considering that the two approaches proposed so far in this chapter provide comparable compression performance, the next section further compares the two schemes in terms of the overhead cost, complexity and convergence. The limitations of the online and offline schemes are discussed.

5.4 Comparison between offline and online learning

This chapter has so far presented the online and offline learning schemes and compared the BD-rate gains achieved in AI configuration. It has been shown that the learning of a transform set on the first frame leads to interesting gains and are higher than the gains using a transform set learned on an independent training set. It has been further shown that the online learning scheme may provide considerable gain for some sequences but may loose in case of other sequences. However, in general due to the high level of correlation among frames of the same sequence, learning on the first frame is certainly helpful in achieving higher compression gains.

Having said that, there are many limitations posed by the online learning scheme which restricts the scheme to achieve further gains without adding more complexity and more overhead. The limitations of the online framework is discussed below.

5.4.1 Short-comings of the online framework

The online transform learning algorithm proposed in this chapter shows a great potential in terms of achieving better compression efficiency compared to the standard HEVC codec which uses only DCT. However, there are many limitations on the practical implementation of the proposed algorithm which are discussed below.

Overhead Cost

In case of the multiple transform learning schemes discussed in this chapter, the additional information that is signaled to the decoder comprises:

- first, the syntax associated with the transform used (the transform index), and
- second, the overhead of the basis vectors of the learned transform (only in case of online learning schemes).

Table 5.9 shows the effect of the first cost associated with the transform used (the transform index) on the BD-rate gain in both online and offline cases. It shows that the signaling takes nearly 6-7% of the coding gain in both cases.

Further, the online method requires signaling of the basis vectors of the learned transforms. This overhead is significant when compared to the bits required to encode a single frame. In this thesis, two methods are proposed to reduce the overhead size of the basis vectors significantly (between 65-75%). These methods are described in detail in Chapter 6. However, even after the proposed reduction, the overhead still takes a large portion of the bit-stream especially in case of low-resolution sequences (i.e. classes C and D). Figure 5.7 shows the average overhead

Table 5.9: Comparison of coding gains with and without transform index signaling.

Learning method	**BD-rate**	
	No Signaling	**With Signaling**
Offline	-7.84%	-1.02%
Online	-8.25%	-1.69%

size as a percentage of the total bits required to encode a single frame for different class of video sequences after the proposed overhead size reduction.

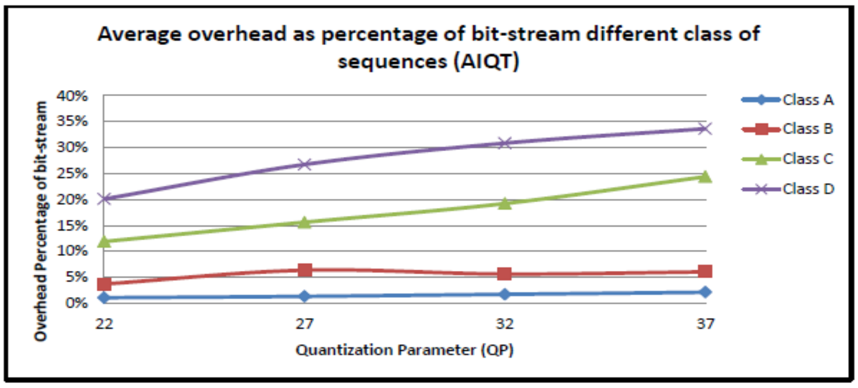


Figure 5.7: Average Overhead for each Class at each QP

As the offline methods do not have this limitation, the number of transforms can be further increased. This is done in several recently proposed schemes such as [39] where multiple transforms are learned per intra-prediction direction and provide better coding gains. On the other hand, in online learning, the increase of the number of transforms comes at a cost of further increase of the overhead.

Convergence of coding gains

For the online case, the iterative learning scheme shows poor convergence when learned on the residuals extracted from a single frame. As seen in Figure 5.1, at each iteration, the computed transforms are fed into an HEVC RDO loop that computes the quad-tree (QT) and the residual quad-tree (RQT) using all the possible combinations of the CU and TU sizes. This leads to a change in the residual set on which the previous transform set was learned and therefore, leads to a non linear convergence of the algorithm.

The proposed annealing scheme and smart-reordering scheme help in improving the overall stability and the performance of the iterative algorithm. But the lack of convergence negatively impacts potential gains even though how much a good-converging iterative algorithm could give us is still an open question.

The offline methods, on the other hand, show good convergence as the underlying residual training set is fixed. However, the higher convergence on a training set does not relate to the higher coding gains on the test sequences. This relationship is usually difficult to anticipate.

Complexity

The proposed online scheme shows that the heavy adaptation of the transforms to one frame can lead to interesting compression gains especially when the overhead is not considered. Of course, this leads to impractical encoding schemes that induce a large number of encoding passes but it exhibits some potential: quite a lot of correlation still exists in the residual blocks. One may argue that pushing further in this direction is worthless as the encoder complexity is simply unrealistic in practical applications. Finally, transform genericity would be the ultimate way to lower the complexity. For example, in the offline case, transforms are learned once on a training set and applied to any content.

Further both online and offline schemes learn a set of non-separable transforms which are costlier to implement in hardware and require large memory to store the transforms. For a size $N \times N$ residual block r, the transform coefficients c can be obtained as follows

$$c = Tr$$

where, T is the non-separable transform matrix of size $N^2 \times N^2$ and the residual block is converted into a 1-D vector of length N^2. This operation can be parallelized to reduce the computational complexity but still requires large storage and computing resources. However, the decoder computing time is observed to be close to the standard HEVC in a software implementation.

Also, the learned transforms are tested inside the HEVC test software in a brute-force fashion which is computationally time consuming. This may be reduced by introducing some speed-ups that analyze the residual blocks at the encoder side in order to avoid test all possible quad-tree and transform configurations.

Finally, even if the encoder complexity is unrealistic for the online approach, the possibility of solid RD gain has been proven.

5.5 Conclusions and Perspectives

As the conventional DCT is not optimal to fully exploit the structures present in the residuals of an intra picture, a rate distortion optimization transform learning algorithm is proposed in this work where the classification is performed inside the HEVC codec and the new optimized transforms are generated based on an iterative approach. The method is tested on the HEVC test software (HM 15.0) and experimental results show that this method can bring a average gain of 1.69% and up to 8% in case of Basketball Drill sequence.

However, the proposed online method is computationally complex and exhibit convergence problem as described in the section 5.4.1. But the gains achieved compared to the conventional HEVC using adaptive sets of transforms show the potential in achieving higher compression efficiency.

5.5.1 Perspectives

The proposed iterative algorithm presented in this chapter provides a basis to develop and test new strategies with a goal to achieve higher compression efficiency and lower complexity. It has turned out that the final performance of any scheme heavily depends on three major factors namely 1) learning algorithm, 2) signaling of the overhead, and 3) selection of transforms at block-level. Following is the list of methods explored in this thesis that are detailed in the subsequent chapters.

Improved offline Transform Learning Schemes

Certainly, offline learning provides an advantage in terms of reduced complexity and zero overhead bits required for signaling the basis vectors. Therefore, the work in this thesis will explore new offline strategies to obtain optimized transforms and to try to improve the existing techniques. This can be done by improving the classification of residual blocks which so far is done using intra-prediction direction and ℓ_0-norm based cost. Some of the techniques developed for online learning scheme such as annealing and CABAC friendly re-ordering can also be used to improve the offline learning scheme and to obtain better sets of adaptive transforms.

Content Adapted vs Pool of Transforms

Chapter 8 of this thesis will explore the possibility to use an offline learning process to determine a pool of transforms and also explores the strategies to perform transform selection from the pool for a given region.

Improved Transform Index Encoding

As shown above, the cost of coding the transform index has considerable impact on the overall BD-rate gain and, therefore, methods to reduce this index cost are explored in this thesis.

- **Partial inference of transform index:** Methods have already employed full inference to avoid signaling the transform choice to the decoder and the transform index is derived from some causal information such as intra-prediction direction. Additionally, the transform index can be partially inferred from the neighboring blocks or any other causal information and this partial inference can then be used to design a CABAC context. This method is described in detail in Chapter 6.
- **Exploiting the hierarchical signaling:** Similar to the hybrid indexing scheme proposed in [1] which involves both CU-level and TU-level indexing, the transform index is split hierarchically into a region-level index and a sub-region level index. This is detailed in Chapter 8 where pool-based transforms are described. This provides a way to automatically adjust the signaling based on the complexity of the region of the video frame and therefore, helps in reducing the side information.

In the next chapter, the proposed methods to improve the signaling of transforms and transform index are presented.

6 Advanced signaling of transform and transform index

6.1 Goal of this chapter

Chapter 5 has described the proposed online learning scheme in detail. Due to the non-linear nature of the learning scheme, the stability was hard to obtain. Further, it was shown that the overhead affects the coding gains significantly. Some methods are proposed to stabilize the learning scheme and to improve the convergence. Annealing-like approach (as described in section 5.2.2.2) and the smart reordering approach (as described in section 5.2.2.1) are shown to improve the coding gains.

As it was shown in the previous chapter that the overhead due to uncompressed transform set is significantly high, the first part of this chapter is dedicated to efficiently code the basis vectors in the bit-stream. In the second part, a novel method to reduce the overhead of signaling the transform index is described which can be applied to both on-line and off-line learning scheme.

6.2 Proposed methods for signaling the overhead

As described in Chapter 5, there are two different overheads that are required to be signaled to the decoder in case of online learning scheme. The first is the signaling of the transform index chosen for each residual block, and the second is the signaling of the basis vectors of the learned transforms. In case of offline learning, the signaling of the basis vectors is not required and hence, the first part of overhead is removed which is advantageous in scaling up the number of transforms that can be learned offline and tested to achieve higher compression gain.

The signaling of the transform index is also very high as shown in Table 5.9 for both online and offline cases. This transform index is usually hard to predict from the neighboring blocks as the neighboring residual blocks have very little correlation. Moreover, the RDO process choses the best transform per block after testing all the possible mode decisions making it even harder to predict the index.

In this section, the proposed transform basis vector coding scheme is presented. It utilizes the properties of the learning scheme and the lossy codec (in this case HEVC) to reduce the overhead considerably. Then, a novel transform index prediction method is presented. It uses the causal information coded in the bit-stream to determine the transform index. The method shows some slight coding gains on top of the existing approach.

6.3 Scheme for coding of transform basis vectors

As already discussed in Chapter 5, the online learning algorithm requires transmitting the newly learned transform basis vectors to the decoder side. This accounts for an additional overhead which affects the final performance gain achieved by utilizing these adaptive transforms. If one does not take care on how to send the vectors, the global bit-rate increases, despite the better compaction, and the adaptive transforms are of no benefit.

One method is to determine the transform every F frames so that the effect of the overhead is reduced by a factor F. Chapter 5 has demonstrates this by coding the transform set once per sequence which is 10 seconds long. An overall gain of 1.69% is observed with some sequences. However, for other sequences, only a slight gain is observed. In this case, full basis vectors of the transform were coded.

In this section, methods to efficiently encode the transform set into the bit-stream are presented. First, the prior art on basis vector coding is briefly discussed. Next, proposed methods to reduce the signaling cost of transforms are presented, as well as the simulation results.

6.3.1 Prior art on basis vector coding

Due to the heavy overhead cost of transform coding as detailed in last section, it is important to efficiently encode transform vectors into the bit-stream. However, the idea to encode the basis vectors in the bit-stream is not new. Singular Value Decomposition (SVD)-based image coders have been proposed where, at the encoder side, the SVD vectors are computed for each sub-block and are vector quantized (VQ) using variable bit-rate coding schemes before coding into the bit-stream. The SVD values are however scalar quantized using Max quantizer [79]. In another scheme [73], the combination of SVD and VQ is proposed in order to reduce the complexity of computing the SVD. The basic idea behind these image coders is that the SVD has the optimal energy compaction properties and, therefore, a block decomposed into its corresponding singular values and singular vectors is approximated by its first k singular values and singular vectors out of the total n. The minimization of the error is guaranteed for a given value of k in case of SVD.

The work in [73] and [79] is focused on image coding and has the following drawbacks

- the method is specifically designed for block based image codec for low bit-rate coding applications,
- it requires offline computation of codebooks of various sizes for vector quantization, and also requires offline computation of max quantizers for each singular value,
- the codebook search for vector coding is computationally demanding.

The proposed method in this chapter takes into account the sum of the energies of quantized coefficient at each coefficient position for determining the threshold used for computing the number of transform vectors to encode. Moreover, a scalar quantization of transform vectors is used in contrast to vector quantization as done in prior art. Also, a suitable precision value in terms of bits is determined for each transform and is signaled to the decoder in the bit-stream along with the quantized transforms.

A similar problem of coding the basis has also been studies in computer graphics in the context of PCA based dynamic mesh compression [80] technique. Usually, the trajectory of V vertices across different frames of a computer graphic sequence can be represented in the form of a matrix M of size 3F×V where F is the number of frames, and the factor 3 denotes 3 dimensions, and V is the number of vertices. This matrix is known to be highly correlated and therefore, PCA is used to de-correlate the matrix M where only first N coefficients are transmitted depending on the user defined quantization value Q. However, the signaling of PCA basis vectors in this case is required which consumes considerable overhead bits and impacts the performance of the system. In order to overcome this problem, the author proposed to use non-least square optimal linear prediction and non-uniform quantization of basis vectors for PCA to achieve higher compression.

The use of non-least square optimal linear prediction in [80] is based on the observation that the basis vectors retain the characteristics of the trajectories and therefore, the variation in the values of basis vectors is smooth and, therefore, a linear or non-linear predictive coding can be applied to predict values inside a basis vector using the previous samples of basis values.

Additionally, a non-uniform quantization of PCA basis vectors has been used to achieve further compression. This is based on the observation that the first coefficients obtained after PCA have higher magnitude compared to the second coefficient and so on. This is due to the energy compaction property of PCA. Following this observation, the author in [80] proposes to utilize finer quantization step size for coding the basis vector corresponding to the first coefficient and slowly increase the step size as one moves from the first towards Nth basis vector. A user defined quantization value Q (which can be seen as QP in case of video compression) is utilized to determine the quantization value Q_i of the i^{th} basis vector as shown in equation below:

$$Q_i = \frac{Q \cdot S}{(N+1) \cdot s_i}$$

where, $S = \sum_{i=1\cdots N} s_i$ and $s_i = \sum_{v=1\cdots V\|c_i^v\|}$. Here, c_i^v is the coefficient value at vertex v corresponding to the ith basis vector and s_i is the sum of the coefficient values corresponding to the ith basis vector across all the vertices. Therefore, as above, the quantization constant is inversely proportional to the sum of coefficients.

In our proposed method, a quantization constant derived for each basis vector inside a transform is based on a content-adaptive statistical model which relies on the variance of quantization error of transformed coefficients in contrast to the sum of coefficient as done in [80]. The proposed content-adaptive statistical model provides an upper bound of quantization constant for a transform vector without impacting much the overall performance.

Also, the method in [80] does not exploit the properties of orthogonal transforms fully which could lead to further compression. The next section will detail the different methods proposed in the context of coding of basis vectors of non-separable orthogonal transforms obtained after the online learning using SVD.

6.3.2 Proposed methods

Transform Vector Coding using Incomplete Representation

Due to the energy compaction property of SVD, it is observed that the overhead cost can be considerably reduced by deducing an incomplete representation of the learned transform, where, only the first k vectors are transmitted to the decoder and the remaining (n - k) transform vectors are either generated using a completion algorithm similar to the Gram-Schmidt method or forced to zero. The completion method is performed both at the encoder and decoder sides. One such method is described in [76].

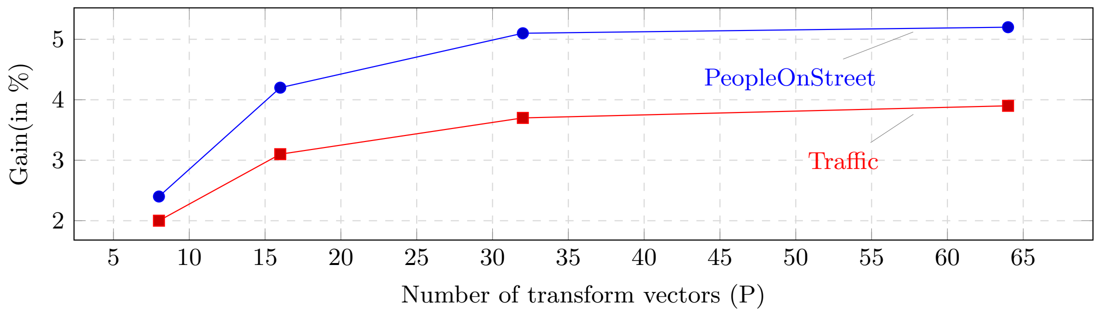


Figure 6.1: BD-rate gain as a function of the number of transform vectors

In order to illustrate the effect on the BD-rate of dropping the last few vectors of a transform, a transform set is computed on the first frame of a given sequence for 8×8 residual blocks using the online learning scheme proposed in Chapter 5, thus n=64. Only the first $k \leq n$ vectors are retained and the rest of the basis vectors are zeroed out. Figure 6.1 shows the effect on the BD-rate gain for different values of k on two different 4K sequences. The gain is shown only for the first frame and the overhead of signaling the basis vectors is not considered.

It is observed that by retaining just the first half of the transform vectors ($k = 32$) the performance drop is found to be negligible. For $k = 16$, the performance drops by 1% compared to the case when all basis vectors are coded, but the overhead cost of transform vectors reduces to one fourth. However, for $k = 8$, there is considerable loss of performance. This shows that one may find a suitable trade-off between the loss in performance and the number of coded basis vectors k. Also, this trade-off is dependent on the content and the QP value. Intuitively, the value of k is lesser at high QP compared to low QP. Therefore, a method is required to estimate the optimal value of k.

The residual signal energy, on average, is concentrated into the first few coefficients, the DC coefficient has on an average maximum energy, and it decreases as one goes towards the higher frequency coefficients. Therefore, most of the high frequency coefficients are quantized to zero. A simple threshold-based method is applied in order to compute the best value of k which is coded into the bitstream.

Let E be the sum of energy of the transform coefficients. A threshold t is defined by the multiplication of a parameter p with E to get $t = p \cdot E$. The value of k is simply computed, by the encoder, from the number of coefficients with average energy greater than this threshold. The value of p is found empirically by testing on different sequences.

Adaptive Quantization of basis vectors

Clearly, additional bit-rate saving can be obtained by lowering the precision of the learned transforms. Let v(i) be an element of a vector **v** of size N. The precision is defined as the number of bits m required to represent v(i) accurately.

$$0 \leq \mathrm{v}(i) < 2^m$$

The precision of the vector **v** is lowered by b bits by applying a rounding as follows

$$\mathrm{v}'(i) = \mathrm{round}(\mathrm{v}(i)/2^b) \cdot 2^b.$$

Figure 6.2 shows the effect of using the learned transforms with reduced precision on the BD-rate gain for two 4K video sequences. Again, the vertical axis is the BD-rate gain in percentage compared to the anchor (HM15.0) and without taking the transform cost into account.

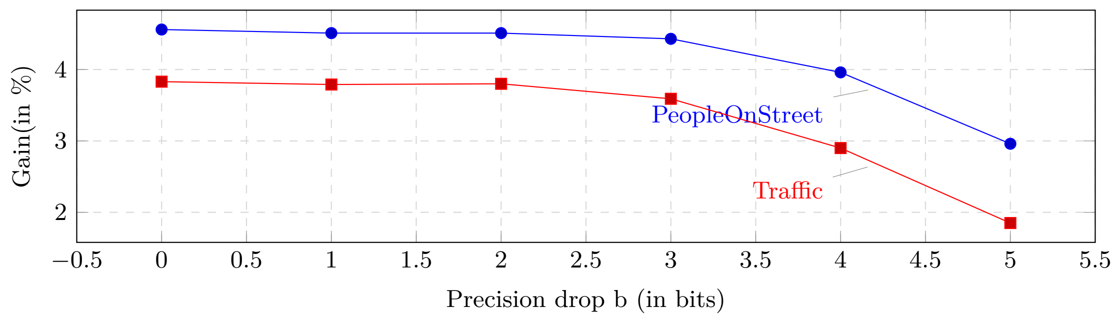


Figure 6.2: BD-rate gain as function of the drop in precision bits

Figure 6.2 shows that lowering the precision by a few bits has a negligible impact on the BD-rate performance. This is beneficial as it reduces the overhead of storing and transmitting the learned transform basis vectors. Figure 6.3 shows the BD-plot to illustrate the effect of precision drop on the performance at different bit-rates for sequences PeopleOnStreet and SteamLocomotive. In Figure 6.3, prec2 refers to a value of b=1 and prec32 refers to a value of b=5. It is observed that the effect of precision drop of adaptive transforms at low bit-rates has a negligible impact on the BD-rate whereas the performance drops significantly at high-rates. Therefore, the parameter b not only depends on the video content but also on the QP parameter, i.e.

$$b = f(\mu, QP),$$

where μ is a factor dependent on the content. A novel method is described below to determine an optimal value of the parameter b for each transform that minimizes the loss in performance due to the drop in precision.

The basic idea behind the content-adaptive statistical model, proposed below, is to find a bound on the error induced by quantization of transform vector such that the variance of the error induced due to the transform quantization is less than the variance of the error induced due to

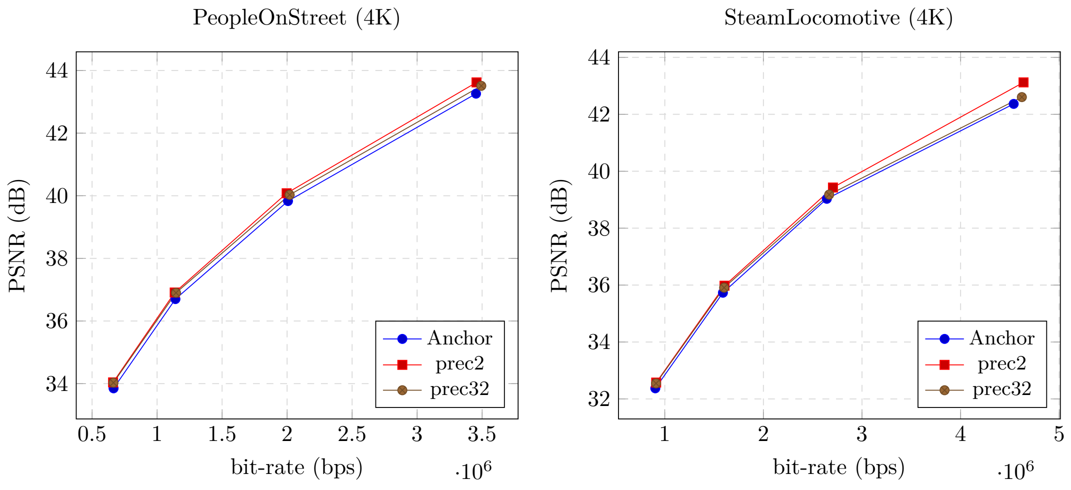


Figure 6.3: Effect of precision drop on performance at different bit-rates

the scalar quantization of transformed coefficients as performed in a video/image codec. Let the variance of error due to quantization of coefficients in a codec be denoted by D(Δ_q) and the variance of error due to the quantization of transform **T** be denoted by D(Δ_g).

The relation between the two variances D(Δ_g) and D(Δ_q) targeted to be kept as

$$D(\Delta_g) < D(\Delta_q) \tag{6.1}$$

A statistical model to determine the optimal value of b for each transform computed for each sequence is deduced in this work and is shown to be

$$b < \log_2\left(\frac{N\cdot\alpha}{\sigma_r^2\cdot\gamma}(\Delta_q)^\beta\right) \tag{6.2}$$

where Δ_q is the quantization step size which depends on QP. Parameters α, β and σ_r^2 are model parameters that are content dependent and parameters γ and N are related to the transform itself. These parameters are derived for each content in order to determine the value of the prediction drop b. Usually, the biggest integer value satisfying the above equation is chosen for b. The derivation of the above equation is presented in the Appendix B.

Huffman coding of the remaining quantized basis values

The first method proposed above exploits the energy compaction property of the transforms and the second method exploits the quantization margin of lossy compression based codec that helps in reducing the precision of the transform. However, both the methods still affects the coding performance even if the drop is not significant.

The remaining basis values are expected to still have some statistical redundancies where values closer to zero are more represented compared to the high values. This motivates to perform a variable length coding instead of a fixed length coding to further gain some bits without effecting the coding gains. A Huffman-coding table is therefore derived in this work which maps the frequently occurring value to a smaller code word and the other values to a larger code word. This Huffman-table is computed by collecting the statistics of the remaining values of the transforms learned on several sequences. Morover, a separate Huffman table is derived for different QP values for better adaptation.

Coding of last coefficient position index for X and Y for partial transforms

Due to the use of partial transforms obtained from dropping the last (n-k) basis vectors, as described in section 6.3.2, one can further improve the coding efficiency of the learning scheme by improving the coding of last significant coefficient position flags inside HEVC. This section describes the proposed improvement. First, the last significant coefficient coding scheme in HEVC is briefly described to understand the proposed modification. Then, the proposed improvement is presented.

Last Significant Coefficient Position Coding in HEVC

The last significant coefficient position is marked as the position of the first non-zero quantized coefficient that occurs when the quantized transformed coefficients are scanned in a given order from highest frequency to the lowest frequency in a coefficient block (CB). Figure 6.4 shows the last significant coefficient position marked in blue inside an 8×8 CB which is divided into 4 coefficient groups (CG). The position is coded in the form of (x,y) coordinates where 0≤(x,y)≤N-1.

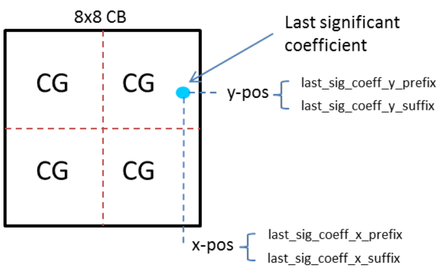


Figure 6.4: Illustration of last significant position inside a coefficient block

As shown in Figure 6.4, each of the coordinates is further split into prefix and suffix. The prefix value and the suffix length (in bits) are shown in Table 6.1 which shows that the prefix and suffix depend on the size of the CB. For a 4×4 TU block, no suffix is coded and the maximum prefix value is 3. For 8×8 TU block, a suffix is only coded for position greater than 3 and the maximum prefix value is 5.

The prefix is binarized using a truncated unary coding. For example, for a 4×4 block, the prefix for position 0, 1, 2 and 3 is coded as 0, 10, 110, 111(truncated). The suffix is binarized using a fixed length coding and is coded only for block sizes greater than 4×4. After binarization, a CABAC is used to entropy code the prefix and suffix.

TU Size	4×4				8×8		16×16		32×32	
Position	0	1	2	3	4-5	6-7	8-11	12-15	16-23	24-31
Prefix	0	1	2	3	4	5	6	7	8	9
Suffix length (bits)	-	-	-	-	1	1	2	2	3	3

Table 6.1: suffix and prefix mapping for a coordinate x or y

Proposed improvement

In the case of partial transforms, only the first m basis vectors are coded and the last (m-n) basis vectors are necessarily zero. Therefore, a residual vector of size n is transformed into m<n coefficients and the last (m-n) coefficients are set to zero. This imposes a constraint on the position of the last coded coefficient and on the values of coordinates (x,y). With m<n transformed coefficients, all the values of (x,y) are not possible and therefore, the solution proposed here is to:

1. restrict the range of possible value of x based on the number of coded basis vectors m, and
2. restrict the range of possible value of y once x is known.

In order to better illustrate the concept, let us consider a 4×4 TU which is transformed using a partial transform P leading to m=9 transform coefficients. As a consequence, one knows that the coefficients 0 to 8 may be non-zero, but the coefficients 9 to 15 are necessarily zero as shown in Figure 6.5a.

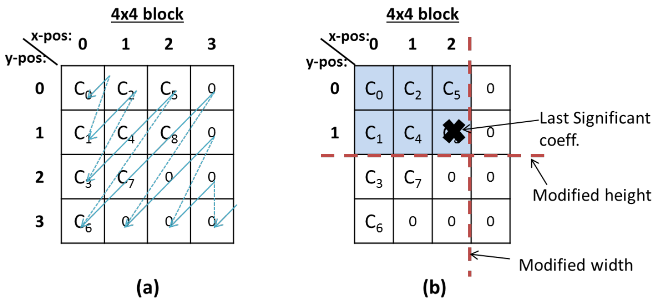


Figure 6.5: Illustrating modification of last significant coefficient position coding

First, when the coefficient block is diagonally scanned, it is shown that the value of x coordinate can never be greater than 2. Therefore, the new block boundary in x-direction can be represented by red vertical line as shown in Figure 6.5b and is marked as "modified width".

Next, for computing the possible range of y, the actual position of the last significant coefficient in x-direction is utilized. In case of the example shown in Figure 6.5:

1. x=0 leads to y in the range [0,3]
2. x=1 leads to y in the range [0,2]
3. x=2 leads to y in the range [0,1]

The modified block boundary in the y-direction in case of x=2 is represented by the red horizontal line as shown in Figure 6.5b and is marked as "modified height".

Finally, the last significant coefficient position is coded according to the modified block size. For the last significant position as marked by a cross in Figure 6.5b, x=2 (resp. y=1) are coded as 11 (resp. 1) compared to 110 (resp. 10) previously, thus saving 2 bits. In short, the truncation of prefix coding is performed at the modified block boundaries compared to the original block boundaries.

6.3.3 Simulation Results

In order to quantify the improvement due to the above modifications proposed to efficiently code the basis vectors, the online transform learning scheme proposed in Chapter 5 is implemented in HEVC test software (HM 15.0) where first, 4 non-separable transforms are learned on each set of 8×8 and 4×4 blocks of the first frame and then used for coding the same frame using transform competition between the learned transforms and core HEVC transforms inside a RDO loop.

Algorithms are tested on Class A HEVC test sequences and the performance is measured in terms of overall BD-rate gain with and without transform coding cost for the following cases:

1. **Adaptive (uncompressed) transforms (ATs)** which denotes the coding gains for using an unmodified set of transforms. The overhead in this case is significantly high.
2. **Adaptive Incomplete Transforms (AITs)** where only first k basis vectors of a transform are retained. The value of k is computed per transform as described in section 6.3.2 and the parameter p is empirically found and set to 0.001. The gains from the proposed last significant coefficient coding modification described above are included when presenting results for AIT.
3. **Adaptive Incomplete and Quantized Transforms (AIQT)** where the precision of AITs is further reduced by b bits. The value of b is adaptively determined using equation 6.2.

Table 6.2: Effect of coding quantized and incomplete basis vectors on BD-rate for 4K sequences

Sequence (4K)	AT	AT+Overhead	AIT	AIT+Overhead	AIQT	AIQT+Overhead
PeopleOnStreet	-5.21%	6.73%	-4.60%	-1.00%	-4.04%	-2.18%
Traffic	-3.87%	9.80%	-3.40%	-0.64%	-3.11%	-1.15%
Nebuta	-3.60%	1.77%	-3.10%	-0.40%	-2.91%	-1.41%
SteamLocomotive	-4.15%	5.00%	-3.80%	0.90%	-3.37%	-0.97%
Overall	-4.21%	5.65%	-3.70%	-0.28%	-3.36%	-1.43%

Table 6.2 shows the Bjontegaard Delta (BD) rate gain of testing AT, AIT and AIQT on the first frame of different 4K sequences. When all the transform vectors are coded at full precision (AT + overhead), heavy loss is observed. A gain of 0.3% and 1.43% is obtained using AIT + overhead and AIQT + overhead thanks to reduction of the overhead.

Further, Table 6.3 shows the average reduction of the overhead cost of Adaptive Incomplete and Quantized Transforms (AIQT) compared to the original Adaptive Transforms (AT). At low QP, the overhead on an average is just 28% of the original size and at high QP, the overhead is only 14% of the original overhead coded into the bit-stream.

Table 6.4 further demonstrate the unitary coding gains due to the last significant coefficient coding modifications proposed above. Table shows a 0.3% BD-rate improvement due to this modification.

Table 6.3: Average bits required to code each basis element and average overhead size ratio at each QP

QP	Average bits to code $v'(i)$				Average compression ratio: (AIQT/AT)
	PeopleOnStreet	Traffic	Nebuta	SteamLocomotive	
22	5.5	5	6.1	5.1	28%
27	4.7	4.6	5.4	4.5	21%
32	4	4	4.5	3.75	15%
37	4	4.5	5	3.5	14%

Table 6.4: Effect of modified last significant coefficient coding on the final BD-rate of 4K sequences.

Sequence (4K)	**AIQT**	**AIQT+last Coeff.**
PeopleOnStreet	-1.68%	-2.18%
Traffic	-0.71%	-1.15%
Nebuta	-1.16%	-1.41%
SteamLocomotive	-0.87%	-0.97%
Overall	**-1.10%**	**-1.43%**

In the next section, a novel method is proposed to improve the overhead related to the signaling of the transform index.

6.3.4 Conclusion

In this section, two methods to reduce the overhead of transmitting the transform basis are presented. It is shown that by efficiently encoding the basis vectors, overhead can be reduced considerably which leads to better overall performance of online transform learning scheme.

6.4 Proposed transform index prediction

The multiple transform schemes proposed in the literature as described briefly in Chapter 3, fall into two categories depending on whether or not an additional syntax to indicate the choice of the transform is added to the bitstream. If not, this information is implicitly derived from the causal information available at the decoder side. Otherwise, the index is explicitly encoded in the bitstream. It has been shown in Chapter 5 that this additional syntax has a huge impact on the overall performance of the scheme. In particular, the impact is maximum for small residual blocks of size 4×4 [40].

In order to reduce the overhead of transmitting the syntax, this work aims at predicting the transform index from the causal part of the bit-stream. In the proposed method, the quantized transform coefficient block (TCB) is utilized to predict the index as it is available at the decoder and is parsed before the parsing of the transform index. Predicting the transform index can be seen as a classification problem where each TCB belongs to a class labeled by its transform index.

Deep convolution neural networks (CNNs) have shown remarkable results in classification tasks where the correlation is not easy to model through simple linear models [81]. For this reason, a CNN is utilized to model the correlation between the TCB and its transform index.

Our contribution involves designing a novel CNN-based transform index prediction method that is trained on a large offline dataset containing a large collection of TCBs from different transforms. Also, the trained CNN is implemented inside the video codec to perform the classification, and the prediction from the CNN is used to improve the entropy coding of the syntax.

This section is organized as follows. First some of the recent related works are discussed. Next, the proposed CNN-based transform index prediction method is detailed and the architecture of the trained model is described in the subsequent sub-section. Finally, simulation results are presented.

6.4.1 Related Works

Video compression, in virtually any video codec, is mainly achieved by employing prediction followed by entropy coding of the residual. The prediction is usually computed using the causal information that is available to both encoder and decoder. By designing a better prediction model, one may achieve higher compression. However, it is extremely difficult to construct a model when there is no prominent pattern in the signal. For such cases, neural networks are suspected to be more efficient and, therefore, many interesting works have emerged recently that utilize CNN-based models to extract difficult features or characteristics present inside the data.

Most of the recent works are focused on making early decisions to speed-up the encoding process. A work carried by Yu et. al. in [82] proposes to use CNNs to provide a binary decision on whether to split the block or not by taking into consideration several features extracted from the block. Similar works have been carried in [83, 84] where a CNN is used to make *aposteriori* decision on splitting coding unit (CU) and prediction unit (PU) blocks. Laude et. al. in [9] have shown that one can mimic the intra-prediction mode decision taken by the RDO using a CNN model which is trained on a large training set.

Our work is related to the above works as we utilize CNNs to perform a classification of coding modes. However, in this work, CNNs are used specifically for the transform index prediction. Moreover, the decision of the CNN is used to drive the entropy coder inside the codec. Finally, the quantized coefficients obtained after the transform and quantization step are used as the input features instead of the hand-crafted features as done in some of the related works. The features are extracted using a CNN learning process which is able to model the complex structures present in the coefficient block.

6.4.2 Proposed trained model-based transform index prediction

In this section, we describe our novel CNN-based transform index prediction method which is employed at both encoder and decoder sides. Firstly, the multiple transform competition scheme (MDTC) as proposed in [39] and various indexing schemes proposed in the literature that are considered as baseline for our method, are described. Then, the proposed transform index coding scheme is detailed.

The MDTC scheme proposed in [39] tests N offline learned transform matrices in competition with the core HEVC transform (DCT/DST) inside a RDO loop and selects the transform matrix that provides the minimum RD cost. A transform index is coded to indicate the choice amongst $N+1$ transforms to the decoder for proper reconstruction of the block. This is done by first coding a flag that indicates whether the DCT/DST transform is used or not. If the flag stipulates it is

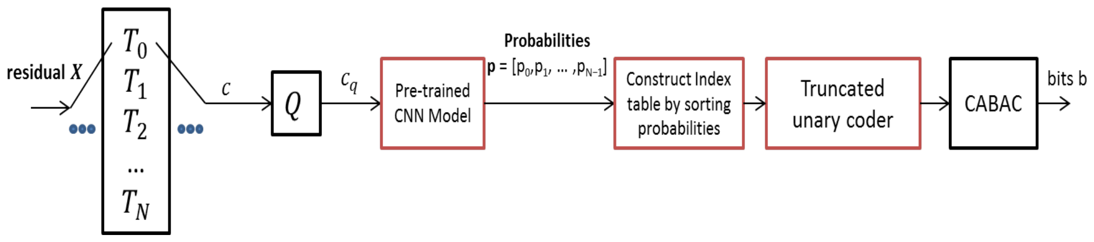


Figure 6.6: Block Diagram of the proposed CNN-based transform index coding

not, the offline learned transforms are used and a fixed length coding is used. The scheme clearly favors the DCT/DST as it requires fewer bits to encode the index signaling.
An alternative way of signaling the transform choice would be to directly binarize the transform index using a fixed length coding, to indicate N+1 transform candidates on b bits where

$$b = ceil(log_2(N+1))$$

These bits are entropy coded using CABAC. This approach does not favor DCT/DST over learned transforms inside the RDO loop. In this work, this indexing scheme is used as a baseline in order to compare the new proposed indexing scheme.

In the proposed scheme, a variable length coding is used instead of a fixed length coding as done in the literature so far. A pre-trained CNN-based model is used to predict the probabilities of a particular transform index which is then used to construct a truncated unary code per block. Figure 6.6 illustrates the block diagram of the CNN-based transform index coding scheme for a 4×4 luma residual block X. The blocks highlighted in red in figure 6.6 shows the modifications over the fixed length coding. Inside the modified HEVC codec [39], the core DST transform (T_0) is tested in competition with offline learned transforms (T_1 to T_N). The CNN-based model is put inside an RDO loop which takes quantized coefficients c_q as input and outputs a vector p of probabilities of predicting a particular transform index i. The vector p is utilized to construct a truncated unary code which is simply done by re-arranging the probabilities p in the decreasing order and using minimum bits (1 bit) for the transform index that is predicted with highest probability and maximum bits (N bits) for the least probable transform index.

In order to understand it better, let us consider N to be equal to 3. The residual X is tested with four transform candidates (T_0 to T_3) and, in each case, the quantized coefficients are passed through the pre-trained CNN model to obtain the probabilities. Let us suppose that T_2 is the selected transform, the residual block is thus transformed with T_2. The quantized coefficients c_q of the transformed block are passed through the trained CNN model which outputs the probability values, say [0.15, 0.1, 0.45, 0.30], of the transforms T_0 to T_3 to be used. Table 6.5 shows the truncated unary code for this example I. In this case, the chosen index 2 is well predicted by

Transform Index	Probabilities	Truncated Unary Code
2	**0.45**	**0**
3	0.30	1 0
0	0.15	1 1 0
1	0.10	1 1 1

Table 6.5: Truncated Unary Code for example I

Transform Index	Probabilities	Truncated Unary Code
2	0.45	0
0	**0.30**	**1 0**
3	0.15	1 1 0
1	0.10	1 1 1

Table 6.6: Truncated Unary Code for example II

the CNN model and, therefore, only 1 bit is coded, namely '0'.

In another example with N=3 and T_0 being the selected transform, the quantized coefficients of the transformed block are passes through the trained CNN model which outputs the probability values, say [0.30, 0.1, 0.45, 0.15]. Table 6.6 shows the truncated unary code for this example. In this case, the chosen index 0 is coded in the bitstream with 2 bits, namely '10'. Finally, the truncated unary code is fed to CABAC in order to entropy code the bits by assigning context to them.

It should be noted that the performance of this approach greatly depends on the classification accuracy of the CNN model that is designed for the task of classifying different transform candidates using the quantized coefficients.

Therefore, the algorithm involves two major steps:

1. offline training of the CNN model per intra prediction direction on a large independent data set as described in section 6.4.3, and
2. applying the trained CNN model inside the HEVC RDO search to predict the transform index as described in this section.

In the next section, the architecture of different layers of the CNN is described in detail along with the method to train the CNN-models for different intra-prediction directions.

6.4.3 Architecture and training of CNN-based model

Deep learning models

In this section, we will describe the CNN-architecture as illustrated in Figure 6.7 for a 4×4 coefficient block. As mentioned in the previous section, the performance of the proposed scheme relies on the classification accuracy of the CNN-models trained offline on a large data set. The selection of training parameters for the model is therefore a critical part in the design of the method.

Table 6.7 summarizes all the model parameters chosen for different CNN layers. The architecture is inspired from Laude et. al. [85], and was adapted to handle smaller input block sizes of a block-based codec. The capacity of the network has been further reduced by using fewer filters and neurons and by using even smaller filter size of 2×2. However, one additional fully connected layer with 36 neurons has been added.

The first convolutional layer takes a coefficient block of size 4×4 as input and is passed through 32 filters of size 2×2 and stride of one. The second convolution layer operates over the output of the first layer and uses 64 filters of size 2×2 and stride of one. A max-pooling layer is used to reduce the size to 2×2×64. This is then fed to the fully connected layers with 36 perceptron. The final softmax layer outputs the probabilities.

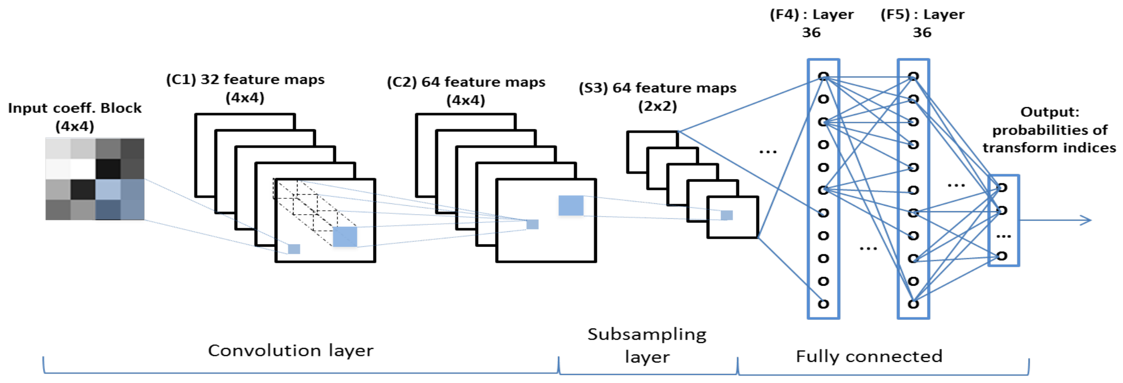

Input coeff. Block (4x4)
(C1) 32 feature maps (4x4)
(C2) 64 feature maps (4x4)
(S3) 64 feature maps (2x2)
(F4) : Layer 36
(F5) : Layer 36
Output: probabilities of transform indices
Convolution layer
Subsampling layer
Fully connected

Figure 6.7: Proposed CNN architecture for the classification of transform on using 4×4 size coefficient blocks

This CNN model is trained using the Keras framework which is a well-known high-level Python neural network library that runs on top of TensorFlow or Theano [86]. Keras is configured to use Theano as the backend. Additionally, the model parameters are optimized by using a gradient-based optimizer called Adam [87] which is available in Keras.

In order to improve the prediction accuracy of the CNN model, the following pre-processings of the training samples are performed.

Layer	Type	Outputs	Filter size	Stride
1	Convolutional C1	32	2x2	1
2	ReLU	-	-	-
3	Convolutional C2	64	2x2	1
4	ReLU	-	-	-
5	Max Pooling S3	64	2x2	2
6	Flatten	256	-	-
7	Fully-connected	36	-	-
8	ReLU	-	-	-
9	Fully-connected	36	-	-
10	ReLU	-	-	-
11	Fully-connected	2	-	-
12	Softmax	-	-	-

Table 6.7: Parameters of the CNN model used for transform index deep learning

– Firstly, training samples have been extracted from an independent dataset [88] which contains images of various buildings.
– only coefficient blocks with atleast 3 non-zero coefficients are considered.
– Finally, imbalanced classes are avoided by manually balancing the number of coefficients in each class.

6.4.4 Simulation Results

The proposed CNN-based transform index coding scheme has been implemented and evaluated in the HEVC test software HM 15.0 which is modified to test multiple transform candidates inside the RDO loop similar to [39]. Two sets of results from various experiments are presented: binary classification (N=1, i.e. DCT/DST vs. one learned transform) and general case (N=3,

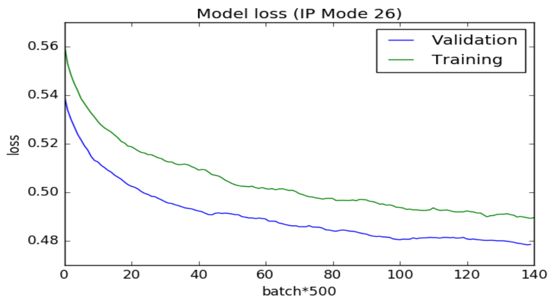


Figure 6.8: Training and validation loss curves using CNN based learning

i.e. four transforms including DCT/DST) for only 4×4 luma intra residual. For all experiments, all-intra (AI) configuration as defined by the HEVC CTC [78] is used.

For the training of the CNN-model, we have taken the Zurich Building dataset [88] which contains over 1000 images in PNG format that are converted to a YUV format of resolution 640×480. The selected transform index based on the RDO at the encoder is used to label the corresponding quantized coefficient block in order to obtain the training classes. Only four CNN-models are trained on the four major intra-prediction modes (IPM), namely DC, Planar, Vertical and Horizontal. A batch size of 32 TCBs is chosen at once to train the model and the number of iterations on the data set (also known as epochs) is set as 20.

In the first experiment, the performance of the training process in terms of classification accuracy on both validation and test data set is evaluated. The validation data set is generated by choosing randomly 10% of the data samples from the training set and are not used for training the model. Validation data set prevents over-fitting. The test data set is generated from the first frame of the HEVC CTC sequences [89]. Figure 6.8 shows the classification loss curve for both training and validation data set. Clearly, both training and validation loss reduce with the number of iterations over the batch of TCB.

Accuracy for	N=1		N=3	
IPM 26	Validation	Test	Validation	Test
CNN	0.75	0.72	0.54	0.45
PCA	0.71	0.67	0.43	0.37

Table 6.8: Trained CNN-model classification accuracy

Table 6.8 shows the CNN-model classification accuracy for both N=1 and N=3 in case of vertical IP mode (i.e. IPM 26). For comparison, the classification accuracy of using a Principle Component Analysis (PCA) along with a decision tree classifier is also presented. From the table, it is observed that CNN-based classifier outperforms the PCA based classifier on both the validation and test data sets. Moreover, a classification accuracy (i.e. the learned model is able to find the good transform index) of over 70% and 45% in case of N=1 and N=3 respectively on both test and validation data sets is achieved. It shows that a correlation exists between the quantized transform coefficients and their corresponding transform indexes across different contents and is well captured by the CNN-model. A random case would provide 50% and 25% respectively for the above two cases showing that the model is unable to find correlation between the data. Similar trends were observed for other IP modes.

Class	Sequence	EP	CTXT	CNN	NoIndex
A	Nebuta	-0.32	-0.28	-0.28	-0.20
	PeopleOnStreet	-0.71	-0.73	-0.80	-1.16
	SteamLocomotive	0.02	-0.01	-0.02	-0.11
	Traffic	-0.81	-0.77	-0.84	-1.19
	Overall	-0.45	-0.45	-0.49	-0.66
B	BasketballDrive	-0.38	-0.57	-0.54	-0.42
	BQTerrace	-1.62	-1.65	-1.79	-2.22
	Cactus	-1.31	-1.20	-1.36	-1.16
	Kimono	0.34	-0.10	0.02	0.04
	ParkScene	-0.44	-0.47	-0.56	-1.35
	Overall	-0.68	-0.80	-0.84	-1.11
C	BasketballDrill	-4.54	-4.66	-5.01	-4.29
	BQMall	-2.13	-1.92	-2.11	-2.72
	PartyScene	-2.19	-2.31	-2.37	-2.87
	RaceHorses	-1.69	-1.49	-1.73	-2.17
	Overall	-2.64	-2.60	-2.80	-3.01
D	BasketballPass	-1.86	-1.51	-1.76	-2.34
	BlowingBubbles	-2.28	-2.22	-2.69	-2.80
	BQSquare	-3.05	-3.04	-3.06	-3.34
	RaceHorses	-2.72	-2.39	-2.50	-2.59
	Overall	-2.48	-2.29	-2.50	-2.77
E	FourPeople	-0.96	-0.97	-0.99	-1.36
	Johnny	-0.26	-0.49	-0.73	-0.91
	KristenAndSara	-1.46	-1.57	-1.75	-1.87
	Overall	-0.89	-1.01	-1.16	-1.38
F	BaskeballDrillText	-4.68	-5.13	-5.24	-4.71
	ChinaSpeed	-1.97	-2.02	-2.12	-2.00
	SlideEditing	-1.54	-1.78	-1.69	1.97
	SlideShow	-1.61	-1.73	-1.98	-1.44
	Overall	-2.45	-2.66	-2.76	-2.53
Overall		-1.60	-1.63	-1.76	-1.91

Table 6.9: BD-Rate gain in % on first frame for N=1 case, using transform index prediction models

In the next experiments, we incorporated the CNN-model inside the HEVC to assist the entropy coding process as illustrated in Figure 6.6. Table 6.9 and 6.10 show the BD-rate gains for different HEVC CTC sequences for N=1 and N=3. In order to show the performance enhancement with the proposed scheme, the results are compared to a conventional fixed length (FL) coding schemes under two variants. The first variant encodes the bits equi-probably (bypass mode) and the second variant utilizes entropy coding with CABAC context (regular mode) when coding the bits. The results of these two variants are presented under *EP* and *CTXT* respectively.

Finally, the BD-rate gains obtained by employing the proposed approach are presented in the tables under *CNN*. A consistent gain across all classes of sequences is observed. The proposed method provides an overall gain of around 0.1% and 0.2% in case of N=1 and N=3 respectively. For sequences like Blowing bubble and Race horses, around 0.5% gain is observed over the conventional transform index signaling approach. In order to illustrate the upper bound of the performance with perfect prediction of index, the BD-rate gain without coding of index for above used IP modes is computed and is presented under *NoIndex*. It is observed that with the help of the proposed CNN model, this performance gap is partially covered.

Further, Table 6.11 presents prediction accuracy of the CNN in percentage when the CNN is

Class	Sequence	EP	CTXT	CNN	NoIndex
A	Nebuta	-0.37	-0.38	-0.40	-0.37
	PeopleOnStreet	-0.75	-0.69	-0.90	-1.75
	SteamLocomotive	0.03	-0.03	-0.03	-0.19
	Traffic	-0.85	-0.83	-1.10	-1.82
	Overall	-0.49	-0.48	-0.61	-1.03
B	BasketballDrive	-0.22	-0.42	-0.48	-0.47
	BQTerrace	-1.44	-1.56	-1.70	-3.11
	Cactus	-1.02	-1.07	-1.25	-2.48
	Kimono	0.18	-0.27	-0.05	-0.08
	ParkScene	-0.45	-0.59	-0.74	-2.81
	Overall	-0.59	-0.67	-0.84	-1.79
C	BasketballDrill	-3.14	-3.36	-3.34	-3.29
	BQMall	-1.74	-1.81	-1.91	-3.33
	PartyScene	-2.03	-2.14	-2.15	-3.89
	RaceHorses	-1.59	-1.57	-1.82	-3.38
	Overall	-2.12	-2.22	-2.31	-3.47
D	BasketballPass	-1.12	-1.32	-1.20	-2.02
	BlowingBubbles	-1.91	-1.76	-2.25	-2.98
	BQSquare	-2.37	-2.22	-2.50	-3.54
	RaceHorses	-2.29	-1.91	-2.50	-2.74
	Overall	-1.92	-1.80	-2.11	-2.82
E	FourPeople	-0.64	-1.00	-1.24	-1.79
	Johnny	-0.58	-0.46	-0.69	-0.83
	KristenAndSara	-0.87	-1.08	-1.09	-2.08
	Overall	-0.70	-0.85	-1.00	-1.57
F	BaskeballDrillText	-3.06	-3.56	-3.83	-3.57
	ChinaSpeed	-1.86	-1.83	-1.89	-2.15
	SlideEditing	-1.21	-1.26	-1.56	-1.93
	SlideShow	-1.42	-1.23	-1.44	-1.67
	Overall	-1.89	-1.97	-2.18	-2.33
Overall		-1.28	-1.33	-1.51	-2.17

Table 6.10: BD-Rate gain in % on first frame for N=3 case, using transform index prediction models

Sequences	Index Prediction accuracy in %		non-DCT/Total
	outside RDO	inside RDO	in %
PeopleOnStreet	71	85	42
BQSquare	73	91	56
SlideShow	81	93	63

Table 6.11: CNN-model prediction accuracy in HEVC vs actual transform usage statistics

used outside and inside RDO respectively. Clearly, RDO favors the prediction decision made using CNN model. This is expected as it is less expensive to encode the bit when CNN predicts correctly.

6.4.5 Conclusions

In this section, a novel CNN-based transform index prediction method has been proposed. It is demonstrated that it is possible to partially infer the transform index from the quantized coefficient values by employing a CNN prediction model trained off-line on a large independent data set. The proposed method shows a consistent improvement in the BD-rate gain over the fixed length coding scheme in almost all cases. This method provides an average gain of around

0.2% and a maximum gain up to 0.59%.

Complexity of the proposed algorithm

The proposed method has a negligible impact on the decoder side complexity but a high impact on the encoder side complexity as the CNN-model is used inside the RDO loop. However, the complexity can be substantially reduced by efficient hardware implementation of CNN as shown in literature [90]. Finally, the complexity reduction is not addressed in this paper and is considered for future work.

Application to other syntax elements inside HEVC

The index prediction method can in general be applied to other syntax elements in the HEVC. For example, a 2-class model similar to the one proposed above can be applied to predict the transform skip flag using the coded coefficient values. Some experiments were carried during this work and showed a classification accuracy of over 90% to predict a transform skip flag from its coded coefficient values. The further exploration of such methods is left for the future work.

6.5 Discussion and Conclusions

Different modifications are proposed in this section to reduce the overhead of coding the basis vectors significantly. It is observed that by reducing the overhead of coding the basis vectors, it is possible to achieve coding gains for 4K sequences. However, the gains in case of resolutions lower than 4K are very low and loss is still observed for Class C and D sequences where the overhead after the proposed improvement is still very high as shown in Figure 5.7. It proves that on-line learning is difficult to make it work.

Further, the last significant coefficient position coding is modified in the context of partial transforms. A small gain of about 0.3% is observed by using this approach.

For improving the transform index coding cost, a CNN-based transform index prediction method has been proposed in this chapter. The proposed method showed consistent improvements in the BD-rate gain over the fixed length coding scheme

Finally, it is concluded here that it is hard to achieve good coding gains for low-resolution sequences where the overhead cost of signaling the basis vectors is still considerable. This makes the online learning less promising. However, the methods presented in this chapter such as incomplete transforms, modified last significant coefficient position coding, transform index prediction using CNN etc. can be applied in general to off-line learning methods to achieve better performances. We will show in the next chapter that improved off-line learning can lead to very solid gains.

III

Improving offline adaptive transform learning

7 Offline scheme improvement

7.1 Introduction

The discussion in Chapter 5 has shown that offline learned non-separable multiple transforms have potential to achieve solid compression improvement in comparison to its online counterpart. This is mainly due to a lesser overhead to encode in the bit-stream. Also, it is well studied in the literature that these transforms have better compression efficiency compared to other methods that either use systematic transforms or offline learned separable transforms. However, the methods involving non-separable transforms are often rejected due to the high complexity at the encoder. Reducing the complexity also leads to reduction in the overall compression gain. To address this problem, in this work, an improved Mode Dependent Transform Competition (IMDTC) scheme has been proposed which provides gains equivalent to state-of-the-art methods but with much less complexity.

In this chapter, the shortcomings of the existing methods are discussed first. Then, several improvements over the existing MDTC scheme [39] are proposed by improving the training process and extending the transform learning to chroma blocks as well as larger blocks of size 16×16. This proposed scheme, named as the Improved MDTC (IMDTC) in this thesis, utilizes less transform candidates to provide similar gains as [39] and considerably reduces the computational complexity. Finally, several modifications are proposed to further reduce the encoder complexity and memory requirements with a minimum impact on the coding gain. This involves use of partial transforms, instead of full transforms, and is named as the Low Complexity IMDTC (Low-IMDTC).

The remainder of this chapter is organized as follows. In Section 7.2, a brief analysis on the shortcomings of the existing methods is discussed. Section 7.3 describes several improvements over the previous work to obtain higher performance. Section 7.4 presents the simulation results along with a detailed comparison of the results against state-of-the-art schemes. Conclusion is presented in Section 7.5.

7.2 Observations on training sets used for the design of MDTC

Ideally, the training set is designed to span a space of all possible residual statistics that are a good representation of the residuals present in natural videos. Therefore, these residuals are in general obtained from a diverse collection of video sequences. In this section, the effect of the choice of the training set on the overall performance of the MDTC scheme is analyzed.

The MDTC scheme proposed in [39] uses a training set obtained from the class B and class C sequences from HEVC Common Test Conditions (CTC) for learning new transforms. These transforms are then tested on the complete CTC test set including class B and class C. Due to this overlap, the learned transforms may adapt better to the residual characteristics that are present in the class B and C sequences and thus, become biased towards these sequences.

In order to illustrate the effect of this overlap, we designed several training sets as shown in Table 7.1 where each training set comprises a large collection of residuals extracted from a given sequence or collection of sequences. Next, a set of 4 non-separable transforms, each for 8×8 and 4×4 TU sizes, is learned using the method described in [39]. Finally, these learned transforms are tested on the HEVC CTC sequences using the transform competition inside the modified HEVC encoder as done in [39].

Training Set	Training Sequences
TSET01	BQTerrace
TSET02	BasketballDrive
TSET03	Class B and Class C sequences from CTC [31]
TSET04	First 40 frames of the following sequences: CIF: Foreman, Paris 720p: city, crew, panslow, raven, spincalendar 1080p: Mitsubishi Traffic, Pedestrian area, station2, rolling tomatoes, Rush hour, sunflower, table_setting, Vintage car, toy and Calendar, walking couple.

Table 7.1: Different Training Sets Used for Learning Transforms

Table 7.1 lists the 4 different training sets and Table 7.2 shows the corresponding performances in terms of BD-rate gain over HEVC when tested on the first frame of the CTC sequences. The columns TSET01 and TSET02 in Table 7.2 shows BD-rate gain when the training set is obtained from a single sequence i.e. BQTerrace and BasketBallDrill respectively. It is clearly visible that the transforms favor the sequences on which they have been learned. The transform set learned on BQTerrace is heavily biased towards BQTerrace. In case of BasketBallDrill, the transform set not only favors itself but also favors sequences that are very closely related (e.g. BasketBallDrill, BasketBallPass, etc.). This is evident from high BD-rate gain for those sequences.

The TSET03 represents the training set used in [39] and the TSET04 is designed to contain 17 different sequences of various resolutions and are completely independent from the CTC test set. The BD-rate gain in Table 7.2 shows that the transforms obtained from TSET03 outperforms all other training set choices as it provides relatively high gain on class B and C sequences due to the overlap between the training set and test set. The gains using training set TSET04 are more uniform and do not favor any particular sequence. This was expected as it has no overlap with the test set.

Class	**Sequences**	**BD-rate in % with training set**			
		TSET01	**TSET02**	**TSET03**	**TSET04**
A	Nebuta	-0.16	-0.78	-1.22	-1.71
	PeopleOnStreet	-0.52	-3.58	-3.93	-4.56
	SteamLocomotive	-0.04	-0.35	-0.53	-0.97
	Traffic	-0.79	-4.02	-4.54	-5.12
	Overall	**-0.38**	**-2.18**	**-2.56**	**-3.09**
B	BasketballDrive	-0.1	-4.67	-3.69	-2.76
	BQTerrace	-9.91	-4.03	-5.54	-3.74
	Cactus	-0.97	-4.02	-5.13	-3.93
	Kimono	0.07	-0.62	-0.78	-0.52
	ParkScene	-0.87	-2.17	-2.76	-3.61
	Overall	**-2.36**	**-3.1**	**-3.58**	**-2.91**
C	BasketballDrill	-1.64	-9.9	-13.91	-7.28
	BQMall	-0.77	-5.09	-5.06	-3.45
	PartyScene	-1.55	-4.51	-4.67	-3.6
	RaceHorses	-1.26	-4.66	-5.11	-4.43
	Overall	**-1.3**	**-6.04**	**-7.19**	**-4.69**
D	BasketballPass	-0.06	-5.18	-4.14	-3.16
	BlowingBubbles	-0.96	-5.67	-5.63	-3.83
	BQSquare	-2.17	-4.45	-4.78	-3.5
	RaceHorses	-0.69	-4.9	-5.55	-4.81
	Overall	**-0.97**	**-5.05**	**-5.03**	**-3.82**
E	FourPeople	-0.44	-5.05	-5.23	-4.51
	Johnny	-0.05	-3.87	-3.92	-2.6
	KristenAndSara	-0.48	-4.65	-5.06	-3.12
	Overall	**-0.32**	**-4.52**	**-4.74**	**-3.41**
	Overall	**-1.07**	**-4.18**	**-4.62**	**-3.58**

Table 7.2: BD-Rate gain on first frame

The BD-rate gains (over 7%) claimed in [39] over HEVC are obtained from a biased training set TSET03. If we exclude the gains on class B and class C sequences in [39] in order to make the test set independent relatively to the training set, the overall gain reduces to 5% instead of 7%.

Therefore, the above experiment shows that the choice of training sequences may lead to a bias towards certain test sequence and provides artificial better final performance. Appendix **??** provides further analysis on the design of an independent training set.

In this chapter, we have used the set TSET04 consisting of 17 different video sequences that are 1) visually diverse, 2) from different resolutions, and 3) representing a large range of diversity in terms of bit-rate and PSNR (shown by BD-rate points on a BD-rate plot in Figure 7.1), as an un-biased training set to learn transforms. The BD-rate gain on the CTC test sequences using these learned transforms is considered as an **anchor** in the rest of the paper and all the proposed modifications are compared to the **anchor**.

In the next section, several proposed improvements over the existing MDTC scheme are described and the results are presented in Section 7.5.

7.3 Proposed improvements to MDTC scheme to get proposed IMDTC

This chapter, so far, analyzed the effect of using different training sets on the performance of the MDTC scheme. The BD-rate gains obtained from an independent un-biased training set (TSET04), which is referred as the **anchor** in this chapter, was also presented in Table 7.2.

In this section, a new improved MDTC scheme (so-called IMDTC) is described and includes

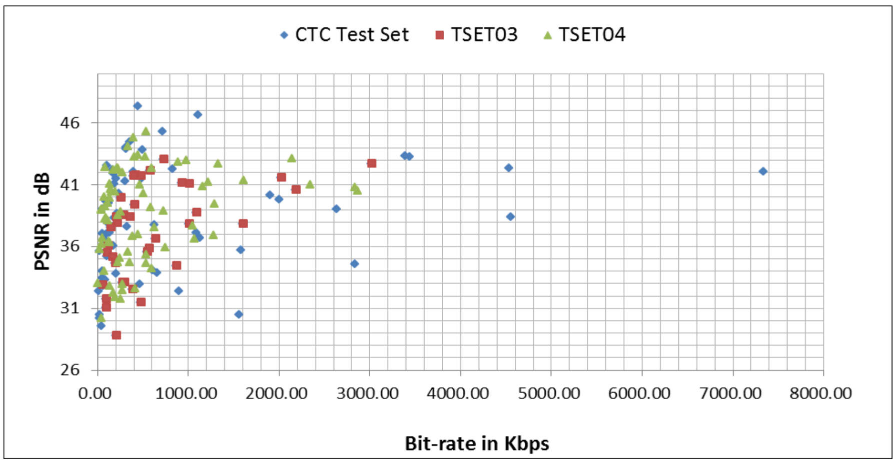


Figure 7.1: Spread of BD-rate points of different sequences taken from three different sets.

a series of improvements over the existing MDTC scheme proposed in [39]. The IMDTC scheme aims at providing state-of-the-art performance at a much lower encoder complexity and memory footprint and thus, revisits the claims made in [91] about the non-applicability of the non-separable transforms for video coding purposes.

In the following sub-sections, modifications to the transform learning scheme are presented. Several extensions to the existing MDTC scheme are presented and aim to improve the performance with minimum increase in the encoder complexity.

7.3.1 Proposed iterative learning on training set

The transforms, so far, are learned on an initial training set which comprise of a large collection of residuals extracted from various sequences in TSET04 using the standard HEVC encoder.

In the proposed method, the initial training set is first replaced by a training set which is generated from various sequences in TSET04 through transform competition between the newly computed transforms and the DCT/DST inside the modified HEVC encoder. Then, given the above training set, new optimized transforms are computed. This two-step iteration is performed as follows:

1. Given a training set of residuals, compute the new optimized transform set.
2. Given the above computed transform set, generate a new training set of residuals from sequences in TSET04.

Figure 7.2 illustrates the above iterative approach using a block diagram. First, the modified HEVC video encoder (left block) encodes the training videos from TSET04 using transform competition between learned transforms and the DCT/DST in order to generate the residual training set. Then, the offline transform learning scheme (right block) uses this new residual training set to compute the new transforms which then replace the old transforms in the modified HEVC encoder for the next iteration.

It is observed from experiments that the transforms generated after several iterations on the training sequences provides better performance than the transforms obtained from the initial

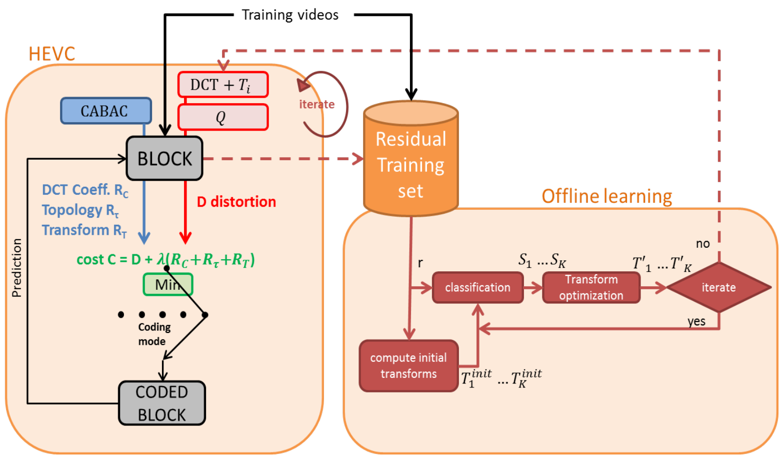


Figure 7.2: The proposed iterative learning on training set.

training set. This is because the residual statistics inside the training set generated from newly computed transforms are different from the statistics found in the initial training set which is generated using only DCT/DST. Therefore, by this iterative approach, we are able to adapt to actual residual statistics that may appear due to the use of newly learning transform candidates.

It has been also observed that the gain due to this iterative approach starts to converge after 3 to 4 iterations.

7.3.2 Learning KLTs on the chroma components

In the literature, the mode dependent schemes have been applied only to the luma residuals and the chroma residuals are always transformed using conventional DCT. This is mainly because the chroma residual statistics are mostly flat. However, it is well known that a correlation exists between the chroma residuals and the luma residuals especially when there is a strong structure present in the scene. This leads to directional structures that can be de-correlated better by using non-separable transforms for chroma residuals as well.

One may simply use transforms learned on luma residuals for the corresponding chroma residuals. In this case, the best transform obtained after brute-force testing on the luma residual is applied on chroma residual, thus avoiding transmitting a separate index for chroma. However, this is possible only in case of 4:4:4 chroma format where the size of luma and chroma residuals are the same.

But, in case of 4:2:0 format, chroma residuals are sub-sampled. Therefore, it is not possible anymore to apply the transforms chosen for luma residual directly on the chroma residual. One possibility is to select a transform as done for luma residuals and an index is transmitted for chroma residuals to indicate the choice of the transform. However, this method would increase both the syntax of transmitting the side information and the encoder complexity.

In order to avoid both these problems, a single transform per IPM is learned and replaces the DCT inside the encoder. A separate transform is learned for Cb and Cr chroma blocks of size 4×4 and 8×8.

Both Cb and Cr residuals of size 4×4 are extracted separately from training sequences under TSET04 to form two data sets, one for Cb and another for Cr. These training sets are referred as TSET04_Cb_init and TSET04_Cr_init. A Mode Dependent Sparse Transform (MDST) scheme [37] is used to generate a set of transforms for each of the training set where the learning process is initialized by a KLT transform on the training set. The performance of this method is further improved by iterating on the training sequences as proposed above in section 7.3.1.

7.3.3 Extension to larger residual blocks

So far, the non-separable transforms have been learned and tested on residuals of size 8×8 and 4×4. As an extension, we have applied various learning algorithms such as MDDT [35], MDST [37] and MDTC [39] to compute sets of adaptive transforms for residuals of size 16×16. These transforms are then tested inside the codec either in place of the conventional DCT (for MDDT and MDST) or in addition to the conventional DCT (for MDTC). However, the residual blocks of size 32×32 are coded using conventional DCT transforms as before in order to maintain complexity reasonably low.

7.3.4 Low-complexity IMDTC using partial transforms

The extension of the transform set to larger blocks as shown in the last section, is accompanied by an increase in both the encoder/decoder complexity and the memory footprint required to store the basis vectors. In order to avoid the problem of storage and encoder/decoder complexity, a low-complexity IMDTC scheme is proposed to provide similar gains but at much reduced complexity and memory footprint.

The transforms learned on the training set have a high energy packing efficiency which is computed as the ratio of the sum of energies in first k coefficients to the total energy contained in all N^2 coefficients for a block of size N×N. During the training process, a singular value decomposition (SVD) is used to find the optimized transform basis vectors that can minimize the error between the residuals and the transformed coefficients inside the transform optimization step. In our proposed modification, the energy compaction property of SVD is exploited and only first k basis vectors of each learned transform are retained.

It has been observed during our experiments that dropping the last few vectors has negligible effect on the final Rate-Distortion (R-D) coding performance of the learned KLTs. This is because, due to the energy compaction, high frequency coefficients carry little signal and are often zeroed out by the quantization process in HEVC.

The number k of basis vectors retained for 256×256-sized transform (associated with 16×16 blocks) is empirically found to be 64 and for 64×64-sized transform (associated with 8×8 blocks) is found to be 48. These new transforms of size 256×64 and 64×48 are referred as partial transforms or partial KLTs in this thesis. However, the transforms for 4×4 residuals are kept as is, and a full set of basis vectors is used.

Additionally, as proposed in section 6.3.2 of Chapter 6, the last significant coefficient position coding inside HEVC is modified to further improve the overall performance. It has been shown that one can infer the position of y-coordinates of the last coefficient position from the x-coordinates in case of partial transforms, thus leading to better R-D performance.

In the next section, results from various experiments are presented to validate the performance improvement obtained from the above proposed modifications.

7.4 Experimental results

This section provides the detailed experimental results in order to validate the performance improvement due to the modifications proposed in Section 7.3. For training purposes, we have chosen several sequences of different resolutions and frame rate as shown in Table 7.1 under TSET04. These sequences are independent to the HEVC CTC sequences. The transform set obtained is then tested inside the modified HEVC test software (HM 15.0) in competition with DCT/DSTs as in [39]. We have utilized two encoder configurations, All Intra (AI) and Random Access (RA), to illustrate the R-D performance of proposed modifications.

Table 7.3 shows the BD-rate gains of the original MDTC scheme (**anchor**) in comparison to HEVC, under the AI configuration, by using 4 transforms per IPM for 8×8 and 4×4 residual block size that are learned on the training set TSET04. An average luma gain of around 3.76% is achieved. However, a loss of over 2% is observed in chroma components.

From now, we will investigate the performance of the proposed IMDTC scheme against both the anchor and the standard HEVC.

In order to investigate the modifications proposed in section 7.3, we have designed several experiments to probe the R-D performance impacted by: 1) iterating over the training set; 2) using transforms for chroma residuals; 3) extending the transforms to 16×16 TUs. Finally, compar-

Class	**Sequences**	**BD-rate in % for**		
		Y	**U**	**V**
A	Nebuta	-1.22	-0.38	0.41
	PeopleOnStreet	-4.63	2.31	3.2
	SteamLocomotive	-0.82	3.81	3.11
	Traffic	-5.11	1.99	2.37
	Overall	**-2.95**	**1.93**	**2.27**
B	BasketballDrive	-3.53	3.98	3.79
	BQTerrace	-3.91	1.79	2.4
	Cactus	-4.05	2.93	3.6
	Kimono	-1.65	2.07	1.77
	ParkScene	-3.7	4.34	3.56
	Overall	**-3.37**	**3.02**	**3.02**
C	BasketballDrill	-7.66	2.28	2.62
	BQMall	-4.01	2.3	2.67
	PartyScene	-3.68	1.81	1.82
	RaceHorses	-4.17	2.21	2.06
	Overall	**-4.88**	**2.15**	**2.29**
D	BasketballPass	-4.1	1.86	1.87
	BlowingBubbles	-3.88	1.99	2.08
	BQSquare	-3.63	2.3	2
	RaceHorses	-4.71	1.32	1.47
	Overall	**-4.08**	**1.87**	**1.86**
E	FourPeople	-4.27	2.76	2.79
	Johnny	-2.97	2.52	2.81
	KristenAndSara	-3.28	2.33	2.47
	Overall	**-3.51**	**2.54**	**2.69**
	Overall	**-3.76**	**2.30**	**2.43**

Table 7.3: BD-Rate gain for 4 KLTs per IPM for residual size 4×4 and 8×8 (AI configuration)

ison of R-D performance, encoding complexity and storage requirement between the standard HEVC, the original MDTC scheme, the proposed IMDTC scheme and the proposed low complexity IMDTC scheme are provided.

The encoder complexity is computed by measuring the total time ratio it takes to encode the whole sequence for the proposed scheme relatively to the standard HEVC. The complexity is presented as a percentage of $T_{\text{proposed}}/T_{\text{HEVC}}$. The term encoder/decoder complexity and the encoding/decoding time are used interchangeably even if the coding time is not the only relevant number to look at for assessing complexity.

For the storage purposes, the basis vectors are the learned transforms are quantized to 1byte/basis-value and the memory requirement is presented in megabytes (MBs).

7.4.1 Effect of iterating over the training set

The R-D performance obtained at each iteration on the training set is shown. To illustrate the effect of the proposed iterative scheme on the final coding performance, a set of 8 transforms per IPM are learned on 8×8 residual blocks only. This is done to illustrate gains only due to this modification, i.e. the iteration of the learning scheme on the training set.

Table 7.4 presents the BD rate gain, under the All Intra configuration, over different class of sequences taken from the HEVC CTC. The gain under Iter01 is achieved by using the transform set that was learned on the initial training set. Then, a new training set is generated at each subsequent iteration and the results presented in Table 7.4 are obtained from the transform set learned for the corresponding iteration.

	BD-rate in % at iteration				
Class	**iter01**	**iter02**	**iter03**	**iter04**	**iter05**
A	-3.08	-3.48	-3.37	-3.40	-3.39
B	-2.56	-2.91	-2.95	-3.07	-3.00
C	-3.20	-3.78	-4.03	-4.32	-4.23
D	-2.19	-2.61	-2.72	-2.79	-2.85
E	-2.85	-3.33	-3.50	-3.67	-3.62
Average	**-2.77**	**-3.22**	**-3.32**	**-3.45**	**-3.42**

Table 7.4: BD-Rate gain at each iteration with a set of 8 transforms per IPM for 8×8 residual size, AI configuration

In this experiment, we illustrate the performance evolution until five iterations. From Table 7.4, one observes that there is a significant gain after the first iteration. This clearly shows that the training set generated at second iterations is able to cover the actual residual statistics better than the initial training set. There is a slight gain in overall performance for some extra iterations, but convergence seems to be obtained rapidly. In general, 3 to 4 iterations are enough to achieve the convergence.

7.4.2 R-D performance over HEVC due to improved chroma residual coding

Table 7.3 shows a performance drop of more than 2% for chroma component U and V compared to the HEVC. This is mainly due to the increase in the bit-rate which happens due to the increase in the syntax associated to the choice of transform for the luma component, while not improving the distortion of the chroma components.

Class	Sequences	BD-rate in % for		
		Y	**U**	**V**
A	Nebuta	-0.11	-1.64	-1.28
	PeopleOnStreet	-0.06	-1.93	-1.01
	SteamLocomotive	-0.01	-0.04	-0.15
	Traffic	0.01	-1.26	-2.12
	Overall	**-0.04**	**-1.22**	**-1.14**
B	BasketballDrive	-0.14	-2.3	-3.2
	BQTerrace	0.01	-3.69	-2.81
	Cactus	-0.08	-2.16	-2.98
	Kimono	0.00	-0.62	-1.08
	ParkScene	0.08	-1.5	-1.99
	Overall	**-0.03**	**-2.06**	**-2.41**
C	BasketballDrill	-0.35	-7.18	-7.91
	BQMall	-0.09	-1.62	-1.54
	PartyScene	-0.04	-2.32	-2.9
	RaceHorses	-0.17	-2.56	-3.37
	Overall	**-0.16**	**-3.42**	**-3.93**
D	BasketballPass	-0.3	-2.84	-3.67
	BlowingBubbles	-0.11	-3.05	-3.6
	BQSquare	0.01	-1.17	-2.41
	RaceHorses	-0.26	-3.03	-4.08
	Overall	**-0.16**	**-2.52**	**-3.44**
E	FourPeople	-0.18	-1.27	-1.28
	Johnny	-0.15	-2.13	-2.05
	KristenAndSara	-0.09	-2.2	-2.64
	Overall	**-0.14**	**-1.87**	**-1.99**
	Overall	**-0.11**	**-2.22**	**-2.58**

Table 7.5: BD-Rate gain using KLTs for chroma 4×4 residuals only.

As described in Section 7.3.2, chroma residuals show strong directional structures especially when there is a strong edge in the luma. Therefore, we have proposed to learn a set of transforms on a large training set consisting of chroma residuals. A separate transform set is learned for Cb and Cr components and inside the codec, this transform set is used in place of the conventional DCT transform.

In order to illustrate the gains only due to chroma residuals, we have designed the experiment such that luma residuals are transformed using conventional DCT/DST, and chroma residuals uses learned transforms instead of the conventional DCT. Table 7.5 shows the R-D performance where a gain of more than 2% is observed in AI configuration over the standards HEVC. The gain shows that the conventional DCT is not optimal in coding the chroma residuals. High BD-rate gain in sequences like BasketBallDrive, BQTerrace and BasketBallDrill shows that the learned transforms are able to better adapt to the directional structures present in both luma and chroma components. A slight gain in luma is observed which is probably due to the reduction of the chroma bit-rate.

7.4.3 Extension of transforms to 16×16 residual blocks

In this subsection, we will present the experimental results illustrating the R-D performance change when using non-separable transforms on the 16×16 residual blocks.

Class	Sequences	BD-rate in % for		
		Case1	**Case2**	**Case3**
A	Nebuta	-0.85	-1.08	-1.2
	PeopleOnStreet	-1.63	-1.85	-2.79
	SteamLocomotive	-0.86	-1.22	-1.93
	Traffic	-1.48	-1.79	-3.51
	Overall	**-1.2**	**-1.49**	**-2.36**
B	BasketballDrive	-0.4	-0.97	-2.61
	BQTerrace	-0.09	-0.4	-1.4
	Cactus	-0.59	-0.97	-2.05
	Kimono	-1.28	-1.66	-2.46
	ParkScene	-0.86	-1.21	-1.93
	Overall	**-0.64**	**-1.04**	**-2.09**
C	BasketballDrill	-1.1	-1.27	-5.42
	BQMall	0.06	-0.4	-1.14
	PartyScene	-0.09	-0.2	-0.47
	RaceHorses	-0.3	-0.6	-1.51
	Overall	**-0.36**	**-0.62**	**-2.14**
D	BasketballPass	0.24	-0.35	-1.14
	BlowingBubbles	0.02	-0.18	-0.55
	BQSquare	0.27	-0.07	-0.29
	RaceHorses	-0.22	-0.54	-1.3
	Overall	**0.08**	**-0.28**	**-0.82**
E	FourPeople	-0.65	-1.02	-2.14
	Johnny	-0.27	-0.79	-1.59
	KristenAndSara	-0.45	-0.78	-1.78
	Overall	**-0.46**	**-0.86**	**-1.84**
Overall		**-0.52**	**-0.86**	**-1.85**

Table 7.6: BD-Rate gain using KLTs for luma 16×16 residuals for three cases

Table 7.6 shows the BD-rate gain for three different configurations (or cases in the table) where

1. 1KLT per IPM in place of DCT;
2. 1KLT per IPM in addition to DCT;
3. 4KLTs per IPM in addition to DCT

are applied to the 16×16 residual blocks. The R-D performance is better in case the transform competition is employed, i.e. for 2nd and 3rd configuration, instead of replacement of the DCT. At the same time, it is found to be computationally more complex compared to the first case.

It should also be noted that the gain obtained is also proportional to the resolution of the video sequence with class A sequences having higher gains compared to class D sequences. This is explained by the fact that the percentage of 16×16 size residual blocks is considerably higher in class A and class B compared to class C and class D.

The encoder complexity for the 3rd configuration is found to be 2.25 times the standard HEVC. This is because the non-separable transforms inside the RDO loop involve a matrix multiplication on the residual block. Therefore, it requires large number of operations for a transform of size 256×256 (associated with 16×16 blocks. Moreover, it is expensive to store the transform basis inside the memory especially in the 3rd configuration which uses 4 KLTs per IPM, thus leading to total of a memory footprint around 9 MB.

In the next subsection, results for the proposed low-complexity IMDTC scheme (Low-IMDTC) are presented. Low-IMDTC considerably reduces both the memory requirement and encoding/decoding complexity without impacting the R-D coding gains much.

7.4.4 Comparison of low-complexity IMDTC and full IMDTC scheme

In this subsection, the R-D performance gain of the complete IMDTC system is presented along with the encoder and decoder complexity in terms of time taken to encode/decode a sequence compared to standard HEVC. The results of the proposed low-complexity IMDTC system is then presented to compare the performance drop and complexity reduction relatively to the full IMDTC system.

For a full IMDTC system, 4KLTs/IPM for all 4×4, 8×8 and 16×16 luma residuals are tested in competition with the conventional DCT/DST inside the codec. In case of 4×4 chroma residuals, the DCT is replaced by a KLT which is chosen from a set of KLTs based on the chroma component and IPM. Table 7.7 shows the BD-rate gain of the full IMDTC scheme for the All Intra configuration.

The result shows that a gain of 5.41% over the standard HEVC is achieved under the AI configuration for luma component. For chroma, a gain of over 2% is obtained for both chroma components. This is 1.65% higher gain in luma compared to the anchor (Table 7.2) and nearly 4.5 – 5% higher gain in chroma. The gains achieved by full IMDTC system are substantial and show the potential of using non-separable transforms.

However, as shown in Table 7.7, the full IMDTC system is on an average 3.5 times more complex than the standard HEVC. This is much less compared to the 20 times complexity of the original MDTC scheme which also provides similar gains as mentioned in section 7.2.

Class	Sequences	Full IMDTC system (AI)					Low-complexity IMDTC system (AI)					Low-complexity IMDTC system (RA)				
		Y	U	V	Enc. Time	Dec. Time	Y	U	V	Enc. Time	Dec. Time	Y	U	V	Enc. Time	Dec. Time
A	Nebuta	-2.23	-4.03	-2.62			-1.71	-3.6	-2.24			0.36	-5.6	-3.74		
	PeopleOnStreet	-6.63	-4	-2.76			-6.41	-3.81	-2.53			-3.29	-1.23	-0.3		
	SteamLocomotive	-2.43	2.13	2.22			-2.12	2.69	2.47			-2.46	2.58	0.87		
	Traffic	-7.22	-2.49	-3.54			-7	-2.27	-3.42			-5.59	-1.72	-3.04		
	Overall	**-4.63**	**-2.1**	**-1.67**	**391%**	**91%**	**-4.31**	**-1.75**	**-1.43**	**301%**	**96%**	**-2.74**	**-1.49**	**-1.55**	**130%**	**126%**
B	BasketballDrive	-5.72	-0.78	-1.85			-5.68	-0.61	-1.89			-3.48	-0.05	-1.58		
	BQTerrace	-4.84	-4.38	-3.08			-4.8	-3.99	-2.54			-3	-4.09	-1.94		
	Cactus	-5.76	-1.7	-2.42			-5.49	-1.65	-2.15			-3.59	-1.5	-1.76		
	Kimono	-3.52	-0.78	-1.75			-3.42	-0.61	-1.69			-1.44	-0.64	-1.19		
	ParkScene	-4.9	-0.3	-1.29			-4.59	0.36	-0.86			-3.05	-0.94	-1.39		
	Overall	**-4.95**	**-1.59**	**-2.08**	**400%**	**94%**	**-4.8**	**-1.3**	**-1.83**	**300%**	**91%**	**-2.91**	**-1.4**	**-1.51**	**124%**	**117%**
C	BasketballDrill	-12.48	-7.45	-8.33			-12.1	-7.57	-8.48			-7.66	-6.04	-6.1		
	BQMall	-5.22	-1.2	-1.2			-5.2	-1.19	-1.15			-2.97	-0.47	-1.3		
	PartyScene	-4.38	-1.77	-2.42			-4.25	-1.4	-2.14			-3.06	-1.71	-2.4		
	RaceHorses	-5.74	-2.2	-3.98			-5.47	-2.07	-3.54			-2.73	-2.06	-2.98		
	Overall	**-6.95**	**-3.15**	**-3.98**	**401%**	**100%**	**-6.75**	**-3.06**	**-3.83**	**301%**	**95%**	**-4.1**	**-2.57**	**-3.2**	**127%**	**130%**
D	BasketballPass	-5.51	-2.69	-3.86			-5.57	-2.88	-3.73			-3.12	-1.98	-2.77		
	BlowingBubbles	-4.94	-2.74	-3.19			-4.83	-2.48	-2.76			-3.72	-2.57	-3.91		
	BQSquare	-4.4	-0.77	-1.69			-4.32	-0.43	-1.4			-3.21	-1.55	-1.99		
	RaceHorses	-6.44	-4.02	-5.03			-6.13	-3.95	-4.81			-3.11	-2.71	-2.77		
	Overall	**-5.32**	**-2.55**	**-3.44**	**404%**	**112%**	**-5.21**	**-2.43**	**-3.17**	**301%**	**103%**	**-3.29**	**-2.2**	**-2.86**	**128%**	**129%**
E	FourPeople	-6.26	-1.64	-1.97			-6.23	-1.58	-1.79			-6.2	-2.04	-1.99		
	Johnny	-4.56	-2.18	-1.27			-4.65	-2.04	-1.3			-4.86	-2	-1.95		
	KristenAndSara	-5.28	-1.92	-2.22			-5.3	-1.77	-2.32			-5.55	-2.16	-2.25		
	Overall	**-5.37**	**-1.91**	**-1.82**	**399%**	**99%**	**-5.39**	**-1.8**	**-1.8**	**295%**	**100%**	**-5.54**	**-2.06**	**-2.06**	**116%**	**114%**
	Overall	**-5.44**	**-2.26**	**-2.6**	**399%**	**99%**	**-5.29**	**-2.07**	**-2.41**	**290%**	**97%**	**-3.72**	**-1.95**	**-2.24**	**125%**	**123%**

Table 7.7: Comparison of BD-rate gain of complete IMDTC system and Low-IMDTC system in All Intra (AI) and Random Access (RA)
(4 KLTs + 1 DCT)/IPM for 4×4, 8×8 and 16×16 Luma Residuals
1 KLT/IPM for 4×4 Chroma Residuals

It is still possible to reduce the complexity further by employing the proposed low-complexity IMDTC scheme as described in section 7.3.4. It is proposed to use partial transforms instead of complete transforms to achieve the reduction in both complexity and memory requirement.

The R-D performance of the low-complexity IMDTC system along with the encoding and decoding time are also illustrated in Table 7.7 in order to compare with the full IMDTC scheme. In this experiment, only one fourth of the total number of basis vectors are retained for 256×256 transform size, thus resulting in partial transforms of size 256×64. Additionally, three-fourth of the total number of basis vectors are retained for transforms size 64×64 resulting in partial transforms of size 64×48.

It is observed that the low-complexity IMDTC provides a saving of nearly 100% in encoding time with a negligible loss of 0.2% in the R-D performance compared to the full IMDTC scheme. The decoding time remains unaffected in both cases. Table 7.7 also illustrates the R-D performance of the low-complexity IMDTC scheme under RA configuration which shows a gain of 3.85% in luma and around 2% in chroma over standard HEVC with a complexity 1.25 times HEVC. The performance of full IMDTC scheme under RA configuration is just 0.05% higher than the low-complexity IMDTC scheme but the encoder complexity is 1.32 times the HEVC. Also, comparing the results to [39] under RA configuration, comparable results are achieved with much less encoder complexity.

Therefore, the final low-complexity IMDTC system definitely provides a better trade-off between the R-D performance and the encoder/decoder complexity. Next sub-section further compares complexity and memory requirements with state-of-the-art methods.

7.4.5 Comparison of R-D performance, encoding complexity and memory requirement with MDTC scheme

In this subsection, a comparison between our proposed IMDTC system and the original MDTC scheme as proposed in [39] is presented in terms of three parameters: 1) Number of transforms, 2) R-D performance, 3) encoder complexity and 4) memory requirement.

Table 7.8 compares the R-D performance of luma component for MDTC [39], the proposed full IMDTC and Low-IMDTC system. The MDTC scheme from [39] still performs better than the proposed approach but this is expected due to two reasons: first the scheme in [39] uses much higher number of transform candidates per IPM (32 for 8×8 and 16 for 4×4), and second the transform set in [39] is learned on class B and class C sequences which creates a bias as described in detail in section 7.2.

Table 7.8 also illustrates the R-D performance achieved when excluding class B and class C sequences for comparing both MDTC and IMDTC system performances. It is observed that the proposed IMDTC system provides equivalent gains to the original MDTC system, but we utilize only 4 transform candidates per IPM for both 4×4 and 8×8 residual size compared to 16 4×4 and 32 8×8 for MDTC. However, for class A sequences, the proposed IMDTC system outperforms the original MDTC.

In terms of encoding time, both full IMDTC and low-complexity IMDTC schemes substantially outperform the original MDTC scheme as shown in Table 7.8. In terms of memory requirement,

		BD-rate in % for (AI)			BD-rate in % for (RA)		
Class	**Sequences**	**MDTC [39] 4x4: 1+16 8x8: 1+32**	**Full IMDTC 4x4: 1+4 8x8: 1+4 16x16: 1+4**	**Low-complexity IMDTC 4x4: 1+4 8x8: 1+4 16x16: 1+4**	**MDTC [39] 4x4: 1+16 8x8: 1+32**	**Full IMDTC 4x4: 1+4 8x8: 1+4 16x16: 1+4**	**Low-complexity IMDTC 4x4: 1+4 8x8: 1+4 16x16: 1+4**
A	Nebuta	-1.15	-2.23	-1.71	-0.13	0.35	0.36
	PeopleOnStreet	-5.65	-6.63	-6.41	-2.27	-3.39	-3.29
	SteamLocomotive	-0.68	-2.43	-2.12	0.03	-3.45	-2.46
	Traffic	-6.06	-7.22	-7	-5.12	-5.8	-5.59
	Overall	**-3.38**	**-4.63**	**-4.31**	**-1.87**	**-3.07**	**-2.74**
B	BasketballDrive	-5.52	-5.72	-5.68	-2.36	-3.52	-3.48
	BQTerrace	-9.22	-4.84	-4.8	-4.93	-3.05	-3
	Cactus	-10.92	-5.76	-5.49	-7.68	-3.73	-3.59
	Kimono	-1.8	-3.52	-3.42	-1.18	-1.5	-1.44
	ParkScene	-5.27	-4.9	-4.59	-3.69	-3.12	-3.05
	Overall	**-6.55**	**-4.95**	**-4.8**	**-3.97**	**-2.98**	**-2.91**
C	BasketballDrill	-25.06	-12.48	-12.1	-14.8	-7.98	-7.66
	BQMall	-6.21	-5.22	-5.2	-3.64	-2.97	-2.97
	PartyScene	-6.19	-4.38	-4.25	-4.3	-3.08	-3.06
	RaceHorses	-6.75	-5.74	-5.47	-3.24	-2.81	-2.73
	Overall	**-11.05**	**-6.95**	**-6.75**	**-6.49**	**-4.21**	**-4.1**
D	BasketballPass	-6.15	-5.51	-5.57	-3.14	-3.22	-3.12
	BlowingBubbles	-6.73	-4.94	-4.83	-3.95	-3.7	-3.72
	BQSquare	-6.12	-4.4	-4.32	-3.61	-3.22	-3.21
	RaceHorses	-7.13	-6.44	-6.13	-3.2	-3.22	-3.11
	Overall	**-6.53**	**-5.32**	**-5.21**	**-3.47**	**-3.34**	**-3.29**
E	FourPeople	-6.33	-6.26	-6.23	-6.7	-6.25	-6.2
	Johnny	-5.21	-4.56	-4.65	-5.78	-4.98	-4.86
	KristenAndSara	-6.01	-5.28	-5.3	-6.17	-5.53	-5.55
	Overall	**-5.85**	**-5.37**	**-5.39**	**-6.22**	**-5.59**	**-5.54**
Overall		**-6.67**	**-5.44**	**-5.29**	**-4.4**	**-3.84**	**-3.72**
Overall without Class B and C		**-5.25**	**-5.07**	**-4.9**	**-3.85**	**-4**	**-3.85**
Enc. Time		**2000%**	**347%**	**256%**	**-**	**132%**	**125%**
Dec. Time		**120%**	**99%**	**97%**	**-**	**121%**	**123%**
ROM		**4.51 MB**	**9.33 MB**	**2.63 MB**	**-**	**9.33 MB**	**2.63 MB**

Table 7.8: Comparison between MDTC, IMDTC and Low-IMDTC scheme

the proposed full IMDTC scheme requires more memory to store the full 256×256 size non-separable transforms for 16×16 residuals. However, the Low-IMDTC scheme requires much less memory to store the transforms. It is shown in Table 7.8 that low-complexity MDTC scheme takes 50% less memory than the original MDTC scheme.

In conclusion, we have shown a R-D coding gain of 5.29% at an encoder complexity of 2.5 times the standard HEVC. This is considerably better than the original MDTC scheme proposed in [39] which provides comparable gains but at an encoding complexity of 20 times the HEVC and a storage twice the low-complexity IMDTC system. In general, a complexity addition of 250% over the standard HEVC with 5.29% bit-rate saving is fairly acceptable when designing a new codec if one refers to the on-going work in the JVET group. Moreover, the proposed IMDTC scheme may further benefit from fast decisions inside the RDO loop and parallelization of matrix multiplication operations for large size transforms, thus leading to even further reduction in both encoder and decoder complexities. Therefore, we conclude here that non-separable transforms still have a potential to provide higher gains without adding too much complexity.

7.5 Conclusion

This chapter has investigated the short-comings of the existing Mode Dependent Transform Competition (MDTC) scheme and proposed an improved MDTC scheme (IMDTC) for HEVC. The proposed scheme employs a series of improvements over the existing MDTC scheme and provides a gain of 5.29% over standard HEVC under all intra (AI) configuration and at encoder complexity of just 2.5 times standard HEVC. The results suggest that the proposed scheme performance is equivalent to the existing MDTC scheme, but is considerably less complex and, therefore, shows that it is practical to use non-separable transforms.

Additionally, this chapter has investigated the effect of using biased and non-biased training sets on the final R-D performance. It is shown that the overlap between the training and test sets could affect the overall gains obtained on the test set and, therefore, it is important to find an independent training set to generalize the transforms.

8 Content adaptability improvement: playing with datasets

8.1 Introduction

In the spirit of improving the off-line adaptive transform learning in this part, the previous chapter has focused on improving the existing MDTC scheme itself by applying several modifications and extensions within the scheme. This chapter, in contract to the previous chapter, focuses on a different axis of improvement and uses different datasets to achieve higher coding gains.

The method proposed in the previous chapter showed significant coding gain of over 5% compared to the standard HEVC and at a complexity of 2.9 times the standard HEVC. It also described some techniques to generate a training set which is not biased towards particular content type so that the learned transforms are generic and therefore, achieve on an average high compression gains on a wide range of sequences.

On the other hand, in Chapter 5, a set of content-adaptive transforms was learned per sequence by employing the proposed on-line learning scheme. These content-specific transforms provide better coding gains compared to the off-line generated transforms that were learned on a large training set. This suggests that the transform set learned on a specific content is able to better adapt to the content-specific variations that may exist in the residual space. Nevertheless, on-line method was proved, in this work, to be less promising because of the cost of signaling the overhead of basis vectors.

In this chapter, the strength of the above two approaches are exploited to improve the content-adaptability of the off-line learning scheme proposed in the previous chapter without impacting the overhead. A pool-based transform coding scheme is proposed that utilizes multiple transform sets compared to a single transform set as done in Chapter 7 to achieve higher compression efficiency.

First, some related work in the direction of content adaptive transforms is briefly discussed. Then, different methods for the generation of multiple transform sets (or a pool) are presented. Next, method to select and signal the best transform set from a pool of transform sets is described. Finally, some preliminary results for the proposed pool-based coding scheme are presented. The last section covers a detailed discussion on methods that can further be explored to achieve better content-adaptability but without adding the encoder complexity.

8.2 Related works

Content-adaptability using pool-based techniques has not been explored much in the past. However, there are few works that are closely related and have so far focused on adaptation of the transforms at a sequence or a frame level. Dony *et al.* in [58] showed that the optimality of a learned transform on a given image is dependent on the class of images on which the transform is learned. Therefore, the authors proposed to learn a different transform set for different class of images where a class was defined by images with similar semantic meaning such as head MRI image class, body CT image class etc. These off-line learned transform sets were then made available on both encoder and decoder side for coding of images where a set of selected based on the class type.

Xu *et al.* in [92] proposed a content-dependent MDDT scheme where different set of MDDTs were learned for different contents that were classified on the basis of histograms of residuals as a feature extracted at a frame level. For encoding, first a histogram was extracted at a frame level and the corresponding set of KLTs were then chosen based on the closest match. The scheme was tested on AVC and showed gains over the already existing MDDT and RDOT scheme.

Although the principle of using multiple transform sets is related to the previous works, our proposed pool-based method is different as the optimal set is selected at a local region level instead of a frame or sequence level. It is motivated from two main observations. First, methods in the literature were tested only on low-resolution sequences where the statistics across the whole frame vary less frequently and therefore, it was easier to find a single transform set that was optimal for the whole frame. However, this is sub-optimal for high resolution sequences such as HD and 4K. So selecting a transform set at region level is expected to provide better adaptation. Second, for the methods proposed so far, the criteria to choose a set of transforms from multiple sets was not based on a rate-distortion criterion. However, in this method, a set that provides better rate-distortion balance is chosen and the chosen set is signaled to the decoder using an index.

Moreover, in the previous chapters, the trade-off between

– increasing the number of transforms to better adapt to the local statistics of the content, and
– keeping the extra transform signaling syntax as low as possible

has been hard to find. Of course, more transforms lead to better adaptability but also to more signaling. It has been shown that adding a few adaptive transforms is beneficial in term of compression performance. However, adding many transforms may decrease the performance because the extra signaling eats all the gain obtained from the better information compaction into the transformed coefficients. Thus, the pool-based method tries to improve the balance between the number of transforms versus the cost of transform signaling.

This method utilizes several transform sets instead of a single transform set to encode each frame of a sequence. These transform sets are learned on different training sets that are separated from each other on a semantic level.

8.3 Generation of a pool of multiple transform sets

In chapter 7, the effect of choosing a different training set on the overall coding gain was illustrated in Table 7.2. It was shown that the transform set learned from its corresponding training set is either biased to certain content types or generic depending on the training set. For example, the transform set learned on the training set TSET01 (Table 7.2) over-fitted the training set and therefore, did not provide good results on sequences outside the training set. But the transform sets learned on TSET02 or TSET04 (Table 7.2) were generic enough to achieve high coding gains on sequences outside the training set.

This essentially arises a question of transform specificity versus transform generality. A transform specific to a certain type of content may be able to exploit within-class correlations that may exist at residual level. This was shown in Chapter 5 where on-line methods provided much higher gains when self-learned. However, a generic transform may be applied to any content and exploits the residual correlation that does not depend on the content. This could be some kind of probable same-pattern failure of the prediction that exist at a block (or TU) level. Chapter 7 showed a method to exploit such correlation and demonstrated solid gains.

One method to further improve the off-line method that utilizes a generic transform set is to add few transform sets that are learned on a specific content so that it is possible to exploit both specific and generic correlations present in the content. Therefore, six different content types are selected that are semantically different from each other. The sequences within each training set are chosen such that they are semantically related and have similar pixel-level statistics. Table 8.1 illustrates the different training sequences under each set.

Training Set	**Training Sequences**
football	football, soccer, touchdownpass [93]
indoor	indoor data set [94]
building	zurich building data set [88]
landscape	landscapes (from flicker)
BBB	Big Buck Bunny sequence [93]
ED	Elephants Dream [93]

Table 8.1: Different training Sets used for learning multiple transform sets

Another method to generate multiple transform sets consists of learning different number of transforms K per intra-prediction mode on a given training set where K $\in \{1 \cdots L\}$ and L is the maximum number of transforms allowed per intra-prediction mode. Intuitively, these transform sets may be used for encoding a sequence such that the less textured or uniform regions are coded using transform set with smaller value of K and the heavy textures regions are coded using transform set with larger value of K.

In this chapter, the six transform sets generated from sequences in Table 8.1 are used while demonstrating the results. In the next section, the method for selection and signaling of the transform set from the generated pool of transform sets is described.

8.4 Proposed pool-based transform coding scheme

The main idea of the proposed pool-based transform coding method is to split the transform signaling syntax in two levels as follows:

– select at a region level, from a pool of sets of transforms, which set of transforms to be used for this region
– decide at a local area level, from the selected set of transforms, which transform to be used for one or a group of blocks.

Such an approach is advantageous because it allows maintaining the usage of many transforms to keep high adaptability to the content and, at the same time, keeping the transform signaling low by putting the common part of signaling at the region level.

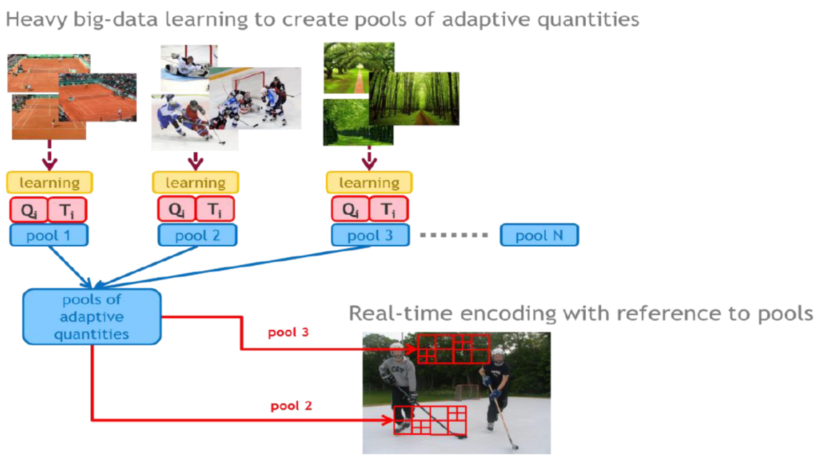

Heavy big-data learning to create pools of adaptive quantities
learning
learning
learning
Q_i
T_i
Q_i
T_i
Q_i
T_i
pool 1
pool 2
pool 3
pool N
pools of adaptive quantities
Real-time encoding with reference to pools
pool 3
pool 2

Figure 8.1: One of the possible usage of adapted transforms in real time encoding

Figure 8.1 shows the basic idea of the pool-based method where given a pool of multiple transform sets learned off-line on different content types as illustrated in the figure, a transform set is chosen for a given region based on some criteria. This criteria can either be determined using some pre-analysis or can be driven by the rate-distortion cost for coding the region using a particular transform set.

Usually the semantic correlations exist at pixel level but does not necessarily hold true in residual space. Therefore, it might be sub-optimal to utilize semantic information to determine the transform set for a given region. In this work, as a preliminary test to determine the upper bound of the gains for using different transform sets, a full Rate-Distortion (R-D) search is performed and the transform set with minimum R-D cost is chosen for that particular region. Moreover, a region in this work is considered to be equal to CTU of size 64×64. This makes the signaling of the pool index easier. Figure 8.2 illustrates the use of different transform sets for different CTUs within a given frame of a sequence.

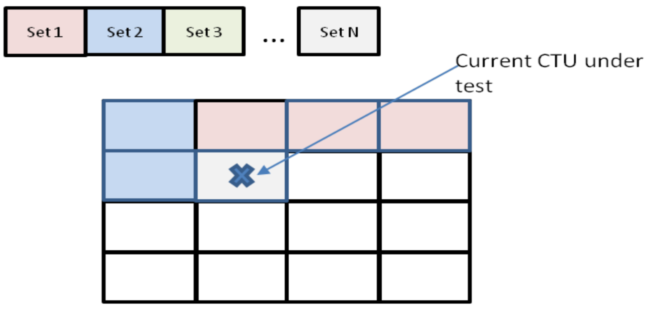


Figure 8.2: Adaptation of transform sets at CTU level

Hierarchical syntax coding

For a given pool consisting of M transform sets, a transform set is selected at a CTU level. Therefore, at CTU level, a pool index of $\log_2(M)$ bits is coded to signal the transform set chosen from the pool. Further, within each CTU, a transform index is coded at TU level to signal the choice of transform from the chosen set of transforms. The signaling method of the transform index is already described in Chapter 7.

The above hierarchical splitting of the syntax provides a way to increase the number of transforms for better adaptation but without adding too much signaling cost. Considering that all transforms within a pool are tested at a TU level where each transform set consist of K transforms per intra-prediction mode, a signaling of $\log_2(\mathrm{M}\times\mathrm{K})$ bits is required. However, with the above hierarchical approach, the TU level signaling cost remains same and only, a syntax is coded at CTU level which has negligible impact on the overall coding gains.

8.5 Simulation Results

This section provides some preliminary results to validate the performance improvements due to the proposed pool-based transform coding method described in this chapter. For the generation of a pool of transform sets, four major categories of content is chosen i.e. indoor scene, two different outdoor scenes (landscape and building), two screen content videos and a football scene. These contents are classified into 6 training sets as illustrated in Table 8.1. A transform set is learned on each training set using IMDTC scheme proposed in Chapter 7. Further, a generic transform set as learned in Chapter 7 is added to form a total of M=7 transform sets. Each transform set consist of K=4 KLTs/IPM for 8×8 and 4×4 size residual blocks. For luma residual blocks of 16×16 and chroma residual blocks of 4×4, a single transform set is learned using a general training set as described in Chapter 7. For the remaining blocks, the conventional DCT is only used.

In the first experiment, each transform set learned on a class of sequences from 8.1 is tested on all other sequences and the BD-rate gains corresponding to each transform set is illustrated in Table 8.2. Clearly, the self-learning consistently out-performs the cross-learning where self-learning refers to learning and testing on the same class of sequences and cross-learning refers to learning on one class of sequences and testing on sequences belonging to another class. This is consistent with our findings in Chapter 5 which proved that an on-line (or self-learning) provides better performance as it can exploit within-class correlations of the content. Moreover, it also proves that despite the residual space issue, semantic is still a promising way to generate different training sets.

Table 8.2: BD-rate gains for using different training sets

	Training sets					
Test Set	Football	indoor	building	landscape	BBB	ED
Football	-3.53	-3.58	-3.06	-2.39	-2.05	-5.84
indoor	-3.51	-6.44	-6.09	-3.04	-2.62	-8.78
building	-2.19	-5.43	-6.78	-2.16	-1.73	-7.31
landscape	-2.74	-3.63	-3.39	-3.07	-1.63	-5.17
BBB	-2.46	-3.24	-2.97	-2.07	-3.7	-5.88
ED	-2.78	-5.44	-5.55	-2.43	-2.52	-10.32

It is further observed in Table 8.2 that the training sets such as ED and indoor are generic as they provide high compression gains for sequences outside their class. But the training sets BBB and Football does not provide high gains for sequences outside their class. This observation can be used to design a pool that consist of a mix of generic and specific transform sets.

In the next experiment, the proposed method is tested within HEVC test software (HM15.0) where a pool designed above is used to encode different CTC sequences. A brute-force test is performed where each transform set from the pool is used to encode the CTU and R-D cost is computed in each case. An additional pass is tested without any pool where only DCT/DST is used.

For signaling, pool index of 3 bits is coded uniformly at CTU level and a transform index is coded at TU level in similar manner as described in Chapter 7. Table 8.3 shows the BD-rate gain of using different pools at CTU level. An overall gain of around 7% is observed over HEVC under All intra (AI) configuration. This is approximately 1.7% higher than the coding gains achieved from a single training set.

The coding gain of 7% comes at a cost of added encoder complexity as the method employs multiple pass at CTU level to decide an optimal transform set for a given CTU. An encoder complexity is approximately 7-8 times the complexity of IMDTC scheme proposed in Chapter 7. However, this complexity can be significantly reduced if the decision of optimal transform set is taken outside the R-D search loop. One such method is to use hand-crafted features (also defined by Xu *et al.* in [92]) to find the closest match between the characteristics of a CTU and the dictionary of transform sets. Another method that can be explored in future is to use machine (or deep) learning models (similar to method illustrated in Chapter 6 section 6.4.2) to extract features that can help in achieving an optimal classification.

8.6 Conclusions and Perspectives

In this chapter, a method is presented to improve the content-adaptability of the proposed IMDTC scheme by using a pool of transform sets that are learned on different content types. This

Table 8.3: BD-rate gains for using a pool of 7 different transform sets and 1 set consisting of only DCT/DST

Class	Sequences	Y	U	V
A	Nebuta	-2.42	-4.73	-3.2
	PeopleOnStreet	-7.59	-6.31	-5.35
	SteamLocomotive	-2.84	1.33	1.29
	Traffic	-8.14	-3.81	-5.2
	Overall	-5.25	-3.38	-3.12
B	BasketballDrive	-7.83	-3.05	-4.31
	BQTerrace	-6.23	-6.61	-5.78
	Cactus	-7.19	-3.36	-4.49
	Kimono	-4.13	-1.68	-2.6
	ParkScene	-5.76	-1.45	-2.57
	Overall	-6.23	-3.23	-3.95
C	BasketballDrill	-15.38	-10.23	-10.96
	BQMall	-7.25	-3.45	-3.48
	PartyScene	-6.04	-3.29	-4.31
	RaceHorses	-7.08	-3.65	-6.04
	Overall	-8.94	-5.15	-6.2
D	BasketballPass	-7.59	-4.88	-5.66
	BlowingBubbles	-6.84	-4.6	-5.15
	BQSquare	-6.22	-2.66	-3.85
	RaceHorses	-7.95	-5.94	-7.06
	Overall	-7.15	-4.52	-5.43
E	FourPeople	-7.95	-3.84	-4.27
	Johnny	-6.7	-4.89	-4.13
	KristenAndSara	-7.49	-4.4	-4.79
	Overall	-7.38	-4.38	-4.4
Overall		**-6.99**	**-4.13**	**-4.62**

chapter presents some preliminary results in order to prove the merits of the proposed scheme that utilize multiple transform sets with small number of transforms instead of a single set with large number of transforms. An overall gain of 7% confirms the potential of this approach to achieve better compaction. However, there are some concerns that are not yet explored in this work and are important for achieving even higher gains at reduced complexity.

First, the encoder complexity of the proposed approach is very high as it requires multi-pass at CTU level to choose among different transform sets. As briefly discussed before, the encoder complexity can be significantly reduced by methods that helps in predicting an optimal set for a given region (or CTU). As encoder decisions are driven by numerous factors, it is not easy to find features to achieve high classification accuracy in predicting correctly the optimal set. Deep learning methods have proved so far to be effective in extracting correlations that are difficult to find otherwise using conventional hand-crafted features. A method in Chapter 6 has already shown the merits of using convolutional Neural Networks (CNN) for classification. Similarly, this can be applied here to predict the pool index in order to speed-up the encoder.

Second, the coding gains shown in this chapter are based on a pool of transform sets that are manually chosen and are expected to be semantically unrelated. However, semantic relationship is not straight-forward in case of residuals as much of the correlation is removed in the prediction step. Therefore, the blocks with specific edges and textures may characterize a class of sequences from others but in general, other blocks are correlated across many sequences. Therefore, different methods to generate a pool may be explored in future that takes into account the local residual statistical variations of a class of sequences along with the semantics in order to differentiate between different classes.

Even with diverse set of training sets, there might be some redundancies in terms of the basis functions derives from learning on different training sets. Thus, finding a distance between two transforms might help in generating a diverse set of transforms. Methods to compute such distance metrics will be explored in future.

Finally, the pool-based method has shown a great potential where a simple pooling approach as presented in this chapter led to a significant gain over the IMDTC scheme. Therefore, it is worth exploring the full potential of this approach using above described methods.

IV

Conclusion and Future Perspectives

9 Conclusions and Future work

First, a summary is presented wherein the important contributions made in each chapter are highlighted. Second, some important observations made during this work are presented. Finally, some possible future directions are presented in the last section.

9.1 Summary

A review of advanced transforms

In this thesis, an extensive review was carried in Chapter 3 on the advances in the transform coding methods carried in the last decade. The different methods proposed in the past were shown to classify into three categories depending on whether the new transforms were 1) already known transforms, 2) learned off-line on a data set or 3) learned on-the-fly for a given content.

It was shown that the recent state-of-the-art methods proposed in context to HEVC and the recent standardization activities carried for next generation video coding technology employ multiple transform candidates instead of a single transform to achieve higher compression gains. Further it is observed that the most popular technique to achieve higher compression efficiency involves multiple transform competition on a residual block to choose the transform with minimum rate-distortion cost and explicit signaling of the chosen transform to the decoder. Chapter 3 presents such methods where multiple transforms can either be systematic known transforms or be data-driven transforms where they are learned off-line on a large training set of intra-predicted residuals.

Analysis of the data-driven transform learning process

The work in this thesis has explored the data-driven transform learning methods in detail where a transform set is learned on a data set. Chapter 4 provided a detailed analysis of the training process employed for the learning of data-dependent adaptive transforms. The appropriability of various classification and transform optimization strategies was discussed. Some new methods for classification and transform optimization were provided in this Chapter. Finally, a comparison was made between different state-of-the-art offline learning methods on the basis of the employed training process for transform learning.

Comparison of on-line and off-line methods to study the feasibility of each approach

So far the data-driven methods were learned off-line on a large training set and therefore, were not able to exploit the content-specific correlations. Therefore, in order to exploit the content-specific correlation that may exist in the residual space, an on-line adaptive transform learning scheme was proposed in this thesis and was described in Chapter 5. The scheme adapted the existing off-line learning scheme such that the classification of the residual blocks was done using the HEVC coding cost instead of an approximative ℓ_0-norm. Therefore, a two step iterative scheme was devised where at first step, the residual blocks within a given frame or a sequence were classified using HEVC R-D cost and then, at second step, an optimal transform was learned for each class that minimized the error between the residuals and the reconstructed residuals for a given quantization parameter. The results showed an improvement of 1-2% over standard HEVC under all intra (AI) configuration.

Further, two novel methods were proposed in chapter 5 in the context of proposed on-line method for improving the stability of the learning scheme itself. The first proposed method used an annealing-like strategy for the transform index coding during the transform learning process where an annealing parameter ϵ was applied to the transform index rate term and was uniformly increased from 0 to 1 during first few iterations. This helped in slowly penalizing the cost of transform index in the optimization loop and showed the merits of jointly optimizing the transform and its syntax. The second proposed method employed smart coefficient re-ordering inside the learning scheme to improve the transform optimization step. The unitary gains of each of the above proposed methods were demonstrated in Chapter 5.

Finally, the last part of Chapter 5 compared the proposed on-line method with an equivalent off-line method where same number of transforms were learned on a large training set. A detailed comparison of coding gains, complexity and side information was made between the two schemes. It was shown that the online method slightly outperforms in terms of coding gains but are significantly complex compared to off-line method. Moreover, it was shown that the overhead of signaling the basis vectors limits the use of higher number of transforms for on-line methods, thus making it less promising. On the other hand, offline methods were shown to have less content-adaptability. It was also shown that the overhead of signaling the choice of transform was similar in both on-line and off-line methods.

Deriving efficient method to signal the side information in the context of on-line and off-line methods

One major limitation of the on-line method proposed in Chapter 5 was the heavy overhead of signaling of transform basis vectors to the decoder for successful reconstruction of residual block at the decoder. This problem was addressed in Chapter 6 of this thesis and two methods to reduce the overhead of transmitting the transform basis were presented in this work. The first method aimed to generate partial transforms where a method was devised to adaptively determine the number of transform vectors k to be coded based on the energies of the transformed and quantized coefficients. In the second method, a statistical model was computed to adaptively determine the value of precision drop b for each content at each QP. It was shown that by employing the proposed modifications, the overhead can be reduced to 14% of its original size at low bit-rate and 28% of its original size at high bit-rate.

Further, the modification of the last significant coefficient position inside HEVC was proposed in the context of partial transforms and was presented in Chapter 6. The unitary gains because of this improvement were presented and showed around 0.3% coding gains.

A yet another contribution in the context of improving the transform index coding was presented in the second part of Chapter 6. The method aimed to partially infer the transform index from the decoded quantized transform coefficients. As the correlation between the coded quantized transform coefficients and the transform index is hard to find, the neural network based models were used to extract the features in order to achieve better classification accuracy. It was shown through experiments that a correlation to some extent exists between a coded quantized coefficient block and the transform index, hence providing a basis to partially infer from the quantized coefficients. The proposed method showed a gain of around 0.4% compared to the off-line method described in Chapter 5 that used conventional signaling method.

Further, many methods proposed in this Chapter are general enough to be applied to both online and offline methods as well as to other schemes. For example, it was shown in Chapter 6 that the transform index prediction method using neural network can also be extended to the prediction of other syntax elements such as transform skip flag etc.

Improving the existing state-of-the-art off-line learning scheme

In this thesis, the on-line and off-line methods were compared. It was shown that off-line methods offered an advantage in terms of lower overhead cost compared to on-line. The second part of this thesis focused on optimizing this specific view point i.e. off-line method by proposing the two axis of improvements which were presented in Chapter 7 and 8. The work in Chapter 7 focused on improving the existing state-of-the-art Mode Dependent Transform Competition (MDTC) scheme that was proposed in the literature. A brief analysis on the choice of training set was provided and the effect of using different training sets on the final coding gains was demonstrated. The effect of using biased and non-biased training set on the final R-D performance over the test set was briefly investigated. It was shown that the overlap between the training and test set could affect the overall gains obtained on the test set and therefore, an independent training set was designed with an aim to generate generic transform set which is not biased towards certain content type or resolution.

Further, one of the key contributions of this thesis was to propose an improved MDTC scheme (IMDTC) for HEVC which employed a series of improvements over the existing MDTC scheme. An iterative approach to learn a transform set on the training sequences was proposed where a training set and a transform set was iteratively computed until convergence. Further, the MDTC method was extended to chroma blocks of 4×4 and 8×8 size as well as to luma blocks of 16×16 size. An overall gain of around 5.4% over standard HEVC under all intra (AI) configuration was provided at an encoder complexity of 3.9 times standard HEVC. The coding gains were shown to be competitive with the existing MDTC method but at much less encoder complexity. In order to reduce the encoder complexity, a low-complexity IMDTC was further proposed in Chapter 7 that utilized partial transforms for 16×16 and 8×8 residual blocks. The last significant coefficient position coding was also modified as described in Chapter 6. The results showed a coding gain of 5.3% with an encoder complexity of 2.9 times the standard HEVC.

Finally, the results achieved using low-complexity IMDTC showed that it is feasible to use non-separable multiple transforms for the future video coding technologies. Moreover, the proposed method is not optimized at all and therefore, the coding complexity can be easily reduced by introducing some pre-analysis steps to speed-up the encoder decisions. Also, the matrix multiplication of the non-separable transform and the residual signal can also be parallelized on a multi-core architecture to achieve further encoding/decoding time reductions.

Development and testing of pool-based transform coding scheme to improve the content adaptability of the off-line methods

The chapter 8 of this thesis proposed to optimize the off-line methods learning multiple transform sets instead of a single transform set. This second axis aimed at improving the content-adaptability of the off-line methods. Some methods to generate these multiple transform sets were presented. For coding of a given frame, an optimal transform set was selected at a region level (here a CTU of size 64×64) based on the minimum coding cost at a CTU level. An index is further coded at a region level to indicate the choice of a training set among multiple sets. The preliminary results showed a gain of around 7% over the standard HEVC for using 7 different transform sets and therefore, showed potentials of achieving even higher compression gains with more sophisticated approaches on generating pool of transform sets and signaling of transform index at different region levels. This method certainly opens new doors to the future exploration of relationship between the various contents, regions within a given content and the multiple transform sets so that an optimal transform set can be determined without performing expensive R-D search.

In the next section, some of the key learnings from this research work are described.

9.2 What we have learnt

9.2.1 h.264 vs. HEVC

In the literature, many have proved decent or even solid RD improvement over h.264 by using new transforms different from the usual DCT. Even some simple learning scheme based on the KLT, and ignoring the actual bit-rate computation process like CABAC or CAVLC, could show some several-percent gains. For instance, the MDDT, introduced in the KTA test model as a successor of h.264, was very successful. It is simply an off-line KLT learning on residual blocks classified by their intra prediction directions. Sadly, when put into HEVC, most of the gains of the MDDT vanish and the technology is no longer promising for future standard. We observed a similar failure when we tried to apply a simple KLT in our learning scheme.

Clearly, HEVC is more demanding than h.264 in term of transform adaptation. It is pretty obvious to explain: the residual blocks have much less pixel correlation in HEVC than in h.264. Basically, there is less to learn in term of residual pixel structure because the prediction itself has become more efficient (more directions, smoothing, etc.). Also the quad-tree structure adds more flexibility in the block topology, thus allowing a better adaptation to the content edges and texture. One must not forget that transforms are not adapted to the content, but to the content residual instead. If the prediction works so well that the residual is just pure noise, then there is nothing to learn and adpative-transform-based technologies are simply promised to fail dramatically. Of course, real-world codec are far from such a powerful prediction, but HEVC has without a doubt closed part of the gap and has made our work more difficult than in a h.264 framework.

9.2.2 Online vs Offline

We have shown in Chapter 5 that heavy adaptation of the transforms to one image can lead to interesting compression gains. Of course, this leads to impractical encoding schemes that induce a large number of encoding passes but it exhibits some potential: quite a lot of correlation still exists in the residual blocks. Also, the lack of convergence partially jeopardizes the potential gains even though how much a good-converging iterative algorithm could give us is still an open question. Moreover, the overhead of signaling the basis vectors in online methods further limits the number of transform candidates that can be used. This overhead can be reduced by using separable transforms instead of the non-separable ones but this also leads to lower coding gains. One may argue that pushing further in this direction is worthless as a good convergence is still hard to find and the encoder complexity is simply unrealistic in practical applications. This clearly makes off-line methods a better approach where the heavy computation of transform learning is done off-line and the learned transforms are made available on both encoder and decoder side as a dictionary. Moreover, the size of the dictionary can easily be scaled as done in Chapter 8 where several transform sets are combined to form a pool.

9.2.3 Penalization and rate models

Transforms should not only adapt to the content residual but also take into account the RD balance; a codec does not chose a coding mode based on the best distortion but on the best Lagrange cost. Forgetting this balance leads to a KLT model that is only based on the pixel statistics. However, transformed coefficients are quantized, coded and their coding cost is put on the balance in the RDO loop through the Lagrange cost. Some authors have improved the learning scheme and proposed learning model with penalization as follows, introducing a penalization factor λ.

$$T = \min_{T'} \|r - T'c\| + \lambda p(c)$$

A simple way to do it is to use the $\ell 0$-norm as the penalization function p. This is simple because mathematically easily trackable. This leads to so-called sparse coefficient and sparse learning models. The $\ell 0$ penalization has led to good improvement of the learning scheme as shown for the MDST. It is wrong to state that the $\ell 0$-norm is nowhere related to the h264/HEVC coding process of transformed coefficients because a significance flag, carrying the information of zero vs. nonzero coefficient, is actually coded and the $\ell 0$-norm is a perfect model for this flag. The optimality of the ℓ_0-norm has been proved in the context of learning method in []. However, an optimality on training does not necessarily prove optimality on the test set within an RDO loop. Therefore, a penalization function closer to HEVC rate is expected to be better adapted to the specific RDO of the codec.

On-line methods can potentially be used to replace approximative ℓ_0-norm with the real HEVC rate but may lead to even much higher complexity. One solution is to find a rate model closer to HEVC that can be employed off-line to learn transforms better adapted to HEVC. This direction may be explored in the future.

9.2.4 Transform genericity versus specificity

A generic transform is the one that provides good coding gains over a wide range of sequences. An example of a generic transform is the DCT transform which provides good approximation of the optimal KLT in case the residual signal is highly correlated with its neighbouring pixels in either horizontal or vertical direction. These kind of correlations does not depend on the content and therefore, makes the DCT a generic transform. Similarly, the methods using offline learned transforms aim to find similar residual lagrange (not only pixel!) correlation patters that does not depend on the content.

On the other hand, a specific transform is the one that provides good coding gains only on some specific content or a specific area of a given content. One method to generate a specific transforms was stated in Chapter 5 as on-line method where self-learning on a single frame was performed. Also, it was shown in Chapter 7 and 8 that an overlapping training and test set might also lead to specific transforms in some cases. This really depends on the type of content. Having said that, the specific transform sets learned on different contents may be advantageous in improving the content adaptability of the transforms. This was briefly explored in Chapter 8. However, it would be interesting to explore how far can go in terms of achieving transform specificity for a certain content type and what is the upper bound of the gains one can achieve using such methods.

In the next section, some of the possible future works are described.

9.3 Future Work

The methods proposed in this thesis have shown to achieve significant compression gains over HEVC with a decent complexity addition. Many methods have provided a strong basis for the future explorations in order to achieve even higher coding gains. Therefore, the work presented in this thesis can be further extended in many directions.

First, the pool-based methods proposed in Chapter 8 still need further investigation. For example, in this work, a rate-distortion cost criterion was used to choose an optimal transform set from a pool of multiple transform sets. This led to an expensive coding scheme where multiple coding passes have to be performed at a region level, thus jeopardizing the practicality of such schemes. However, the complexity can easily be reduced by first employing some fast decisions where the rate-distortion search can be skipped for some regions (or CTUs) based on complexity of the structure within that region. Moreover, a classification method can also be used to pre-determine an optimal transform set from the pool. These classification methods may use some hand-crafted features extracted at the residual level or may be based on the more sophisticated deep neural network methods similar to the one proposed in Chapter 6 for index prediction.

Second, the effect of different training sets on the coding gains was briefly studies in Chapter 7 and 8. It was shown that for some training sets, the learned transform sets are specific to a given content type and for some other training sets, the learned transform sets are generic enough to show gains on different content types. Therefore, finding a criterion or a feature that leads to the transform specificity versus transform generality for different training sets needs to be explored. The future work will aim to devise a method to find closeness between the different training sequences in order to determine an efficient pool.

Third, in context to the learning scheme itself, different classification and transform optimization techniques have been described in Chapter 4. In the on-line method, we used the true rate distortion cost to do the classification. Similarly, in the off-line method, one may use a rate model within the rate-distortion cost function to achieve classification which is closer to the classification performed within HEVC. For the transform optimization step, the above rate model can again be used to determine a transform set that is optimal in the true rate distortion sense. Future work may aim at learning non-orthogonal transforms by relaxing the orthogonality constraint within the optimization equation as described in Chapter 4. However, an inverse transform need to be computed for a non-orthogonal transform that is required at the decoder. In [28], an optimized non-orthogonal transform scheme was studies for the image compression application.

The cost of coding the transform index is shown to have considerable impact on the overall BD-rate gain and therefore, a method to reduce this index cost was explored in Chapter 6 where the index was partially inferred from the decoded quantized coefficient block. The method showed some gains but still there is huge potential to achieve even higher coding gains. It was also shown that the gains depend on the accuracy of the prediction model. Therefore, the future work will first explore some methods to improve the accuracy of the existing prediction model. This can be done by using other causal information present at the decoder as an input to the CNN-based model. The other method to reduce the signalling cost would require using a hybrid approach that signals a syntax at CU or CTU level instead of signalling at a local block level. By employing such methods, the encoder adapts automatically to choose low block level signalling for regions with less structure and heavy block level signalling for regions with high structure or edges. One such method has been explored in [1].

Finally, the methods proposed in Chapter 7 and 8 use non-separable transforms that require more number of multiplication and addition operations compared to its separable counter-part. Though the overall encoder time-complexity of the proposed low-complexity IMDTC is 2.9 times the standard HEVC, the matrix multiplication is easily parallelizable and could be for example implemented on multi-core CPUs to achieve even less encoder time complexity. Apart from the complexity, the memory required to store the non-separable transforms is higher than its separable counter-part but at the same time, the non-separable transforms provide advantage in terms of higher coding gains. One method to effectively find a trade-off between the two is by deducing a row-column transform from the non-separable transform as proposed in [95]. Further, the precision of the transform coefficients can also be adaptively reduced to further reduce the memory footprint.

A

Smart re-ordering of coefficients during learning

In chapter 5, the proposed smart re-ordering method is briefly described that re-orders the coefficients during learning to further stabilize the learning scheme. Here, the proposed improvements are discussed in more details. First the motivation behind the smart re-ordering approach is described and then the proposed smart re-ordering methods are presented. Finally, the unitary gain obtained thanks to the proposed approaches is demonstrated using experiments.

Motivation

HEVC utilizes DCT/DST to transform a 2-D residual block of size N×N into a 2-D coefficient block (CB). The coding process of this CB is detailed in Chapter 2 section **??**. The entropy coder CABAC has been designed to achieve better compression and high throughput capacity. In general, CABAC expects to have small coefficient levels at higher frequency and high coefficient levels at low frequencies. Therefore, the transforms are required to generate the coefficients in the suitable magnitude order to assist the entropy coding using CABAC.

The online learning scheme proposed in Chapter 5 classifies the residual blocks based on the actual coding cost inside HEVC, which is determined by coding the coefficients using CABAC. For each class, an optimal transform is then determined by minimizing the distortion between the residuals and the quantized coefficients generated by HEVC. The transforms learned from the above iterative scheme do not guarantee that the coefficients are in a good order. Therefore, to improve the stabilization of the learning scheme, the transforms must be optimized such that the coefficients generated from these transforms are more CABAC friendly. This can be done by re-ordering the coefficients prior to the transform optimization step.

However, one notes that this is not required in case of offline methods (as described in Chapter 5) where the rate is approximated by the number of non-zero coefficients. The transform optimization step in this case just focuses on learning a sparse transform.

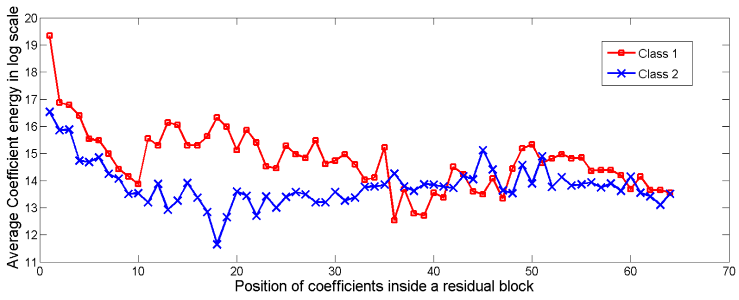


Figure A.1: Average coefficient energy (log scale) vs position inside block of size 8×8

Figure A.1 shows the average coefficient energy (in log scale) computed using sum of absolute magnitude at each frequency channel of an 8×8 coefficient block after the first iteration for two different classes. It is observed from the plot that the average coefficient energy of the coefficients in each cluster S_i is not necessarily sorted in the decreasing order. This shows that the newly computed transforms are not optimized to generate coefficients in a good order. Therefore, it is necessary to perform efficient re-ordering of coefficients inside each class in order to compute an optimized set of transforms for the next iteration.

In the next sub-section, a modified transform optimization step is presented. The proposed modification computes optimized transforms that generate coefficients which are better adapted to the coefficient coding scheme in CABAC. The proposed modifications are described in context to the online learning scheme, but these modifications can in general be applied to any transform learning scheme where the rate is determined using a model closer to the HEVC entropy coder.

Improved Transform Optimization by Smart Coefficient Re-ordering

The proposed modification involves first computing an order O_i specific to each class i and then sorting each of the coefficient vector inside said class.

As described above, it is more efficient to encode the high coefficient levels at low frequencies and vice-versa. This can be achieved by re-ordering the coefficients in the decreasing order of energy from low frequency to high frequency position inside a coefficient block. This enables the efficient use of the HEVC entropy coding engine for coefficients generated from adaptive transforms without any further modifications. Figure A.2 shows the modified online transform learning scheme in which the newly added blocks are shown in red. The classification step remains unchanged. However, the input to the transform optimization step has been modified and two additional steps are performed before the computation of optimized transforms.

These steps involve computation of an order for each class and then performing the re-ordering of each coefficient inside a class based on the computed order. Two different methods for re-ordering have been proposed in this work. In the first method, the sum of coefficient energies E_i

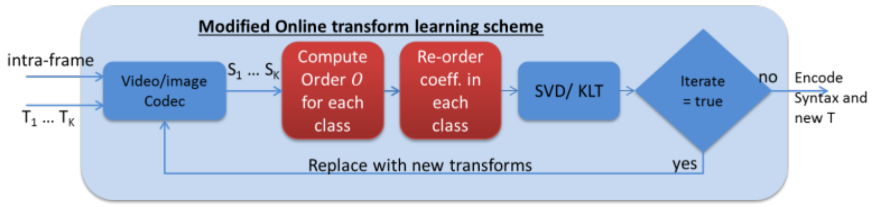

Modified Online transform learning scheme
intra-frame
$T_1 \dots T_K$
Video/image Codec
$S_1 \dots S_K$
Compute Order O for each class
Re-order coeff. in each class
SVD/ KLT
Iterate = true
no
Encode Syntax and new T
yes
Replace with new transforms

Figure A.2: The proposed modified online transform learning scheme including re-ordering

at each channel i is computed for the whole class,

$$E_i = \sum_{j \in S} \|c_i^j\|_2^2, \tag{A.1}$$

and an order O is determined by sorting the energies in the decreasing order. Next, each of the coefficient vector c^j inside the class is re-ordered based on the order O and the final optimized transforms are obtained by minimizing the reconstruction error between the residuals and the corresponding re-ordered coefficient vectors. This re-ordering is performed for each class and at every iteration inside the learning scheme.

The above proposed method is based on the ℓ_2-norm which might suffer from the outliers. Therefore, in the second method, the re-ordering is further improved by computing an entropy of coefficients at each channel based on the Generalized Gaussian Distribution (GGD) Model. In general, a GGD is represented as

$$GGD = \beta exp(-|x/\alpha|^{\beta})/2\alpha\Gamma(1/\beta) \tag{A.2}$$

and is defined by two parameters α and β. These two parameters are determined for each channel i of coefficients inside a given class. Finally, the entropy H_i at each channel i is computed as

$$H_i = 1/\beta - \log(\beta/2\alpha\Gamma(1/\beta)) \tag{A.3}$$

Finally, an order O is determined by sorting the entropies in the decreasing order and steps similar to the first method are followed to compute optimized transforms. The performance obtained thanks to these two modifications is presented in the result section below.

Simulation Results

In this section, the BD-rate gain coming from different coefficient re-ordering methods as described above is presented. The online transform learning scheme implementation in HEVC test model (HM15.0), as described in Chapter 5, is used for this experiment where the initial diagonally-oriented DCT-like transforms are used. Also, the annealing-like index coding technique is employed to stabilize the learning scheme. This provides an anchor to be compared to.

In the experiment presented here, the online transform learning scheme as shown in Figure A.2 is used where for each class, the proposed re-ordering is employed before the transform optimization step. This is marked in Figure A.2 by blocks in red. The re-ordering is performed at each iteration of the learning process and optimized transforms are generated.

These final learned transforms are then tested on the first frame of each sequence and the BD-rate gain is presented in Table A.1 without considering the overhead cost. In this work, two different methods of re-ordering have been proposed and therefore, the results of each of the method is presented in Table A.1 under **ATs + ℓ_2 re-order** and **ATs + GGD re-order**.

Table A.1: BD-rate gain on first frame for different re-ordering

Class	**ATs**	**ATs+ℓ_2 re-order**	**ATs+GGD re-order**
A (2560×1600)	-3.26%	-4.34%	-4.33%
B (1920×1080)	-2.37%	-2.83%	-2.86%
C (832×480)	-5.93%	-6.36%	-6.52%
Overall	**-3.85%**	**-4.50%**	**-4.60%**

The result are compared to the unmodified online transform learning scheme where the re-ordering is not applied. The results shows that the ℓ_2-norm based re-ordering provides an overall improvement of 0.75% over the unmodified learning scheme and a gain of 4.50% over standard HEVC. Further, the GGD based re-ordering shows a further improvement of 0.1% over the first case and the overall gain is around 4.6% over HEVC.

B

Statistical model to determine precision drop value b

The basic idea behind this content-adaptive statistical model is to find a bound on error induced by quantization of transform vector such that the variance of the error induced due to the transform quantization is less than the variance of the error induced due to the scalar quantization of transformed coefficients as performed in a video/image codec. Let the variance of error due to quantization of coefficients in a codec is denoted by $\mathrm{D}(\Delta_q)$ and the variance of error due to the quantization of transform $\mathbf{T}$ is denoted by $\mathrm{D}(\Delta_g)$.

Quantization of Transform

The drop of precision can be seen as quantization of the transform where a step size of the quantizer Δ_g is shown as $\Delta_g = 2^b$. Let $\mathbf{T}$ be a non-separable adaptive transform, obtained for instance by a learning algorithm as described above. Let $\mathbf{G} = \mathrm{g}(\mathbf{T})$ be the transform obtained after the quantization of $\mathbf{T}$, where $\mathrm{g}(\cdot)$ is defined as a quantization function which depends on the step Δ_g as follows

$$\mathbf{G} = \operatorname{sgn} \mathbf{T} \cdot \lfloor \frac{\mathbf{T}}{\Delta_g} + 0.5 \rfloor \cdot \Delta_g$$

Let $\mathbf{r}$ be a residual block (straightforwardly put under a vector form to get obvious matrix notation) to be transformed and quantized in order to be compressed by the video codec. The associated transformed coefficient blocks $\mathbf{c}$ and $\mathbf{c}'$ after transformation by $\mathbf{T}$ and $\mathbf{G}$ respectively are

$$\mathbf{T} \cdot \mathbf{r} = \mathbf{c}$$

$$\mathbf{G} \cdot \mathbf{r} = \mathbf{c}'$$

The average variance of the error in the transformed domain is defined by

$$\sigma_g^2 = \frac{1}{N}\mathrm{E}[(\mathbf{c}' - \mathbf{c})^2] = \frac{1}{N}\mathrm{E}[(\mathbf{c}' - \mathbf{c})^T(\mathbf{c}' - \mathbf{c})]$$

$$\sigma_g^2 = \frac{1}{N}\mathrm{E}[(\mathbf{G}\cdot\mathbf{r} - \mathbf{T}\cdot\mathbf{r})^T(\mathbf{G}\cdot\mathbf{r} - \mathbf{T}\cdot\mathbf{r})]$$

where N is the size of transform vector. A quantization error matrix $\mathbf{\Lambda}$ is defined as $\mathbf{\Lambda} = \mathbf{G} - \mathbf{T}$. Therefore, the error variance equation is written as

$$\sigma_g^2 = \frac{1}{N}\mathrm{E}[(\mathbf{\Lambda}\cdot\mathbf{r})^T(\mathbf{\Lambda}\cdot\mathbf{r})] = \frac{1}{N}\mathrm{E}[\mathbf{r}^T\mathbf{\Lambda}^T\mathbf{\Lambda}\mathbf{r}] = \frac{1}{N}\mathrm{E}[\mathbf{r}^T\mathbf{M}\mathbf{r}]$$

where $\mathbf{M} = \mathbf{\Lambda}^T\mathbf{\Lambda}$. An upper bound for the above equation is easily found to get

$$\sigma_g^2 \leq \frac{1}{N}\sigma_r^2 \cdot \sum_i\sum_j |M(i,j)| \tag{B.1}$$

As each value in the matrix $\mathbf{\Lambda}$ is proportional to the step size Δ_g and lies between (-$\Delta_g/2$, +$\Delta_g/2$), the sum of absolute values in $\mathbf{M}$ is proportional to the step size such that

$$\sum_i\sum_j |M(i,j)| = \gamma\cdot\Delta_g = \gamma\cdot 2^b$$

where γ is a parameter computed experimentally for each transform. Therefore, an upper bound of the variance due to the quantization of the transforms is given as

$$D(\Delta_g) = \frac{1}{N}\sigma_r^2\cdot\gamma\cdot 2^b$$

Quantization of transformed coefficients

The coefficients obtained after transformation of the residual block are quantized using a uniform scalar quantifier Q(·), with a dead-zone, whose step size Δ_q depends on the quantization parameter (QP) which may be chosen at the frame level or at the block level using so-called adaptive QP. The mean-squared error (MSE) due to the quantization of a coefficient is

$$D(\Delta_q) = \int_{-(\Delta_q - o\Delta_q)}^{(\Delta_q - o\Delta_q)} x^2 p(x)dx \,+\, 2\sum_{i=1}^{\infty}\int_{i\Delta_q - o\Delta_q}^{(i+1)\Delta_q - o\Delta_q}(x - i\Delta_q)^2 p(x)dx$$

where x denotes the coefficient values, o is the rounding offset, and p(x) denotes the probability distribution of the coefficient. It is known from the literature that the coefficients obtained from DCT-like integer transforms follow more or less a zero-mean Laplace distribution defined by

$$p(x) = \frac{\lambda}{2}\mathrm{e}^{-\lambda|x|}$$

where λ is the Laplacian distribution parameter related to the standard deviation σ of the coefficients by $\lambda = \sqrt{2}/\sigma$. Substituting the above formulation of p(x) in the MSE equation provides an expression in terms of λ and Δ_q. This suggests that the variance of quantization error depends on these two parameters and, therefore, the expression can be simply expressed as

$$D(\Delta_q) \approx \alpha(\Delta_q)^\beta$$

where the two parameter α, $\beta > 0$ depend on λ which in turn depends on the content. The expression shows that there is an exponential relationship between the quantization error variance and the quantization step-size.

In order to minimize the effect of the change in transformed coefficient values due to the quantization of the transform, the variance D(Δ_g) should be kept strictly less than the variance of the error induced due to the scalar quantizer present in a video/image codec. Therefore, the relation between the two variances D(Δ_g) and D(Δ_q) as obtained above is shown as

$$D(\Delta_g) < D(\Delta_q) \tag{B.2}$$

Substituting the expressions from above, the relation can be shown as

$$b < \log_2 \left(\frac{N \cdot \alpha}{\sigma_r^2 \cdot \gamma} (\Delta_q)^{\beta} \right)$$

A relationship between the precision drop b and the quantization step size Δ_q, which is related to QP, has been deduced. The parameters α, β and σ_r^2 are content dependent and the parameters γ and N are related to the transform itself. Therefore, these parameters are derived for each content in order to determine the accurate value of precision drop b.

Bibliography

[1] X. Zhao, J. Chen, M. Karczewicz, L. Zhang, X. Li, and W.-J. Chien, "Enhanced multiple transform for video coding," in *Data Compression Conference (DCC), 2016.* IEEE, 2016, pp. 73–82. 7, 9, 38, 39, 40, 90, 141

[2] J.-R. Ohm, *Multimedia Communication Technology.* Springer, 2003. 9, 19, 20

[3] G. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," 2012. 9, 21, 22, 23, 24, 25, 26, 34

[4] M. Wien, *High Efficiency Video Coding – Coding Tools and Specification.* Berlin, Heidelberg: Springer, Sep. 2014. 9, 26

[5] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the h. 264/avc video compression standard," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 13, no. 7, pp. 620–636, 2003. 9, 27

[6] G. Bjontegard, "Calculation of average PSNR differences between RD-curves," *ITU-T VCEG-M33*, 2001. 9, 30, 31

[7] E. Alshina, A. Alshin, J. Min, K. Choi, A. Saxena, and M. Budagavi, "Known tools performance investigation for next generation video coding," ITU-T SG 16, Doc. VCEG-AZ05, Jun. 2015. 9, 42

[8] B. Zeng and J. Fu, "Directional discrete cosine transforms-a new framework for image coding," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 18, no. 3, pp. 305–313, 2008. 9, 37, 43, 44

[9] C.-L. Chang and B. Girod, "Direction-adaptive partitioned block transform for image coding," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.* IEEE, 2008, pp. 145–148. 9, 44, 45

[10] R. A. Cohen, S. Klomp, A. Vetro, and H. Sun, "Direction-adaptive transforms for coding prediction residuals," in *Image Processing (ICIP), 2010 17th IEEE International Conference on.* IEEE, 2010, pp. 185–188. 9, 37, 45, 46

[11] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Rate-distortion optimized transform for intra-frame coding," in *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on.* IEEE, 2010, pp. 1414–1417. 9, 48, 49, 61

[12] M. Wang, K. N. Ngan, and L. Xu, "Efficient h. 264/avc video coding with adaptive transforms," *IEEE TRANSACTIONS ON MULTIMEDIA*, vol. 16, no. 4, p. 933, 2014. 9, 38, 50

[13] C. Lan, J. Xu, and F. Wu, "Enhancement of hevc using signal dependent transform (sdt)," *VCEG-AZ08, Warsaw*, 2015. 9, 38, 51

[14] M. Biswas, M. R. Pickering, and M. R. Frater, "Improved h. 264-based video coding using an adaptive transform," in *Image Processing (ICIP), 2010 17th IEEE International Conference on.* IEEE, 2010, pp. 165–168. 9, 38, 52

[15] V. Sze and M. Budagavi, “High throughput cabac entropy coding in hevc,” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1778–1791, 2012. 9, 25, 60

[16] J. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, “Comparison of the coding efficiency of video coding standards-including high efficiency video coding (hevc),” *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1669–1684, 2012. 11, 20

[17] V. Baroncini, J. Ohm, and G. Sullivan, “Report on preliminary subjective testing of hevc compression capability,” *JCT-VC of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, San José, CA, USA*, vol. 8, no. 7, p. 88, 2012. 11

[18] S. Hollister, “Apple announces native hevc support in macos high sierra and ios 11,” https://www.cnet.com/news/apple-answers-iphone-storage-woes-with-smaller-photos-videos/. 11

[19] C. Kok, “Fast algorithm for computing discrete cosine transform,” *IEEE Transactions on Signal Processing*, pp. 757–760, 1997. 12

[20] A. Dapena and S. Ahalt, “A hybrid DCT-SVD image-coding algorithm,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 2, pp. 114–121, 2002. 13, 52

[21] A. Gersho and R. M. Gray, *Vector quantization and signal compression.* Springer Science & Business Media, 1992. 13

[22] M. Effros, H. F. H. Feng, and K. Zeger, “Suboptimality of the Karhunen-Loeve transform for transform coding,” *IEEE Transactions on Information Theory*, vol. 50, no. 8, pp. 1605–1619, 2004. 13, 33

[23] I. Rec, “H. 265 and iso/iec 23008-2: High efficiency video coding,” *ITU-T and ISO/IEC JTC*, vol. 1. 20

[24] I. Richardson. (2013) An introduction to high efficiency video coding. [Online]. Available: http://www.vcodex.com/h265 20, 22

[25] J.-R. Ohm and G. J. Sullivan, “High efficiency video coding: The next frontier in video compression [Standards in a Nutshell],” *Signal Processing Magazine, IEEE*, vol. 30, no. 1, pp. 152–158, 2013. 21, 23

[26] M. Wien, “High efficiency video coding,” *Coding Tools and specification*, 2015. 24

[27] W. Li and P. Ren, “Efficient cabac bit estimation for h.265/hevc rate-distortion optimization,” *Int. J. Multimed. Data Eng. Manag.*, vol. 6, no. 4, pp. 40–55, Oct. 2015. [Online]. Available: http://dx.doi.org/10.4018/IJMDEM.2015100103 28

[28] O. G. Guleryuz and M. T. Orchard, “Optimized nonorthogonal transforms for image compression,” *Image Processing, IEEE Transactions on*, vol. 6, no. 4, pp. 507–522, 1997. 32, 141

[29] N. Ahmed, T. Natarajan, and K. R. Rao, “Discrete cosine transform,” *Computers, IEEE Transactions on*, vol. 100, no. 1, pp. 90–93, 1974. 33

[30] G. Strang, “The discrete cosine transform,” *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999. 33

[31] H. S. Malvar and D. H. Staelin, “The lot: Transform coding without blocking effects,” *Acoustics, Speech and Signal Processing, IEEE Transactions on*, vol. 37, no. 4, pp. 553–559, 1989. 33

[32] G. J. Sullivan, P. N. Topiwala, and A. Luthra, “The h. 264/avc advanced video coding standard: Overview and introduction to the fidelity range extensions,” in *Optical Science and Technology, the SPIE 49th Annual Meeting.* International Society for Optics and Photonics, 2004, pp. 454–474. 34

[33] A. K. Jain, "A sinusoidal family of unitary transforms," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, no. 4, pp. 356–365, 1979. 34, 48

[34] H. E. Egilmez, A. Said, Y.-H. Chao, and A. Ortega, "Graph-based transforms for inter predicted video coding," in *Image Processing (ICIP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 3992–3996. 35

[35] Y. Ye and M. Karczewicz, "Improved h. 264 intra coding based on bi-directional intra prediction, directional transform, and adaptive coefficient scanning," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on.* IEEE, 2008, pp. 2116–2119. 38, 47, 57, 116

[36] J. Sole, P. Yin, Y. Zheng, and C. Gomila, "Joint sparsity-based optimization of a set of orthonormal 2-d separable block transforms," in *Image Processing (ICIP), 2009 16th IEEE International Conference on.* IEEE, 2009, pp. 9–12. 38, 47, 65, 66, 74, 84

[37] O. G. Sezer, R. Cohen, and A. Vetro, "Robust learning of 2-d separable transforms for next-generation video coding," in *Data Compression Conference (DCC), 2011.* IEEE, 2011, pp. 63–72. 38, 47, 57, 61, 65, 66, 67, 116

[38] X. Zhao, L. Zhang, S. Ma, and W. Gao, "Video coding with rate-distortion optimized transform," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 1, pp. 138–151, 2012. 38, 59, 74

[39] A. Arrufat, P. Philippe, and O. Déforges, "Mode-dependent transform competition for hevc," in *Image Processing (ICIP), 2015 IEEE International Conference on.* IEEE, 2015, pp. 1598–1602. 38, 49, 57, 74, 88, 101, 102, 104, 111, 112, 113, 114, 116, 117, 122, 123

[40] ——, "Rate-distortion optimised transform competition for intra coding in hevc," in *Visual Communications and Image Processing Conference, 2014 IEEE.* IEEE, 2014, pp. 73–76. 38, 49, 74, 100

[41] X. Zhao, J. Chen, A. Said, V. Seregin, H. E. Egilmez, and M. Karczewicz, "Nsst: Non-separable secondary transforms for next generation video coding," in *Proceedings of 32nd Picture Coding Symposium (PCS), Nuremberg, Germany*, 2016. 38

[42] C. Lan, J. Xu, G. Shi, and F. Wu, "Exploiting non-local correlation via signal-dependent transform (sdt)," *IEEE Journal of Selected Topics in Signal Processing*, vol. 5, no. 7, pp. 1298–1308, 2011. 38, 51

[43] C. Lan, J. Xu, W. Zeng, G. Shi, and F. Wu, "Variable block-sized signal dependent transform for video coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. PP, no. 99, pp. 1–1, 2017. 38, 57

[44] "Jvet software repository." [Online]. Available: https://jvet.hhi.fraunhofer.de/svn/svn_HMJEMSoftware/ 38

[45] E. Alshina, A. Alshin, and F. C. Fernandes, "Rotational transform for image and video compression," in *Image Processing (ICIP), 2011 18th IEEE International Conference on.* IEEE, 2011, pp. 3689–3692. 41

[46] K. McCann, W.-J. Han, I.-K. Kim, J.-H. Min, E. Alshina, A. Alshin, T. Lee, J. Chen, V. Seregin, S. Lee *et al.*, "Samsung's response to the call for proposals on video compression technology." 41

[47] R. Cohen, C. Yeo, and R. Joshi, "Ce7: Alternative transforms," *document JCTVC-C507, Guangzhou, China*, 2010. 41

[48] A. Saxena and F. C. Fernandes, "On secondary transforms for intra prediction residual," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, March 2012, pp. 1201–1204. 42

[49] A. Alshin, E. Alshina, M. Budagavi, K. Choi, J. Min, M. Mishourovsky, Y. Piao, and A. Saxena, "Coding efficiency improvements beyond hevc with known tools," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2015, pp. 95 991C–95 991C. 42

[50] M. K. X. Zhao, J. Chen, "Mode-dependent non-separable secondary transform," ITU-T SG 16 (Study Period 2013) Contribution 1044, Doc. COM16-C1044, Sep. 2015. 42

[51] X. Zhao, J. Chen, A. Said, V. Seregin, H. E. Egilmez, and M. Karczewicz, "Nsst: Non-separable secondary transforms for next generation video coding," in *2016 Picture Coding Symposium (PCS)*, Dec 2016, pp. 1–5. 43

[52] S. Zhu, S.-K. A. Yeung, and B. Zeng, "Rd performance upper bound of transform coding for 2-d directional sources," *Signal Processing Letters, IEEE*, vol. 16, no. 10, pp. 861–864, 2009. 43

[53] T. Sikora and B. Makai, "Shape-adaptive dct for generic coding of video," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 5, no. 1, pp. 59–62, 1995. 43

[54] T. Sikora, "Low complexity shape-adaptive dct for coding of arbitrarily shaped image segments," *Signal Processing: Image Communication*, vol. 7, no. 4, pp. 381–395, 1995. 43

[55] P. Kauff and K. Schuur, "Shape-adaptive dct with block-based dc separation and δdc correction," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 8, no. 3, pp. 237–242, 1998. 43

[56] J. MacQueen *et al.*, "Some methods for classification and analysis of multivariate observations," 1967. 46, 59

[57] R. D. Dony and S. Haykin, "Optimally integrated adaptive learning," in *Acoustics, Speech, and Signal Processing, 1993. ICASSP-93., 1993 IEEE International Conference on*, vol. 1. IEEE, 1993, pp. 609–612. 46

[58] ——, "Optimally adaptive transform coding," *Image Processing, IEEE Transactions on*, vol. 4, no. 10, pp. 1358–1370, 1995. 46, 47, 59, 126

[59] M. Effros and P. A. Chou, "Weighted universal transform coding: Universal image compression with the karhunen-loeve transform," in *Image Processing, 1995. Proceedings., International Conference on*, vol. 2. IEEE, 1995, pp. 61–64. 47

[60] K. Sayood, J. D. Gibson, and M. C. Rost, "An algorithm for uniform vector quantizer design," *Information Theory, IEEE Transactions on*, vol. 30, no. 6, pp. 805–814, 1984. 47

[61] C. Archer and T. K. Leen, "Optimal dimension reduction and transform coding with mixture principal components," in *Neural Networks, 1999. IJCNN'99. International Joint Conference on*, vol. 2. IEEE, 1999, pp. 916–920. 47, 59

[62] ——, "Adaptive transform coding as constrained vector quantization," in *Neural Networks for Signal Processing X, 2000. Proceedings of the 2000 IEEE Signal Processing Society Workshop*, vol. 1. IEEE, 2000, pp. 308–317. 47

[63] C. Archer and T. Leen, "A generalized lloyd-type algorithm for adaptive transform coding," *IEEE Trans. Image Processing*, vol. 52, no. 1, 2004. 47

[64] K. Sühring, G. Heising, D. Marpe *et al.*, "Kta (2.6 r1) reference software, 2009." 47

[65] S. Ma, S. Wang, Q. Yu, J. Si, and W. Gao, "Mode dependent coding tools for video coding," *Selected Topics in Signal Processing, IEEE Journal of*, vol. 7, no. 6, pp. 990–1000, 2013. 48, 53, 59, 61

[66] A. Arrufat, P. Philippe, and O. Déforges, "Non-separable mode dependent transforms for intra coding in hevc," in *Visual Communications and Image Processing Conference, 2014 IEEE*. IEEE, 2014, pp. 61–64. 48, 57

[67] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja, "Mode-dependent transforms for coding directional intra prediction residuals," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 4, pp. 545–554, 2012. 48

[68] R. Joshi, P. Chen, M. Karczewicz, A. Tanizawa, J. Yamaguchi, C. Yeo, Y. Tan, H. Yang, and H. Yu, "Ce7: Mode dependent intra residual coding," *document JCTVC-E098, MPEG-H/JCT-VC, Geneva, Switzerland*, 2011. 48

[69] A. Saxena and F. Fernandes, "Mode-dependent dct/dst without 4* 4 full matrix multiplication for intra prediction," *ITU-T & ISO/IEC JCTVC-E125*, 2011. 48

[70] T. Wiegand, W.-J. Han, B. Bross, J.-R. Ohm, and G. J. Sullivan, "Wd3: Working draft 3 of high-efficiency video coding," in *JCT-VC 5th Meeting, JCTVC-E603*, 2011. 48

[71] F. Zou, O. C. Au, C. Pang, and J. Dai, "Enhanced intra mode dependent directional transform," in *APSIPA Annual Summit and Conference 2010*, 2010. 48

[72] V. K. Goyal, J. Zhuang, M. Vetterli, and C. Chan, "Transform coding using adaptive bases and quantization," in *Image Processing, 1996. Proceedings., International Conference on*, vol. 1. IEEE, 1996, pp. 365–368. 51

[73] J.-F. Yang and C.-L. Lu, "Combined techniques of singular value decomposition and vector quantization for image coding," *Image Processing, IEEE Transactions on*, vol. 4, no. 8, pp. 1141–1146, 1995. 53, 92

[74] F. Zou, O. C. Au, C. Pang, and J. Dai, "Rate distortion optimized transform for intra block coding for hevc," in *Visual Communications and Image Processing (VCIP), 2011 IEEE*. IEEE, 2011, pp. 1–4. 57

[75] F. Zou, O. C. Au, C. Pang, J. Dai, X. Zhang, and L. Fang, "Rate-distortion optimized transforms based on the lloyd-type algorithm for intra block coding," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1072–1083, Dec 2013. 57, 66

[76] I. W. Selesnick and O. G. Guleryuz, "A diagonally-oriented dct-like 2d block transform," in *SPIE Optical Engineering+ Applications*. International Society for Optics and Photonics, 2011, pp. 81 381R–81 381R. 58, 73, 94

[77] O. G. Sezer, O. Harmanci, and O. G. Guleryuz, "Sparse orthonormal transforms for image compression," in *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*. IEEE, 2008, pp. 149–152. 61, 62, 67, 68, 72, 73

[78] F. Bossen *et al.*, "Common test conditions and software reference configurations," *Joint Collaborative Team on Video Coding (JCT-VC), JCTVC-F900*, 2011. 77, 80, 105

[79] C. McGoldrick, W. Dowling, and A. Bury, "Image coding using the singular value decomposition and vector quantization," 1995. 92

[80] L. Vása and V. Skala, "Cobra: Compression of the basis for pca represented animations," *Computer Graphics Forum*, vol. 28, no. 6, pp. 1529–1540, 2009. [Online]. Available: http://dx.doi.org/10.1111/j.1467-8659.2008.01304.x 93

[81] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105. 100

[82] X. Yu, Z. Liu, J. Liu, Y. Gao, and D. Wang, "Vlsi friendly fast cu/pu mode decision for hevc intra encoding: Leveraging convolution neural network," in *Image Processing (ICIP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 1285–1289. 101

[83] Z. Liu, X. Yu, S. Chen, and D. Wang, "Cnn oriented fast hevc intra cu mode decision," in *Circuits and Systems (ISCAS), 2016 IEEE International Symposium on*. IEEE, 2016, pp. 2270–2273. 101

[84] Z. Liu, X. Yu, Y. Gao, S. Chen, X. Ji, and D. Wang, "Cu partition mode decision for hevc hardwired intra encoder using convolution neural network," *IEEE Transactions on Image Processing*, vol. 25, no. 11, pp. 5088–5103, 2016. 101

[85] T. Laude and J. Ostermann, "Deep learning-based intra prediction mode decision for hevc," in *Proceedings of 32nd Picture Coding Symposium (PCS), Nuremberg, Germany*, 2016. 103

[86] F. Chollet, "Keras," https://github.com/fchollet/keras, 2015. 104

[87] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014. 104

[88] H. Shao, T. Svoboda, and L. Van Gool, "Zubud-zurich buildings database for image based recognition," 2003. 104, 105, 127

[89] F. Bossen, B. Bross, K. Suhring, and D. Flynn, "Hevc complexity and implementation analysis," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1685–1696, 2012. 105

[90] S. Han, H. Mao, and W. J. Dally, "Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding," *arXiv preprint arXiv:1510.00149*, 2015. 108

[91] A. A. Batalla, "Multiple transforms for video coding," Ph.D. dissertation, INSA de Rennes, 2015. 114

[92] L. Xu, K. N. Ngan, and M. Wang, "Video content dependent directional transform for intra frame coding," in *2012 Picture Coding Symposium*, May 2012, pp. 197–200. 126, 130

[93] Xiph.org video test media. [Online]. Available: https://media.xiph.org/video/derf/ 127

[94] A. Quattoni and A. Torralba, "Recognizing indoor scenes," in *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on.* IEEE, 2009, pp. 413–420. 127

[95] H. E. Egilmez, O. G. Guleryuz, J. Ehmann, and S. Yea, "Row-column transforms: Low-complexity approximation of optimal non-separable transforms," in *2016 IEEE International Conference on Image Processing (ICIP)*, Sept 2016, pp. 2385–2389. 141

Thèse de Doctorat

Saurabh PURI

Apprentissage, sélection et codage de nouvelles transformées de blocs dans et pour la boucle d'optimisation de codeurs vidéo

Learning, selection and coding of new block transforms in and for the optimization loop of video coders

Résumé

Les transformées sont un élément clé dans les systèmes de codage vidéo par blocs. Cette thèse approfondit les schémas d'apprentissage à multiples transformées.
Une première contribution de ce travail est consacrée à l'évaluation de schémas d'apprentissage de transformées de deux types en ligne et hors ligne. Les deux approches sont comparées en détail et leur pertinence respective révélées. Nous proposons ensuite plusieurs techniques afin d'améliorer la stabilité du système d'apprentissage et réduire le coût de la signalisation.
Une seconde contribution concerne les schémas d'apprentissage multi-transformées hors ligne déjà connus qui sont étendus avec pour objectifs de 1) fournir des transformées plus génériques et moins biaisées, 2) obtenir des gains de compression plus élevés, 3) réduire la complexité d'encodage et de décodage. On propose un schéma dit IMDTC (Improved Mode Dependent Transform Competition) qui offre un gain de codage très significatif, plus de 5% par rapport à HEVC standard sous la configuration All Intra (AI), avec une augmentation de complexité raisonnable.
Enfin, l'adaptabilité au contenu de l'apprentissage hors ligne est étendue en explorant une nouvelle approche d'apprentissage des transformées basée sur des jeux de transformées adaptées à des contenus. Plusieurs ensembles contenant de multiples transformées sont appris sur différents contenus et regroupés en jeu. Lors du codage d'une région donnée d'une image, un ensemble de transformées est sélectionné localement à partir du jeu. Les résultats numériques montrent le potentiel élevé de cette approche par rapport aux approches classiques en ligne et hors ligne.

Abstract

Transforms are a key element in block-based video coding systems which, in conjugation with quantization, is important for the overall compression efficiency of the system. This thesis explores multiple-transform-based learning schemes.
A first contribution of this work is dedicated to the evaluation of transform learning schemes with two flavors 1) online learning, and 2) offline learning. The two approaches are compared against each other and their respective appropriability is studied in detail. Some novel techniques are proposed in this work to 1) improve the stability of the learning scheme and 2) to reduce the signaling cost.
In a second contribution of this thesis, the offline multiple-transform learning schemes already known in the literature are further extended with the aims to altogether 1) provide more generic transforms that are less biased towards specific classes of contents, 2) achieve higher compression gains, 3) reduce encoding and decoding computational complexity. An improved Mode Dependent Transform Competition (IMDTC) scheme is proposed which provides a considerable gain of over 5% compared to standard HEVC under All Intra (AI) configuration at a complexity just 2.9 times the standard HEVC.
Finally, the content adaptability aspect of the offline learning is explored through a novel content-adapted pool-based transform learning approach where several multiple-transform sets are learned on different contents and pooled together. During the coding of a given region of an image, one transform set is selected locally from the pool. Numerical results show the high potential of this approach against the conservative online and offline approaches.

Mots clés
Transformations de blocs, transformations dépendantes du signal, transformations basées sur l'apprentissage, optimisation transformée-débit-distortion , HEVC, compression vidéo.

Key Words
block transforms, signal-dependent-transforms, learning-based-transforms, rate-distortion-optimized-transform selection, HEVC, video compression.