Thèse

THESE INSA Rennes
sous le sceau de l'Université européenne de Bretagne
pour obtenir le titre de
DOCTEUR DE L'INSA DE RENNES
Spécialité : Traitement du Signal et des Images

présentée par
Adrià Arrufat

ECOLE DOCTORALE: MATISSE
LABORATOIRE: IETR

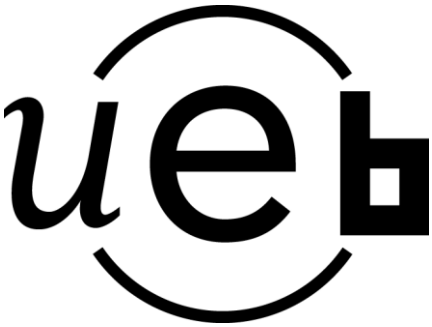
Multiple transforms for video coding

Thèse soutenue le 11.12.2015
devant le jury composé de :

Christine Guillemot
Directrice de Recherche à l'INRIA de Rennes (France) / *présidente*
Béatrice Pesquet-Popescu
Professeur à Telecom ParisTech (France) / *rapporteur*
Mathias Wien
Chercheur à l'Université de Aachen (Allemagne) / *rapporteur*
Fernando Pereira
Professeur à l'Institut des Télécom à Lisbonne (Portugal) / *examinateur*
Philippe Salembier
Professeur à l'Université Polytechnique de Catalogne (Espagne) / *examinateur*
Pierrick Philippe
Ingénieur de recherche à Orange Labs Rennes (France) / Co-encadrant de thèse
Olivier Déforges
Professeur à l'INSA de Rennes (France) / Directeur de thèse

Transformées multiples pour la compression vidéo

Adrià Arrufat



En partenariat avec

Acknowledgements

First of all, I would like to thank my PhD supervisor, Pierrick Philippe from Orange Labs. He has transmitted me the enthusiasm in the daily work by flooding me with new ideas and challenges every day. I feel that I am very lucky and honoured to have been able to work with him and his team and I hope that our paths will cross again during our professional and personal lives.

I also want to thank the rest of the team at Orange Labs: Gordon Clare and Félix Henry for their invaluable help at the beginning, which allowed me to dive into the code and get a better understanding on video coding. A special mention goes to Patrick Boissonade, with whom I shared the office during these years. No matter how complicated seemed a technical difficulty, he always managed to impress me with his knowledge on everything: from coding and optimisation on different architectures, to system administration. I also want to thank him the patience he has shown every time I made the cluster crash with my experiments and how he came up with a way of finding the issue and solving it in record time. Without all these efforts, the results of my work during the last three years would be very different.

Other co-workers at Orange also made my stay a lot more pleasant with the interesting discussions we had at lunch time about almost any topic. These discussions have allowed me to get to know better my team and other people such as Patrick Gioia and Stéphane Pateux.

I also feel very grateful to Didier Gaubil, our team manager, for being always available whenever I needed him. Travelling feels so much safer in the knowledge that someone like him is in charge and will know what to do in case of an emergency.

I have to thank Olivier Déforges for his advice, dedication and support, especially when deadlines for publications approached. Moreover, having attended to two conferences with him, allowed me to get to be more confident and to know him better.

A special mention for Hendrik Vorwerk, whose work during his intern-ship served as an invaluable starting point for my results, and without which I would have struggled to achieve the same results.

The members of the jury (Christine Guillemot, Béatrice Pesquet-Popescu, Mathias Wien, Fernando Pereira and Philippe Salembier) deserve a distinctive mention as well for having accepted to review, assisted to my PhD defence and given constructive feedback.

On a side note, I must mention that all the experiments have been carried out using free and open-source software, as such I thank the Internet and Linux community for making all this kind of knowledge available on-line.

Last but not least, I want to thank my parents, for bearing me on endless phone conversations almost every day and having supported me through all these years.

French summary

Les codeurs vidéo état de l'art utilisent des transformées pour assurer une représentation compacte du signal. L'étape de transformée constitue le domaine dans lequel s'effectue la compression, pourtant peu de variabilité dans les transformées est observé dans la littérature : habituellement, une fois que la taille d'un bloc est sélectionné, la transformée est figée, habituellement de type Transformée en Cosinus Discrète (TCD).

D'autres transformées autres que cette transformée, qui constitue le choix de facto, ont récemment reçu une attention en application de codage vidéo. Par exemple, dans la dernière norme de compression vidéo appelée HEVC (High Efficiency Video Coding, codage vidéo à haute efficacité), la Transformée en Sinus Discrète (TSD) est également utilisée pour traiter les blocs issus de la prédiction pour les tailles 4×4. De plus, pour ces blocs particuliers, HEVC a le choix complémentaire de ne pas transformer le bloc, par utilisation du signal transformSkip. Ce fait révèle l'intérêt croissant pour étendre les choix entre transformées pour accommoder les insatiables besoins en compression vidéo.

Cette thèse se concentre sur l'amélioration des performances en codage vidéo par l'utilisation de multiples transformées. Les résultats sont présentés pour le codage des images Intra, c'est-à-dire des images qui ne sont codées qu'à partir de données locales à celle-ci. Dans cette configuration la norme de compression HEVC (publiée en 2013), qui représente la solution la plus aboutie en la matière, améliore la performance de compression du précédent standard appelé AVC (publié en 2003) de 22%.

HEVC obtient cette amélioration par la démultiplication des alternatives de codage comme l'utilisation de plusieurs tailles de bloc (4, 8, 16, 32 et 64) et modes de prédiction (35 modes) pour générer le signal résiduel (différence entre les pixels de l'image originale et l'image issue de la prédiction) qui est ensuite transformé par une transformée donnée selon la taille sélectionnée. L'objectif pour le codeur est de trouver le meilleur compromis entre la distorsion apportée par la quantification et le débit nécessaire pour transmettre les valeurs approximées. On se rend compte que HEVC investit une part importante dans la génération de résidus, mais peu d'alternatives existent quant à la transformée.

Cette thèse est motivée par le fait que l'utilisation de plusieurs transformées permet d'obtenir une représentation plus parcimonieuse du signal que dans le cas d'une seule transformée. Comme ce thème est relativement peu abordé en codage vidéo, cette thèse tente de combler le vide pour considérer des transformées autres que la transformée en cosinus discrète.

Pour ce faire, un aspect de cette thèse concerne la conception de transformées en utilisant deux techniques qui sont détailléesădans ce manuscrit. L'approche traditionnelle à base de transformées de Karhunen-Loève (KLT) et une transformée optimisée débit distorsion nommée RDOT. La KLT est une transformée qui a pour vocation à minimiser la distorsion sous une hypothèse de haute résolution au travers d'une allocation de bit optimale, cela implique une décorrélation du signal dans le domaine transformé. La RDOT quant à elle, essaie de rendre le signal le plus parcimonieux possible tout en limitant la quantité de distorsion induite par la quantification.

La première approche basée transformée multiples est au travers d'une technique nommée MDDT (Mode Dependent Directional Transform). Celle-ci consiste à utiliser une transformée adaptée, par le biais d'une KLT ou d'une RDOT, pour chaque mode de prédiction intra. Par une utilisation de transformées séparables, un petit gain est observé par rapport à HEVC (de l'ordre de 0.5% du débit est écono-

misé). Néanmoins, l'utilisation de transformées non-séparables révèle des gains tangibles de l'ordre de 2.4% lorsque les transformées sont adaptées au travers de la RDOT. Ce gain est plus favorable que celui observé lorsque les transformées sont construites à partir de l'approche KLT : celle-ci n'améliore HEVC que de 1.8%. Les résultats de cette étude sont résumées dans l'article intitulé "Non-separable mode dependent transforms for intra coding in HEVC" présenté à la conférence VCIP 2014. Ce chapitre conclut que les transformées basées sur la RDOT ont de meilleures performances que celles basée KLT.

Dans l'objectif d'étendre l'approche MDDT, le chapitre suivant décrit une approche nommée MDTC (Mode-Dependent Transform Competition) dans la quelle chaque mode de prédiction est équipé de plusieurs transformées. Lors du codage, ces transformées entrent en compétition de la même façon que les modes de prédiction et tailles de blocs sont sélectionnés. Ce système apporte des gains de l'ordre de 7% pour des transformées non-séparables et 4% pour les transformées séparables, en comparaison avec HEVC. Les résultats de ce chapitre sont publiés dans l'article "Mode-dependent transform competition for HEVC" publié lors de la conférence ICIP 2015. Néanmoins la complexité de tels systèmes est notoire, à la fois en ressources de calcul et en espace de stockage : un facteur de 10 en temps de codage et la complexité de décodage est accrue de 40% par rapport à HEVC. Le stockage des transformées requiert en outre plus de 300 kilo-octets.

En conséquence les chapitres suivants de la thèse développent des approches permettant de simplifier visant à simplifier les systèmes MDTC tout en conservant dans la mesure du possible l'amélioration en débit. Comme les transformées non-séparables apportent les gains les plus prometteurs, le chapitre 5, présente une approche plus simple permettant d'utiliser néanmoins les transformées non-séparables. Ces travaux ont été publiés dans la référence "Image coding with incomplete transform competition for HEVC" présentée à la conférence ICIP 2015. L'approche développée consiste à ne plus utiliser l'ensemble des vecteurs de base lors de la transformation, mais de ne conserver que la première base. Un ensemble de transformées incomplète est ainsi produit et utilisé en complément de la transformée HEVC qui conserve sa base complète. Des gains en compression de l'ordre de 1% sont observés avec cette technique, avec une complexité au décodeur notablement abaissée par rapport aux précédentes approches : elle devient même plus faible que celle de HEVC.

Finalement, une procédure de construction de systèmes MDTC à basse complexité est présentée. Ces travaux sont repris dans la publication "Low complexity transform competition for HEVC" acceptée à la conférence ICASSP 2016. Cette approche à basse complexité s'appuie sur trois composantes qui sont évaluées : tout d'abord une sélection du nombre adéquat de transformées par mode est effectuée, ce qui permet de réduire le nombre de transformées et limiter l'espace de stockage et la complexité de codage. De plus des symétries entre modes de prédiction sont exploitées pour réduire la ROM d'un facteur 3. Pour terminerăl'utilisation de transformées trigonométriques (DTT, Discrete Trigonometric Transforms) est motivé par existence d'algorithmes rapides. L'ensemble de ces contributions réunies permet de proposer un système d'une complexité d'encodage de 50% accrue par rapport à l'état de l'art avec une complexité ajoutée, au niveau décodage et stockage, mineure.

En conclusion les résultats de cette thèse montrent que les transformées multiples apportent des gains significatifs en comparaison avec le plus récent standard de codage vidéo. Des gains très substantiels par rapport à HEVC sont apportés si l'on néglige les aspects complexité. Néanmoins pour des systèmes réalistes des gains tangibles sont obtenus pour des complexités compétitives.

Contents

List of Acronyms

AI All Intra 12
AR Auto-Regressive 21
AVC Advanced Video Coding 7
BD Bjøntegaard Delta 14
CABAC Context Adaptive Binary Arithmetic Coding 11
CB Coding Block 9
CE Core Experiment 36
CTB Coding Tree Block 8
CTC Common Test Conditions 12
DCT Discrete Cosine Transform 10
DPCM Differential Pulse-Code Modulation 9
DST Discrete Sine Transform 10
DTT Discrete Trigonometric Transform 10
GGD Generalised Gaussian Distribution 26
GOP Group of Pictures 13
HDR High Dynamic Range 1
HD High Definition 5
HEVC High Efficiency Video Coding 7
HFR High Frame Rate 1
HVS Human Visual System 6
IEC International Electrotechnical Commision
IPM Intra Prediction Mode 9
IP Internet Protocol 6
ISO International Organisation for Standardisation
ISP Internet Service Provider 6
ITU International Telecommunication Union
JCT Joint Collaborative Team
JCT-VC JCT on Video Coding 34
JPEG Joint Photographic Experts Group
KLT Karhunen-Loève Transform 20
KTA Key Technical Areas 34
MDDT Mode-Dependent Directional Transform 33
MDTC Mode-Dependent Transform Competition 43
MPEG Moving Picture Experts Group

List of Figures

List of Tables

General introduction

Context

Nowadays, video services play a major role in information exchanges around the world. Despite the progress achieved in the last years with video coding standards, improvements are still required as new formats emerge: as High Frame Rate (HFR), High Dynamic Range (HDR) and High Definition (HD) formats become more and more common, new needs for video compression appear that must exploit properties in these domains to achieve higher compression rates.

All these formats are made realistic in terms of service deployment thanks to the fact that around every 10 years, the coding efficiency doubles for equivalent quality. In 2003, the H.264/MPEG-4 AVC standard was defined, providing a compression rate of around 50% with regards MPEG-2 video, defined in 1993. In January 2013, the HEVC standard was released, which outperforms H.264/MPEG-4 AVC by 50% in terms of bitrate savings for equivalent perceptual quality [50].

The work carried out in this thesis started in November 2012, with the HEVC standard almost completely defined. Consequently, the focus has been put on improving HEVC with new techniques that could be adopted in a future standard, tentatively for around 2020. Recently, ITU and ISO, through their respective groups VCEG and MPEG, have started working towards a possible future video coding standard for that time frame.

Being at the beginning of the post-HEVC era, the first steps in this thesis strive to achieve important bitrate savings over HEVC by relaxing complexity constraints.

This thesis is strongly connected to the standardisation context. The first exploratory direction points towards finding new techniques regarding the role of transforms in video coding, such as different transform design methods and the usage of multiple transforms adapted to the nature of video coding signals. Then, the studies move towards making these new techniques involving multiple transforms admissible in a standardisation context, which implies having reasonable impact on standardisation aspects, such as complexity, especially on the decoder side.

Accordingly, this thesis has been organised into the following Chapters:

Chapter 1 starts with an introduction to the basics of video coding and some essential concepts for this thesis on modern video coding standards. The focus is quickly put on the transform stage and its crucial role in the video coding scheme.

Chapter 2 contains a detailed study on the transform role inside video coding applications.

As mentioned, the main motivation of this thesis is to improve video coding by making use of multiple transforms. In order to conceive those transforms, two design methods are studied: the Karhunen-Loève Transform (KLT) and a Rate-Distortion Optimised Transform (RDOT).

The KLT defines a well-known transform design method to conceive transforms that minimise the distortion under the high-resolution quantisation hypothesis and to provide optimal bit-allocation via signal decorrelation in the transform domain.

The RDOT, presented in details in this thesis, describes another design method that tries to output a signal which is as sparse as possible while minimising the distortion introduced by the quantisation.

Chapter 3 compares the KLT and RDOT design methods introduced in the previous Chapter by using multiple transforms in a modified version of HEVC through the Mode-Dependent Directional Transform (MDDT) technique, where one adapted transform (KLT or RDOT) is provided per Intra Prediction Mode.

This experiment questions the optimality of the KLT for video signals. Moreover, the impact of transform separability on video coding in terms of bitrate savings and coding complexity is reconsidered.

Chapter 4 extends the MDDT system by introducing the Mode-Dependent Transform Competition (MDTC) system. The main idea is to provide several transforms in each Intra Prediction Mode. These transforms compete against each other in the same way that block sizes and Intra Prediction Modes do. MDTC systems are able to offer notable bitrate savings at the expense of encoding and decoding complexity and the storage requirements for the used transforms. Therefore, the following Chapters of the thesis contain approaches on simplifying the MDTC system whilst keeping bitrate improvements.

Chapter 5 explores a new way of simplifying non-separable MDTC systems, which provided the most promising results in the previous Chapter, by using incomplete transforms.

Incomplete transforms are low complexity transforms where, instead of using all the basis vectors for the non-separable transforms, only the first one is retained. They are used as companions of the default HEVC transforms.

Chapter 6 presents different methods for reducing the storage requirements of MDTC systems. The proposed methods are based on the fact that not all Intra Prediction Modes present a uniform behaviour inside HEVC in terms of usage frequency and signalling. Therefore, not all Intra Prediction Modes need the same number of transforms, which allows for a reduction of signalling, encoding complexity and storage.

Moreover, symmetries observed in Intra Prediction Modes are exploited to further reduce the storage requirements while having low impact on the bitrate savings.

Chapter 7 replaces the RDOT-based MDTC system with another one based on Discrete Trigonometric Transforms (DTTs).

DTTs can be expressed compactly via an analytical formula, which reduces notably the storage requirements. And, since they are backed up by fast algorithms, they make suitable candidates for low complexity systems that have no impact on the decoding complexity compared to HEVC.

Chapter 8 is a summary of the work carried out in this thesis in order to provide a better overview of the achievements of systems making use of multiple transforms. Since many systems offering various trade-offs between complexity and performance are presented in this thesis, the most relevant are grouped in this Chapter for an easier comparison.

Chapter 9 concludes on the thesis results and presents some perspectives for future work on multiple transforms for video coding.

Thesis contributions

Work carried out in this thesis contains several objectives and contributions, amongst which:

— Reconsider the separability on transforms for video coding.

— Revisit the use of the KLT for video coding.

— Define a metric to be able to rank transforms and allow learning and classification algorithms.

— Design video coding systems that make use of multiple transforms.

The thesis ends with a summary of the thesis objectives, followed by the conclusions and the perspectives for future work based on the use of multiple transforms for video coding.

Chapter 1

Video coding fundamentals

Contents

1.1 Introduction to video coding

The purpose of video coding is to compress video streams, which consist of a sequence of images that, at some point, will be either transmitted or stored. Compressing means reducing the quantity of information so that the amount of bits required to represent that information is low enough to enable the use of video-based applications.

For instance, a video in High Definition (HD) format (1920×1080) at a frame rate of 25 images per second and 8 bits to represent each one of the Red, Green and Blue (RGB) channels, requires:

$$\frac{1920 \times 1080 \text{ pix}}{1 \text{ image}} \times \frac{25 \text{ images}}{1 \text{ s}} \times \frac{3 \text{ channels}}{1 \text{ pix}} \times \frac{8 \text{ bits}}{1 \text{ channel}} \approx 1.20 \text{ Gbit/s}$$

Through this simple example, it is obvious that video coding is compulsory to stream or even store video files: the amount of bitrate reported is this example is beyond the limit of current computing and service architectures and networks.

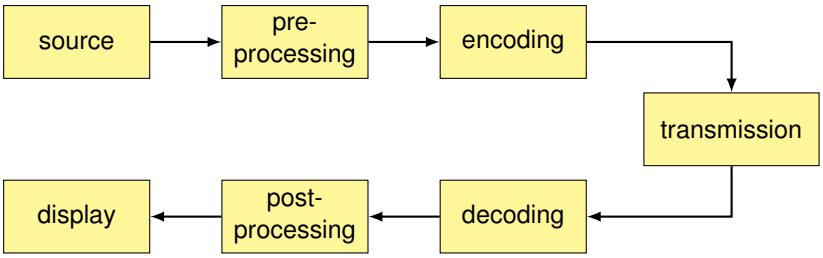


Figure 1.2.1 – Video coding system

Depending on the target quality, it is common to have compression rates ranging from 10 to 1000. For content providers, being able to reduce the size of the content they deliver the opportunity to increase the number of contents they can store, as well as the number of subscribers they can reach using the same resources for storage and network capacity.

In 2013, two thirds of the Internet Protocol (IP) traffic was due to video streaming, and this trend is foreseen to continuously increase, reaching up to 84% of the IP traffic by 2018 [13]. Such forecasts highlight the need to continue the research on new video coding techniques.

As a network operator, being also an Internet Service Provider (ISP) that delivers video services such as live TV streaming or video on demand services, Orange is, therefore, particularly motivated by coding schemes allowing better trade-offs between the quantity and quality of served videos and the amount of reachable clients.

1.2 The video coding system

The video coding system describes a work-flow to process with video sequences. It is composed of several stages: from the video acquisition at the source and to the video display. A good understanding of each part is crucial in order to be able to take the proper decisions when delivering a video coding solution.

This section presents a scheme containing the most important concepts used in state-of-the-art video codecs. Figure 1.2.1 describes a way in which a complete video coding system can be organised as a block diagram. The composing blocks are explained more in details in the following subsections.

1.2.1 Pre-processing

Once the digital video has been captured at the source, which may be of many different sorts such as natural scenes or synthetic computer-generated content, it may need to be processed in order to be encoded. Usually, the pre-processing stage can include some filtering, scaling and colour space conversions on the raw (uncompressed) video sequence.

Up-sampling and down-sampling

The video captured at the source might not have the desired resolution. Consequently, scaling operations should be applied at source to assure the minimum impact to quality.

The sequences used in this thesis are centred around the HD formats (720p and 1080p). However, other resolutions will also be considered, such as WVGA (800×480) and WQXGA (2560×1600).

Colour space conversions

A colour space is an abstract mathematical model specified by primary colours that is used to describe a representation of a picture, usually as a set of tristimulus values.

Colour space conversions are used to change the colour representation of the content to better fit the Human Visual System (HVS) and to decorrelate the components for better coding efficiency.

Often, the colour space at the source is in the RGB format, but since the HVS is more sensible to light variations than to colour variations, the grey scale version of the image, which contains the light information (luma) is separated from the colour information (chroma) [36]. The family colour spaces where the luma is separated from chroma are usually referred to as YUV colour spaces. This way the colour information can be sub-sampled without any visual degradation.

As a result, in this thesis, the improvements on video coding quality will focus the luma component, since it plays a more important role in perceptual quality and represents an important part in the final bitstream.

1.2.2 Encoding

This stage converts the pre-processed raw video sequence into a coded video stream (called bitstream) to ease the storage and transmission. It is, at this point, where the bitrate reduction takes place. The amount of bits required to represent the video stream is reduced by limiting the number of redundancies in the video and by introducing some approximations such as the quantisation step, while limiting the impact of these approximations on the perceived quality.

A deep look at the inners of a widely used coding scheme is provided in §1.3.

1.2.3 Transmission

The transmission stage represents the channel through which the encoded bitstream is made available to the decoder. The channel can be a physical storage, such as optical discs, or any other transmission channel: wired/wireless connections with 1 to 1 (unicast) or 1 to many (multicast/broadcast) transmissions.

Depending on the application and the channel, the behaviour of the encoder may vary: a video that is encoded for storage purposes will not have a real time constraint, present in streaming applications. For example, the Random Access (RA) technique, i.e. the ability to access to a particular piece of a video sequence, needs to be guaranteed for some applications like TV, while the latency needs to be kept as low as possible to enable services like surveillance or video conferencing systems. This latency limitations need to be taken into account at the encoding stage.

1.2.4 Decoding

As the stream is received by the decoder, it is buffered and used to reconstruct the encoded data into the appropriate format, as signalled by the encoder. MPEG and ITU video coding standards define two key points: the bitstream conveying the compressed video data and the bit-exact decoding process, aiming at recovering the sequence of images. MPEG (the JCT 1/SC 29/WG 11 of the ISO/IEC organisation) and VCEG (the question 6 of ITU-T SG 16) are the main organisations specifying video coding algorithms. The most recent video coding specifications include the MPEG-4 part 10 standard / ITU H.264 recommendation, known as Advanced Video Coding (AVC) [22] or MPEG-H Part 2 / ITU H.265, called High Efficiency Video Coding (HEVC) [23]. The encoder must comply with this specification by generating a decodable stream, there is no other normative behaviour defined by a video coding standard.

1.2.5 Post-processing

The post-processing stage performs operations for image enhancement and display adaptation, such as converting back to the original colour space and to the display format.

1.3 The hybrid video coding scheme

State-of-the-art video coding standards such as H.264/MPEG-4 AVC and HEVC use a hybrid video coding scheme. The overall coding structure appeared in H.261, in 1988. Since then, all video coding

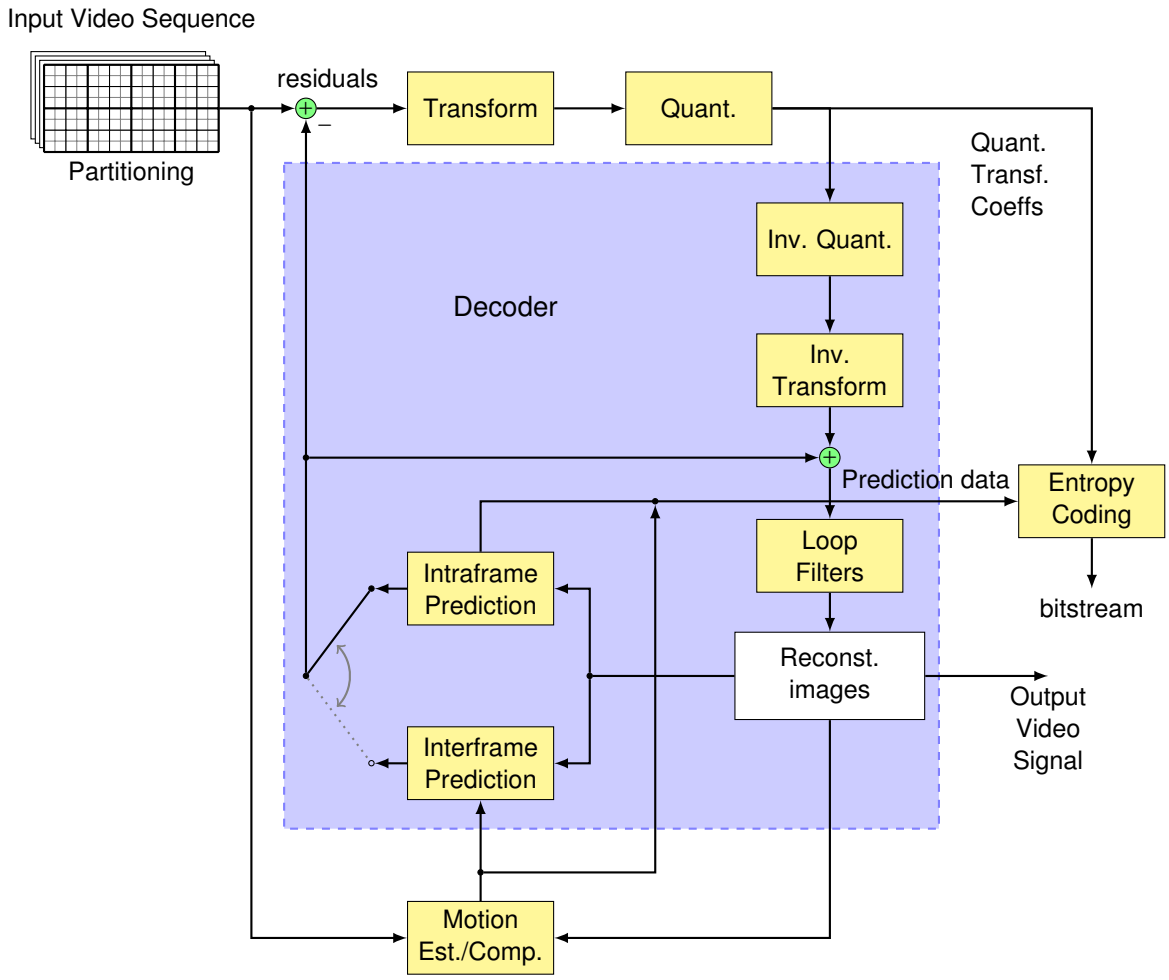


Figure 1.3.1 – Hybrid video coding scheme

standards and recommendations issued by VCEG and the MPEG use this coding structure [56].

The hybrid video coding scheme is named after its use of both temporal prediction and transform coding techniques for the prediction error. A basic structure of the hybrid video coding scheme is presented in figure 1.3.1.

The hybrid video coding scheme provides an efficient way of compressing a video signal into a bitstream of the smallest possible size for a given level of fidelity. The key features to achieve such a small bitstream are the signal prediction and the transformation and quantisation of the prediction error.

A decoder is included in the encoder, represented inside a blue box, to be able to perform its coding decisions based on what the decoder would do while decoding a bitstream.

The building blocks of the hybrid video coding scheme are explained in the following subsections.

1.3.1 Partitioning

In order to process the video frames, they are exhaustively partitioned into non-overlapping blocks. Sub-partitioning a picture allows for a better matching to the spatial distribution of energy. The subdivision blocks ease, as well, the succeeding stages of prediction and transform: the blocks can be processed, under some constraints, independently so that a parallel processing is made possible. The partitioning does not necessarily imply same-sized blocks, allowing rectangular blocks of different sizes to be used, as illustrated in figure 1.3.2 (a). This is due to the quad-tree partitioning that is implemented in HEVC. The figure provides a partitioning for a certain level of quantisation. It can be seen that the picture has been divided into uniform regions.

The optimal choice of the block size is left to the encoder, through Coding Tree Blocks (CTBs)

in HEVC. Each CTB can be split into four Coding Blocks (CBs), which can also be split into four Prediction Units (PUs), which can finally be split into four Transform Units (TUs).

1.3.2 Prediction

Instead of coding the blocks coming from the original source picture directly, the encoder computes an estimation of the block samples, which is then subtracted from the original pixels, generating the residual block. This technique is known as Differential Pulse-Code Modulation (DPCM) in the literature [14].

Those block estimations are carried out by the prediction module, using some information from previously processed blocks. This way, predictable information present in the original blocks is removed and the energy of the resulting signal is lowered so that it requires fewer bits for a given distortion. This is a result of the source coding theory [26]. In other terms, the prediction mechanism aims at removing inter-block redundancies in the video signal.

Predictions must be performed the same way at both encoder and decoder side, and thus computed inside the blue box in figure 1.3.1, referring to the decoder. For this reason, the encoder uses reconstructed blocks (blocks that have already been encoded) as the input data to compute the predictions, as these blocks are equivalent to those the decoder will handle.

Commonly, predictions are of two types, depending on the origin of the prediction source:

— Intra prediction, also called spatial prediction, for those blocks predicted using information within the same picture.

— Inter prediction, also called temporal prediction, for those blocks predicted from pictures other than the one under consideration.

Figure 1.3.2 (b) provides an example of an intra predicted picture using the partitioning from figure 1.3.2 (a). Figure 1.3.2 (c) displays the residual picture (the difference between the original and predicted pictures). This image evidences the parts of the picture that could not be predicted properly and will have to be encoded and transmitted.

Intra prediction

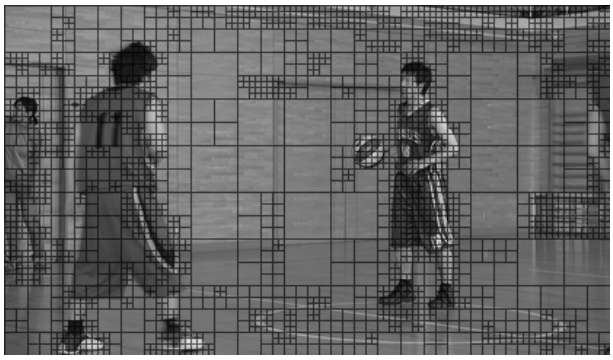
Intra prediction, sometimes referred to as spatial prediction, is used to eliminate spatial redundancies by removing the correlation within local regions of a picture. The basic principle of intra prediction is based on the fact that the texture of a picture region is similar to the texture of its neighbourhood and can be predicted from there. Pictures coded using this technique exclusively are named I-slices.

Different models of predictions can be used through projections of adjacent decoded blocks. These models include directional projections, gradient projections (called Planar) and the projection of the mean value (called DC). The Intra Prediction Modes (IPMs) are used to derive the predictions of the current block from its available boundaries, formed by reconstructed blocks.

Inter prediction

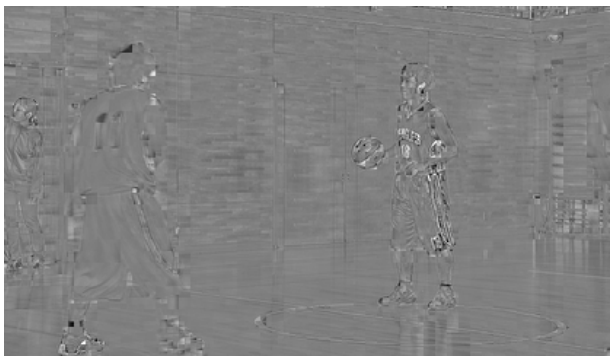
Inter prediction, or temporal prediction, takes advantage of the fact that temporally close pictures share many similarities, and that some of their component regions will move as a whole. Since the encoding order is not necessarily the same as the viewing one, inter predictions can have their origins in either past or future frames, and also combine both origins. This feature facilitates movement tracking across frames. Pictures that are predicted using only one picture either from the past or from the future are called predicted pictures or P-slices. Bi-predicted pictures or B-slices have prediction origins in two different pictures.

At the encoder side, an extra operation, called motion estimation, is carried out. This stage searches the best matching area in the reference picture for the current prediction block. It is one of the most complex parts of video coders in terms of computational requirements. Once a good prediction has been found, a motion vector is created, indicating the offset that has to be applied in the block from the reference picture.



(a) Partitioning

(b) Predicted Picture



(c) Residual Picture

Figure 1.3.2 – Example of a picture at different encoding points for AI main configuration at QP 32

1.3.3 Transform

The transform stage reduces the remaining correlations from the residual block, computed as the difference between the original and the predicted blocks. The goal of the transform is to concentrate the residual signal into as few coefficients as possible in the transform domain. In the spatial domain, the residual signal is spread among the samples of the blocks, while the objective of transform domains is to concentrate as much as possible the residual signal into a few transform domain coefficients exhibiting a large amplitude and the rest might be considered negligible.

This idea of the energy compaction is the main property of the transform stage.

Most of the transforms used in standardised video coding schemes belong to the Discrete Trigonometric Transform (DTT) family. Amongst those, the Discrete Cosine Transform (DCT) of type II has received a considerable amount of attention in the past and is the *de facto* standard transform used in ITU and MPEG codecs since MPEG-1/H.261.

Additional choices were introduced recently, especially in HEVC, where the Discrete Sine Transform (DST) of type VII was adopted.

Provided that the subject of this thesis is centred around the transforms for video coding, the transform stage will be explained thoroughly in Chapter 2.

1.3.4 Quantisation

The quantisation, applied in the transform domain, is used as an approximation operator, reducing the amount of possible output values. In standards like HEVC, the quantisation is scalar: each coefficient is approximated independently from its neighbouring values. In these coding schemes, the quantisation step is controlled by a Quantisation Parameter (QP) that discards any coefficient whose energy level is below a certain threshold. High energy coefficients are also affected by the quantisation. It is worth noticing that the quantisation is the only non-reversible step in the whole hybrid video coding scheme, which induces lossy video coding. Lossless (or near-lossless) video coding can be attained by not using quantisation in the process.

1.3.5 Loop filters

Loop filters have the goal of improving the quality of the reconstructed picture for display purposes. Being located within the loop, these filters have a strong impact on the overall performance of the video coding scheme.

These filters can be classified into two classes, depending on their target region. The fist class is applied to a region of a picture or even a complete picture, whereas filters from the second class operate on a local spatial domain of the picture.

A detailed explanation on how these filters improve the picture quality is explained in §2.4.6 and §9 of [56].

1.3.6 Entropy coding

The last operation consists in reducing the amount of bits transmitted through the use of an entropy code. This is a lossless operation, as such the bit allocation performed during this stage is reversible: no approximation operation is performed at this stage.

Once the transform coefficients have been quantised, they are scanned to make sure they are sorted in a way that will make the entropy coder work efficiently.

The scanning operation is a conversion from a 2D array, containing the quantised transformed coefficients, towards a 1D vector containing the same values sorted in a way that facilitates a compact transmission. An appropriate scanning is crucial for efficient entropy coding [60].

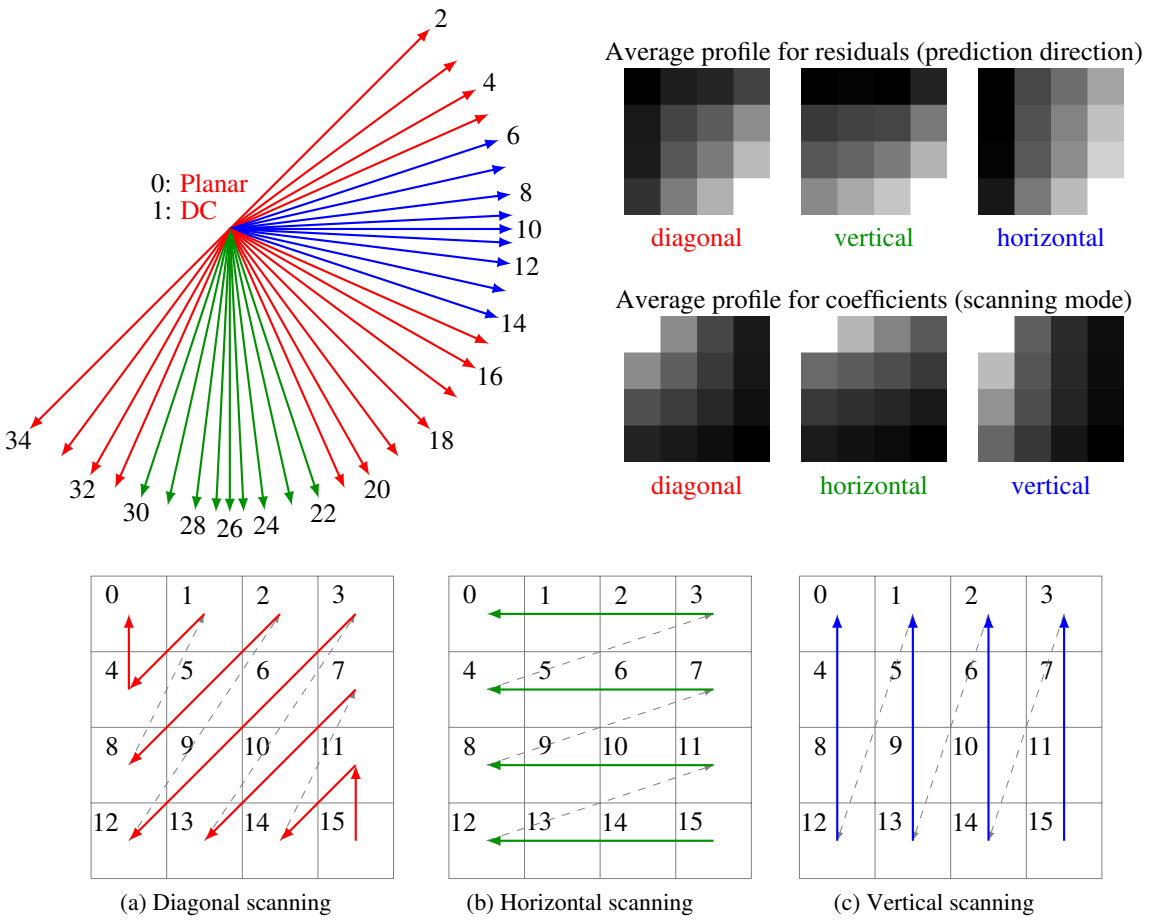
Signalling is also conveyed into the bitstream at this point, and the entropy coder ensures a correct binarisation while using the adequate number of bits. In HEVC, the entropy coding is named Context Adaptive Binary Arithmetic Coding (CABAC).

1.3.7 Intra coding in HEVC

This subsection explains some particularities of the intra coding inside HEVC, as the work carried out in this thesis focuses on the intra coding and take advantage of them.

With regards to previous standards, such as H.264/MPEG-4 AVC, which has 9 different prediction modes for intra coding, HEVC has an improved prediction system, with 35 different prediction modes. The upper-left part of figure 1.3.3 illustrates them. A detailed explanation on how predictions are derived from the block boundaries using those prediction modes can be found in Chapter 6 of [56].

Depending on the selected IPM, residuals present different patterns, and so do their transformed coefficients. The top-right part of figure 1.3.3 presents the average HEVC 4×4 residuals by scanning mode, together with their average representation in transform domain through the DST-VII. The average residual profiles have lower (dark) values near the available borders, which increase with the distance from the boundaries: residuals issued from horizontal and vertical IPMs only have the left and upper borders available, respectively, whereas the remaining IPMs tend to have both borders available. It can



(a) Diagonal scanning (b) Horizontal scanning (c) Vertical scanning

Figure 1.3.3 – HEVC Intra Prediction Modes referred to their scanning patterns together with some average residual examples in the transform domain for 4×4 TUs

also be observed that the scanning patterns match reasonably well the transformed coefficients in each case, sorting them in an increasing order.

These patterns in the transform domain determine different scanning orders (in different colours), as presented in the lower part of figure 1.3.3. An adapted mode to each pattern ensures, in average, a correct order of the coefficients that will group all the null values together. The patterns are described for 4×4 blocks, and the same pattern is used on higher block sizes, which are recursively split into 4 sub-blocks until size 4×4 is reached [48].

1.4 Encoder control

The encoder control is used by essentially all blocks in the diagram from figure 1.3.1. This set of operations allows the encoder to take decisions related to coding based on the application requirements. These decisions include the block sizes and the prediction to use. For each block size and type of prediction, the encoder computes the distortion, using its own decoder, and estimates the rate as illustrated in figure 1.4.1.

The appropriate set of choices, mainly among the prediction modes and block sizes, is performed by comparing the Lagrange values, which balance the distortion and rates obtained for each coding choice. This behaviour is completely described in the standard specification [23].

Depending on the coding configuration defined in the JCT-VC Common Test Conditions (CTC), the encoder may also decide whether to use references from the current picture or from other pictures. The All Intra (AI) configuration encodes each picture independently from the rest, whereas in the RA

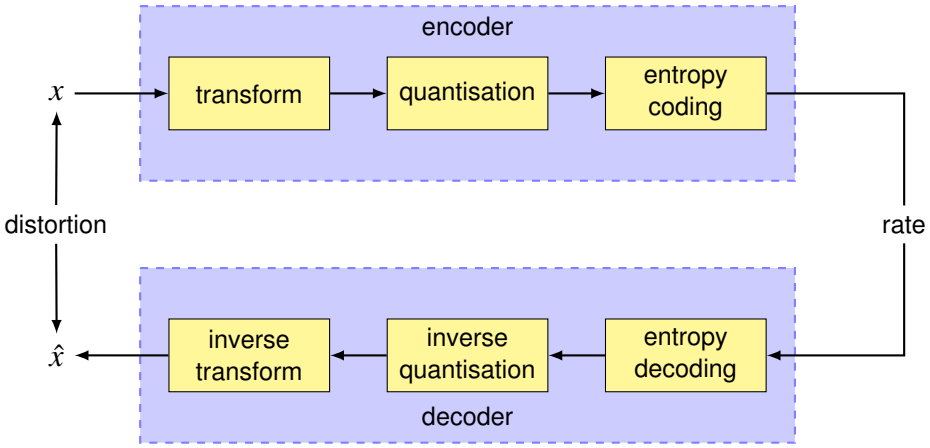


Figure 1.4.1 – Rate-distortion scheme of a transform-based codec

configuration, the pictures conforming the sequences are organised into Groups of Pictures (GOPs), formed by an I-slice, and several P and B-slices.

1.4.1 Distortion measures

This section presents the two distortion measures for objective quality used in this thesis. Although other measures exist, such as structural similarity (SSIM) [53], which is designed to improve on traditional methods such as the MSE and the PSNR, explained in detail below.

The reason for using the traditional methods over other ones is to be able to provide more consistent comparisons with existing techniques present in standardisation, which use the BD measurements based on the PSNR, even if they have can be inconsistent with the HVS.

Mean squared error

The Mean Squared Error (MSE) is the average of the square difference between two signals. For two-dimensional signals, such as images, the MSE can be computed as:

$$MSE_{I,K} = \frac{1}{mn} \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} [I(i,j) - K(i,j)]^2 \tag{1.4.1}$$

Where I and K are two images of $m \times n$ pixels. Usually, I is the reference image and K the coded image.

Peak signal-to-noise ratio

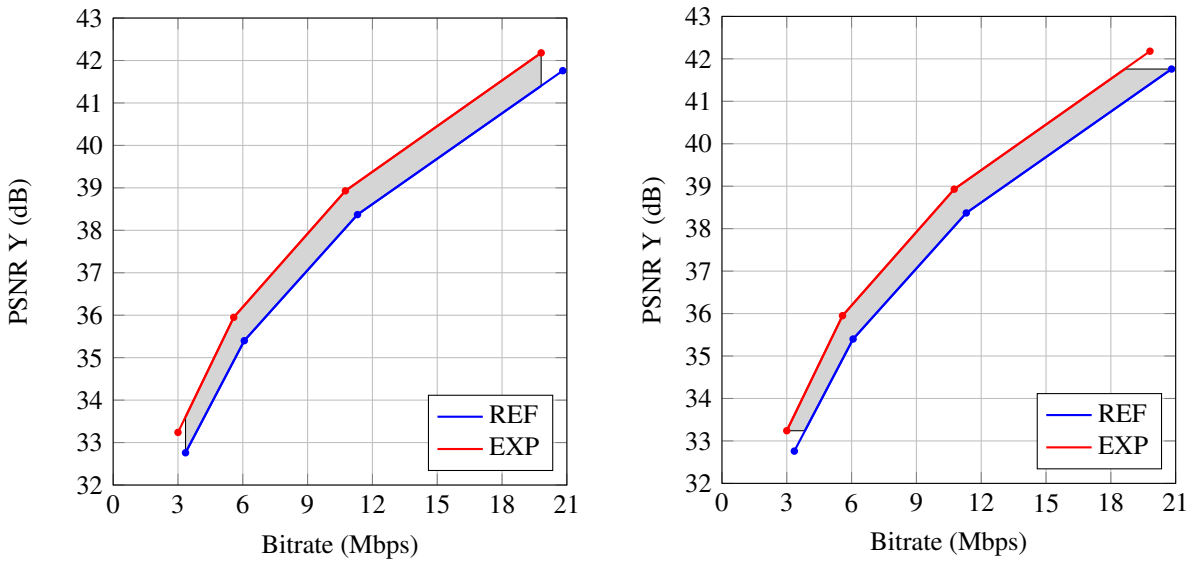
The Peak Signal-to-Noise Ratio (PSNR) is an objective measure of quality that computes the ratio between the maximum possible value of a signal and the energy of the noise that affects the fidelity of its approximation. It is usually defined in a logarithmic scale to cope with the wide range that signals might have. Defining the PSNR in terms of the MSE from (1.4.1), it can be expressed as:

$$PSNR = 10\log_{10}\left(\frac{MAX_I^2}{MSE_{I,K}}\right) = 20\log_{10}(MAX_I) - 10\log_{10}(MSE_{I,K}) \tag{1.4.2}$$

For 8-bit depth images, which are the main format considered in this thesis, the maximum pixel values writes: $MAX = 2^8 - 1 = 255$.

1.4.2 Rate-distortion optimisation

In order to carry out the most sensible decision each time, the encoder can make use of a Rate-Distortion Optimisation (RDO) criterion [51].



(a) BD-PSNR

(b) BD-rate

Figure 1.5.1 – Schematic representation of rate-distortion plots using the BD measurements

Each time the encoder has to make a decision about choosing a particular block size for the partitioning or a prediction mode, it checks the distortion that decision might cause as well as an estimation of the bitrate needed. The encoder performs this computation iteratively on the same block, exploring different coding possibilities and finally selects the one that provides the best score in terms of rate and distortion. This is called the RDO loop. For instance, in HEVC, it allows choosing the best PU size (4,8,16,32,64), TU size (4,8,16,32), IPM (0,1,...,34), prediction source (intra, inter), amongst others.

The trade-off between the distortion and the rate is commonly expressed using a Lagrange measure.

$$J(\lambda) = \text{Distortion} + \lambda \text{Rate} \tag{1.4.3}$$

As seen in the previous subsections, computing the distortion is reasonably straightforward. However, estimating the bitrate is a bit more delicate, since the whole entropy coder cannot be run each time the encoder explores the different possibilities for a block because it has a huge impact on complexity. As a consequence, an estimation of the bitrate is often used in the RDO loop.

1.5 Bjøntegaard Delta measurements

Comparing two video coding techniques objectively might be complicated, as both the distortion and the bitrate savings have to be taken into account jointly.

Metrics introduced by Gisle Bjøntegaard, know as Bjøntegaard Delta (BD) measurements have become the current *de facto* procedure to objectively compare the result of two encodings [8, 9]. Two different metrics are defined and displayed in figure 1.5.1:

(a) BD-PSNR: computes the relative quality improvement in dB.

(b) BD-rate: computes the relative savings in bitrate for an equivalent distortion in percent.

The two encodings in figure 1.5.1, labelled as REF and EXP, represent the reference and the experiment encodings, respectively.

The BD-rate measurement is extensively used in this thesis to appreciate the performance of the proposed systems, as it is the metric used by JCT-VC in standardisation.

1.6 Conclusions

This Chapter has presented the motivation for video compression as well as some general concepts concerning an overview of the video coding system.

Current video coders exploit redundancies existing within images of the video sequence via predictions. These predictions can be either spatial or temporal, depending respectively on whether the prediction source is the same image or another image. Unpredictable parts of the image, called residuals, are then passed through a transformation step in order to concentrate the transformed residual into as few coefficients as possible.

The appropriate block size is selected for the prediction and the transform sizes, named PUs and TUs, respectively. A lot of flexibility is allowed for the prediction and sizes while the set of choices for a given codec is rather limited for the transform.

This thesis will focus on the extension of the choices for the transform stage. Currently, the decision of the prediction source (intra or inter), the block sizes and IPMs are based on a single transform, usually the DCT-II. However, a single transform is unlikely to provide optimal signal compaction for all kinds of possible signals.

This thesis focuses on the extension of the choices for the transform stage, as set of transforms will be provided such that the encoder is able to better adapt the transform to the varying nature of the prediction residuals.

Next Chapter explores the details of the transforms used in video coding, namely their design principles and a comparison between two design approaches.

Chapter 2

Transform coding

Contents

2.1 Introduction to transforms

In the previous Chapter, transforms were identified as an important part in current video coding standards. This Chapter studies the design and properties that make transforms useful in video coding. Although this Chapter is mainly a summary of the literature, a detailed mathematical analysis on the ℓ_0 norm is presented in §2.3.3.

A transform is a mathematical transfer function of a signal from a representation domain to another. The high energy compaction offered by transform process has led this technique to be part of all the international video coding standards.

Transforms allow reducing existing signal correlations in the spatial domain, leading to a more decorrelated signal in the transform domain and ensuring a more compact representation. This is of great importance for the upcoming stages of scanning and entropy coding.

Transforms can be very abstract since they tend to work in N-dimensional spaces, where N represents the number of residual samples processed by the transform. Typical values vary from $N = 4 \times 4$ to $N = 32 \times 32$ using powers of two in modern video codecs, such as HEVC.

However, restraining ourselves to the two-dimensional case, one of the most visual and representative example of transforms are rotations. The example in figure 2.1.1 helps to visualise the transform role in signal coding.

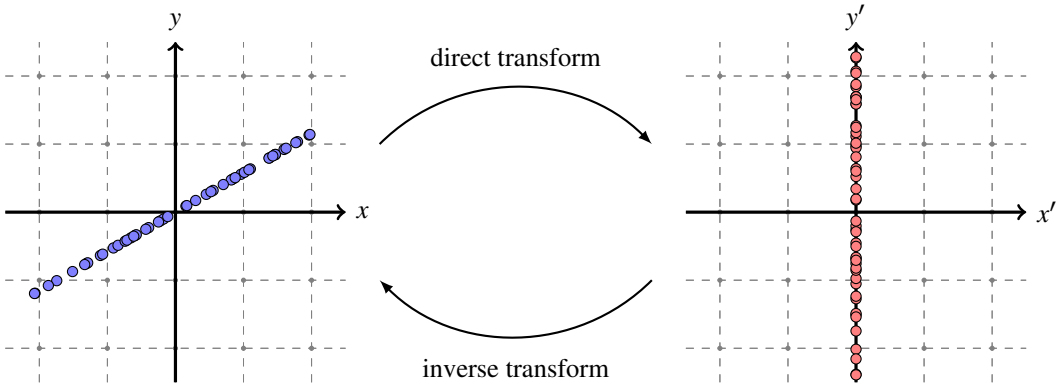


Figure 2.1.1 – Simple transform performing a rotation

Whereas on the left signal (in blue), both coordinates are needed to describe the signal accurately, on the right signal (in red), one coordinate, namely the vertical dimension is enough to provide an equally accurate signal representation, since the horizontal dimension remains constant.

2.1.1 Block transforms

Block-based coding is widely adopted in image/video systems, such as JPEG [28], H.264/MPEG-4 AVC and HEVC.

In these systems, the image to be transformed is split into non-overlapping blocks, and each one is treated and transformed independently [57]. This provides an advantage of being less expensive in terms of computing that other kinds of transforms, such as wavelet transforms used in JPEG-2000 [29].

Since HEVC uses block transforms only, they will be used in this thesis to build up new systems.

2.1.2 Orthogonal transforms

Transforms used in image processing and video coding systems are orthogonal. Orthogonal matrices are square matrices whose rows and columns are orthogonal unit vectors, also known as orthonormal vectors, with:

$$\mathbf{A}^T\mathbf{A} = \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{2.1.1}$$

As a consequence, the inverse matrix of an orthogonal matrix is its transposed version:

$$\mathbf{A}^T = \mathbf{A}^{-1} \tag{2.1.2}$$

This property offers some benefits:

— Fast computation of inverse transform with no need to store it separately.

— Re-use of fast algorithms for both direct and inverse transform applications.

— Energy preservation.

2.1.3 Separability

Image and video coding deal with image blocks, which are two-dimensional signals and, consequently, use transforms able to process those signals.

The straightforward approach to work with those signals is to use non-separable transforms. These transforms take the residual samples from a block previously reshaped into a single-dimensional signal. For instance, a 4×4 block becomes a 16×1 vector, as illustrated in figure 2.1.2. Afterwards the transform is applied normally:

$$\mathbf{X} = \mathbf{A}\,\mathbf{x} \tag{2.1.3}$$

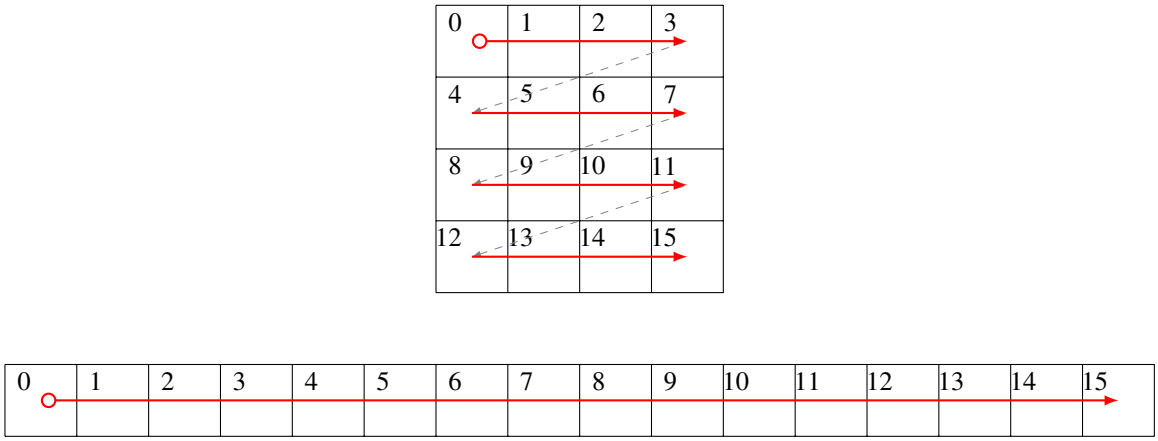


Figure 2.1.2 – Example of a 4×4 block linearisation

Where $\mathbf{x}$ is an $N \times N$ block, reshaped into an $N^2 \times 1$ vector and $\mathbf{A}$ is an $N^2 \times N^2$ matrix. The main disadvantage of this approach is the number of calculations required to obtain the transformed signal: for an $N \times N$ block, the number of operations required to transform it in a non-separable way is: N^4 multiplications and $N^2(N^2-1)$ additions.

On the other hand, any linear correlation can be exploited between any sample within a block: complex relationships between distant samples can be handled regardless their position in the block.

Due to the high amount of operations needed to transform a block using non-separable transforms, separable transforms are widely used in video coding. A block is transformed separately using horizontal and vertical transforms $\mathbf{A}_h, \mathbf{A}_v$ for its rows and columns, respectively, as:

$$\mathbf{X} = \mathbf{A}_v\left(\mathbf{A}_h \mathbf{x}^T\right)^T = \mathbf{A}_v \mathbf{x} \mathbf{A}_h^T \tag{2.1.4}$$

The operation inside the parenthesis transforms the rows of $\mathbf{x}$, and the outer part, the columns of the result. By performing the horizontal and vertical transforms separately, the number of operations required is reduced to $2N^3$ multiplications and $2N^2(N-1)$ additions.

However, this reduction in complexity comes at a price: non-separable transforms are able to exploit any correlation amongst samples within a block, whereas separable transforms can only decorrelate samples that share the same row or column. In other words, separable transforms might be less efficient than their non-separable counterparts. The impact in performance due to separability will be studied in detail in Chapter 3.

If a transform can be expressed by the Kronecker product, denoted as $\otimes$, of two separate transforms for horizontal and vertical directions, it might be considered to be separable. Moreover, a separable 2D transform can be equivalently be represented by a 1D operation as shown in (2.1.5).

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}\mathbf{B} & \cdots & a_{1n}\mathbf{B} \\ \vdots & \ddots & \vdots \\ a_{m1}\mathbf{B} & \cdots & a_{mn}\mathbf{B} \end{bmatrix} \tag{2.1.5}$$

Where $\mathbf{A}$ is an $m \times n$ matrix and $\mathbf{B}$ is a $p \times q$ matrix. More explicitly:

$$\mathbf{A} \otimes \mathbf{B} = \begin{bmatrix} a_{11}b_{11} & a_{11}b_{12} & \cdots & a_{11}b_{1q} & \cdots & \cdots & a_{1n}b_{11} & a_{1n}b_{12} & \cdots & a_{1n}b_{1q} \\ a_{11}b_{21} & a_{11}b_{22} & \cdots & a_{11}b_{2q} & \cdots & \cdots & a_{1n}b_{21} & a_{1n}b_{22} & \cdots & a_{1n}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{11}b_{p1} & a_{11}b_{p2} & \cdots & a_{11}b_{pq} & \cdots & \cdots & a_{1n}b_{p1} & a_{1n}b_{p2} & \cdots & a_{1n}b_{pq} \\ \vdots & \vdots & & \vdots & \ddots & & \vdots & \vdots & & \vdots \\ \vdots & \vdots & & \vdots & & \ddots & \vdots & \vdots & & \vdots \\ a_{m1}b_{11} & a_{m1}b_{12} & \cdots & a_{m1}b_{1q} & \cdots & \cdots & a_{mn}b_{11} & a_{mn}b_{12} & \cdots & a_{mn}b_{1q} \\ a_{m1}b_{21} & a_{m1}b_{22} & \cdots & a_{m1}b_{2q} & \cdots & \cdots & a_{mn}b_{21} & a_{mn}b_{22} & \cdots & a_{mn}b_{2q} \\ \vdots & \vdots & \ddots & \vdots & & & \vdots & \vdots & \ddots & \vdots \\ a_{m1}b_{p1} & a_{m1}b_{p2} & \cdots & a_{m1}b_{pq} & \cdots & \cdots & a_{mn}b_{p1} & a_{mn}b_{p2} & \cdots & a_{mn}b_{pq} \end{bmatrix} \tag{2.1.6}$$

2.1.4 Transform design

Since the objective of the transforms is to be able to represent the signal with as few coefficients as possible while minimising the distortion introduced by the quantisation, transform design methods need to consider trade-offs between the distortion and the number of bits needed to represent those signals in the transform domain, as shown previously in §1.4.2.

Most transform design methods use the MSE from (1.4.1) to evaluate the distortion introduced by a quantiser applied in the transform domain. However, different transform designs exist depending on how the bitrate is estimated and modelled.

Next sections will study different transform design approaches based on different modellings of the rate constraint, namely the KLT (§2.2) and the RDOT (§2.3).

2.2 The Karhunen-Loève transform

The components of source signals within a residual block tend to be correlated amongst them. This correlation is expressed in terms of a correlation matrix containing the linear inter sample correlations. For two N-dimensional signals $\mathbf{x}, \mathbf{y}$, the covariance signal between them can be computed as:

$$\begin{aligned} \mathbf{C}_{\mathbf{x},\mathbf{y}} &= \mathrm{E}\left\{\mathbf{x}\mathbf{y}^T\right\} = \mathrm{E}\left\{\begin{bmatrix} x_0 \\ x_1 \\ \vdots \\ x_{N-1} \end{bmatrix} \begin{bmatrix} y_0 & y_1 & \cdots & y_{N-1} \end{bmatrix}\right\} \\ &= \mathrm{E}\left\{\begin{bmatrix} x_0y_0 & x_0y_1 & \cdots & x_0y_{N-1} \\ x_1y_0 & x_1y_1 & \cdots & x_1y_{N-1} \\ \vdots & \vdots & \ddots & \vdots \\ x_{N-1}y_0 & x_{N-1}y_1 & \cdots & x_{N-1}y_{N-1} \end{bmatrix}\right\} \\ &= \begin{bmatrix} \mathrm{E}\{x_0y_0\} & \mathrm{E}\{x_0y_1\} & \cdots & \mathrm{E}\{x_0y_{N-1}\} \\ \mathrm{E}\{x_1y_0\} & \mathrm{E}\{x_1y_1\} & \cdots & \mathrm{E}\{x_1y_{N-1}\} \\ \vdots & \vdots & \ddots & \vdots \\ \mathrm{E}\{x_{N-1}y_0\} & \mathrm{E}\{x_{N-1}y_1\} & \cdots & \mathrm{E}\{x_{N-1}y_{N-1}\} \end{bmatrix} \end{aligned} \tag{2.2.1}$$

If $\mathbf{x} = \mathbf{y}$, the covariance matrix is the correlation matrix of $\mathbf{x}$.

It is possible to select an orthogonal matrix $\mathbf{A}$ that will make $\mathbf{X} = \mathbf{A}\mathbf{x}$ have pairwise uncorrelated components in the transform domain [17]. The Karhunen-Loève Transform (KLT) is defined as the linear orthogonal transform that reduces the redundancy by a maximum decorrelation of the data, so that the signal can be stored more efficiently [38].

In this section, the KLT is presented under its well-known optimal condition: the high-resolution quantisation assumption [18]. The high-resolution assumption states that the number of quantisation

levels is high and the quantisation step size is small enough to consider the Probability Density Function (PDF) constant for each quantisation interval.

Under this condition, the KLT is the transform that achieves optimal bit allocation for the quantisation of transform coefficients by distributing their variances in such a way as to minimise their geometric mean [26], while minimising the overall distortion [17].

The KLT decorrelates the signal in the transform domain, that is the correlation function of the signal in the transform domain $\mathbf{C_X}$ is a diagonal matrix, which can be computed as follows. Let $\mathbf{x}$ be a zero-mean process and $\mathbf{A}$ an orthogonal transform, then, in the transform domain:

$$\mathbf{X} = \mathbf{A}\mathbf{x} \qquad \text{s.t. } \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{2.2.2}$$

The covariance matrix in the transform domain is expressed as:

$$\mathbf{C_X} = \mathrm{E}\left\{\mathbf{X}\mathbf{X}^T\right\} = \mathbf{A}\mathrm{E}\left\{\mathbf{x}\mathbf{x}^T\right\}\mathbf{A}^T = \mathbf{A}\mathbf{C_x}\mathbf{A}^T \tag{2.2.3}$$

Or equivalently:

$$\mathbf{A}^T\mathbf{C_X} = \mathbf{C_x}\mathbf{A}^T \tag{2.2.4}$$

And since $\mathbf{C_X}$ is diagonal:

$$\mathbf{C_x}\mathbf{a}_i = \lambda_i\mathbf{a}_i \tag{2.2.5}$$

Where:

— $\mathbf{a}_i$'s are the eigenvectors of $\mathbf{C_x}$.

— λ_i's are the eigenvalues of $\mathbf{C_x}$.

As a result, the KLT is a transform whose base vectors are the eigenvectors of the correlation matrix of the input signal.

2.2.1 Particular case on natural images: the DCT

One of the most used transforms in image and video coding is the Discrete Cosine Transform (DCT). In this section, the DCT is justified over a particular kind of signals: natural images. The statistics of pixels in natural images match closely a first order Auto-Regressive (AR) process. A first order AR model, also known as Markov-1 process, is a stochastic process that can be generated through the following regression formula:

$$x(n) = \rho x(n-1) + w(n) \tag{2.2.6}$$

Where ρ is the correlation coefficient between two adjacent samples and $w(n)$ is a white noise with zero mean, whose variance is related to the variance of $x(n)$ σ_x^2 as:

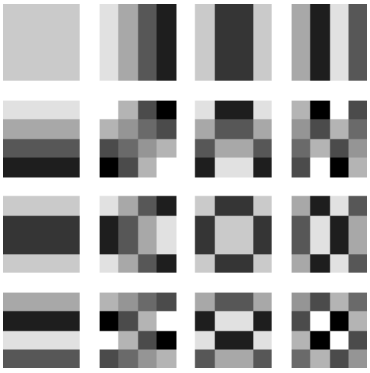
$$\sigma_w^2 = \mathrm{E}\{w(n)w(n)\} = \left(1-\rho^2\right)\sigma_x^2 \tag{2.2.7}$$

The correlation matrix of this process takes the form of a Toeplitz matrix [2]:

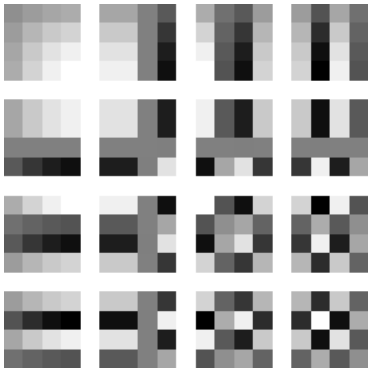
$$\mathbf{R}_x = \sigma_x^2\begin{pmatrix} 1 & \rho & \rho^2 & \cdots & \rho^{N-1} \\ \rho & 1 & \rho & \cdots & \rho^{N-2} \\ \rho^2 & \rho & 1 & \cdots & \rho^{N-3} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ \rho^{N-1} & \rho^{N-2} & \rho^{N-3} & \cdots & 1 \end{pmatrix} \tag{2.2.8}$$

The KLT for this kind of processes, that is, the eigenvectors of $\mathbf{R_x}$, tends to the DCT as $\rho \to 1$ [12]. The DCT-II can be expressed compactly as:

$$\left[C_N^{II}\right]_{n,k} = \sqrt{\frac{2}{N}}\varepsilon_k \cos\left(\frac{\pi(2n+1)k}{2N}\right) \quad n,k = 0,\ldots,N-1 \tag{2.2.9}$$

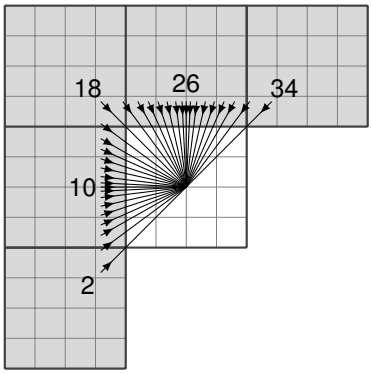


(a) DCT-II 4×4

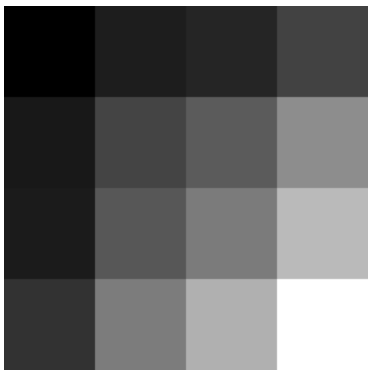


(b) DST-VII 4×4

Figure 2.2.1 – Transforms used in HEVC for 4×4 blocks



(a) The Intra Prediction Modes for the current block (white) from previously reconstructed blocks (grey)



(b) Average intra prediction residual

Figure 2.2.2 – Prediction scheme showing all Intra Prediction Modes and a prediction residual example

$$\varepsilon_k = \begin{cases} \frac{1}{\sqrt{2}} & k = 0 \\ 1 & \text{otherwise} \end{cases} \tag{2.2.10}$$

The fact that the DCT approximates the KLT for image signals and its efficient implementation has made it the preferred choice in image and video coding algorithms to decorrelate the signals and provide optimal bit allocation [48].

2.2.2 Particular case on prediction residuals: the DST

Although the DCT has been proved to be nearly the KLT for natural images, it is used in predictive transform coding based video standards. This kind of coding leads to signals whose nature differs from that of natural images: prediction residuals. More precisely, prediction residuals resulting from intra prediction tend to have particular properties. Those residuals are computed using predictions from already reconstructed blocks, usually available on the left and upper borders (in grey) of the current residual (in white), as shown in figure 2.2.2.a. An example of an average intra prediction residual is provided in figure 2.2.2.b. It can be noticed that the energy of the residual is lower (dark) near the borders where reconstructed blocks are available, and that the error gets higher (lighter) as one moves away from the boundaries. These properties motivated a study on this particular kind of blocks, resulting in a transform that performs better on them than the DCT: the Discrete Sine Transform (DST) [20].

Intra prediction residuals present a correlation matrix that can be modelled using a Toeplitz tridiagonal matrix, as the one in (2.2.11) [20, 64]. The eigenvalues of that correlation matrix can be expressed in a closed form using the DST-VII from (2.2.12).

$$\tilde{\mathbf{R}}_x = \begin{pmatrix} 1+\rho^2 & -\rho & 0 & 0 & \cdots & 0 \\ -\rho & 1+\rho^2 & -\rho & 0 & \cdots & 0 \\ 0 & -\rho & 1+\rho^2 & -\rho & \cdots & 0 \\ 0 & \ddots & \ddots & \ddots & \ddots & \vdots \\ \vdots & \ddots & \ddots & \ddots & \ddots & -\rho \\ 0 & \cdots & 0 & 0 & -\rho & 1+\rho^2-\rho \end{pmatrix} \tag{2.2.11}$$

$$\left[S_N^{VII}\right]_{n,k} = \frac{2}{\sqrt{2N+1}} \sin\left(\frac{\pi(2n-1)k}{2N+1}\right), \quad n,k = 1,\ldots,N \tag{2.2.12}$$

Figure 2.2.1.b presents the DST-VII bases for 4×4 blocks. The similarity between an average residual block from figure 2.2.2.b can be spotted by comparing it to the first base vector from the DST.

The use of the DST in HEVC over the DCT on 4×4 blocks leads to a bitrate reduction of 1% on average [50] for intra blocks. An explanation on the adoption of the DST into HEVC is detailed in Chapter 3, more precisely in §3.1.2.

Note on the KLT

The KLT is often presented as the optimal transform, sometimes even for all possible sources of signals. However, it has been proved to be suboptimal in the transform coding / bit allocation sense in some cases [15]. For this reason, next section studies another kind of transform design that leads to optimal transforms under different conditions.

2.3 The rate-distortion optimised transform

As discussed before, transforms try to find a compact representation of the signal in the transform domain. The KLT, under certain assumptions, provides optimal bit allocation for transformed coefficients and signal decorrelation.

Sezer proposes an alternative kind of transform to optimise the trade-off between the distortion and the rate from (1.4.3) by taking into account the sparsity of the output signal in their design [46, 47]. This transform, named Rate-Distortion Optimised Transform (RDOT), unlike the KLT, does not make use of the high-resolution assumption. Instead, the bitrate constraint is modelled via the sparsity of the transformed coefficients, by the introduction of the ℓ_0 norm, which counts the number of non-zero coefficients in a vector. The proposed RDOT can be expressed as:

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \sum_{\forall i} \min_{\mathbf{c}_i} \left(\|\mathbf{x}_i - \mathbf{A}^T\mathbf{c}_i\|_2^2 + \lambda \|\mathbf{c}_i\|_0 \right) \tag{2.3.1}$$

Where $\mathbf{x}_i$ are the input signals, i.e. a block of the training set and $\mathbf{c}_i$ are its quantised transformed coefficients using the transform $\mathbf{A}$. $\mathbf{A}^T$ is its transposed matrix, as $\mathbf{A}$ is chosen orthogonal. The constraint in the cost function is the ℓ_0 norm of the coefficients, i.e. the number of non-zero coefficients. Finally, λ is the Lagrange multiplier that tunes the constraint.

The fact that this transform design strives to obtain sparse signals in the transform domain seems to be adapted to the state-of-the-art video coding standard, HEVC. There are numerous syntax elements in HEVC that deal with sparsity: for a given transformed residual, the position of the last non-zero value is signalled, meaning that, instead of explicitly transmitting all the following zeroes, it is indicated that from that point onwards, all the values are zero. More generally, flags indicating whether the individual coefficients are significant or not are also part of the standard syntax. Therefore, increasing the number of zeros of the quantised coefficients in the transform domain seems a good objective.

A thorough study of (2.3.1) analysing its properties and consequences is detailed below.

2.3.1 The RDOT metric

The value that RDOT minimises is expressed in (2.3.2) for a single signal. This metric depends exclusively on the quantisation step Δ and the used transform $\mathbf{A}$. It is shown in §2.3.3 that the Lagrange multiplier λ is tied to the quantisation step Δ.

$$J(\lambda) = \|\mathbf{x} - \mathbf{A}^T\mathbf{c}\|_2^2 + \lambda\|\mathbf{c}\|_0 \tag{2.3.2}$$

The first part of the equation represents the distortion introduced by the quantisation. The second term serves as a rate constraint, by ensuring that the number of significant values in the transform domain is minimised together with the distortion.

The minimisation of the metric is done in two steps, carried out iteratively until convergence:

1. Finding the optimal coefficients for a given transform.
2. Updating the transform for the optimal coefficients.

These two steps are detailed in the subsections below.

Optimal coefficients for a given transform

The optimal coefficients that introduce the minimum distortion for a given quantisation step Δ are obtained by transforming the signal and hard-thresholing them:

$$\mathbf{c} = \lfloor\mathbf{X}\rceil = \lfloor\mathbf{A}\mathbf{x}\rceil \tag{2.3.3}$$

The threshold value is tightly related to the Lagrange multiplier λ, as demonstrated in §2.3.3:

$$\mathbf{c}[n] = \begin{cases} \mathbf{X}[n] & |\mathbf{X}[n]| \geq \dfrac{\Delta}{2} \\ 0 & \text{otherwise} \end{cases} \tag{2.3.4}$$

Optimal transform for given coefficients

Once the optimal coefficients have been found, the transform $\mathbf{A}$, chosen orthogonal, is updated to provide the mapping between $\mathbf{x}$ and $\mathbf{c}$ while minimising the reconstruction error.

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(\sum_{\forall i} \|\mathbf{x}_i - \mathbf{A}^T\mathbf{c}_i\|^2 \right) \qquad \text{s.t. } \mathbf{A}\mathbf{A}^T = \mathbf{I} \tag{2.3.5}$$

Since the expression is a scalar value, it can be rewritten using the trace (the sum of a matrix diagonal):

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(\sum_{\forall i} \mathrm{tr}\left(\left(\mathbf{x}_i - \mathbf{A}^T\mathbf{c}_i\right)^T \left(\mathbf{x}_i - \mathbf{A}^T\mathbf{c}_i\right) \right) \right) \tag{2.3.6}$$

Operating:

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(\sum_{\forall i} \mathrm{tr}\left(\mathbf{x}_i^T\mathbf{x}_i - \mathbf{x}_i^T\mathbf{A}^T\mathbf{c}_i - \mathbf{c}_i^T\mathbf{A}\mathbf{x}_i + \mathbf{c}_i^T\mathbf{A}\mathbf{A}^T\mathbf{c}_i \right) \right) \tag{2.3.7}$$

Since the trace is a linear operator and $\mathbf{A}\mathbf{A}^T = \mathbf{I}$:

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(\sum_{\forall i} \mathrm{tr}\left(\mathbf{x}_i^T\mathbf{x}_i\right) - \mathrm{tr}\left(\mathbf{x}_i^T\mathbf{A}^T\mathbf{c}_i\right) - \mathrm{tr}\left(\mathbf{c}_i^T\mathbf{A}\mathbf{x}_i\right) + \mathrm{tr}\left(\mathbf{c}_i^T\mathbf{c}_i\right) \right) \tag{2.3.8}$$

Making use of the cyclic property of the trace and removing $\mathbf{A}$-independent terms:

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(\sum_{\forall i} -2\mathrm{tr}\left(\mathbf{c}_i\mathbf{x}_i^T\mathbf{A}^T\right) \right) \tag{2.3.9}$$

Defining $\mathbf{Y} = \sum_{\forall i} \mathbf{c}_i \mathbf{x}_i^T$ and its Singular Value Decomposition (SVD) $\mathbf{Y} = \mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{V}^T$, where $\mathbf{U}$ and $\mathbf{V}$ are orthogonal and $\mathbf{\Lambda}$ is a positive semi-definite diagonal matrix. The equation rewrites as follows:

$$\mathbf{A}_{opt} = \arg\min_{\mathbf{A}} \left(-2\mathrm{tr}\left(\mathbf{U}\mathbf{\Lambda}^{1/2}\mathbf{V}^T\mathbf{A}^T \right) \right) \tag{2.3.10}$$

Minimising a negative expression is equivalent to maximising its positive version, and re-arranging the terms using the trace cyclic property:

$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \left(2\mathrm{tr}\left(\mathbf{\Lambda}^{1/2}\mathbf{V}^T\mathbf{A}^T\mathbf{U} \right) \right) \tag{2.3.11}$$

Let $\mathbf{P} = \mathbf{V}^T\mathbf{A}^T\mathbf{U}$. Since $\mathbf{V}$, $\mathbf{A}$ and $\mathbf{U}$ are orthogonal, so is $\mathbf{P}$. The equation is now:

$$\mathbf{A}_{opt} = \arg\max_{\mathbf{A}} \left(\mathrm{tr}\left(\mathbf{\Lambda}^{1/2}\mathbf{P} \right) \right) \tag{2.3.12}$$

Since $\mathbf{\Lambda}$ is a diagonal matrix whose entries are non-negative by definition and $\mathbf{P}$ is orthogonal, the maximisation is achieved when $\mathbf{P} = \mathbf{I}$:

$$\mathbf{V}^T\mathbf{A}_{opt}^T\mathbf{U} = \mathbf{I} \quad \Rightarrow \quad \mathbf{A}_{opt} = \mathbf{U}\mathbf{V}^T \tag{2.3.13}$$

Summing up, the optimal transform is obtained using the SVD of the covariance matrix between the output signal (the transformed and quantised coefficients) and the input signal (the prediction residuals).

2.3.2 Separable RDOT design

The methods for transform design and learning presented in the previous sections are non-separable. This means that the input block is linearised and the transform is applied in one single step. Due to computational complexity issues, non-separable transforms are hardly used in performing solutions. A lower complexity approach involves separable transforms, which may not be able to concentrate the energy in the transform domain as well as their non-separable counterparts, as explained in §2.1.3. In order to design and learn separable transforms, the design and learning methods have to be adapted. In the case of the KLT, it is straightforward to see that one can learn a horizontal KLT to transform the rows of the signal and a vertical KLT for the columns. However, the RDOT algorithm needs further tuning in order to obtain separable transforms. A possible way of learning a separable RDOT was also proposed by Sezer [46] and validated by independent researches [49]. The proposed method consists in updating each one of the horizontal and vertical transforms separately, also chosen to be orthogonal transforms.

The separable transformation of a block $\mathbf{x}$ has been previously defined as a two step transformation in (2.1.4):

$$\mathbf{X} = \mathbf{A}_v \left(\mathbf{A}_h\mathbf{x}^T \right)^T = \mathbf{A}_v\mathbf{x}\mathbf{A}_h^T$$

Where $\mathbf{x}$ is the two-dimensional block to transform, $\mathbf{A}_h$ is the horizontal transform, used to transform the rows of $\mathbf{x}$ and A_v is the vertical transform, used to transform the resulting columns.

The equation to be optimised using separable transforms reads as follows:

$$\mathbf{A}_{v_{opt}}, \mathbf{A}_{h_{opt}} = \arg\min_{\mathbf{A}_v, \mathbf{A}_h} \left(\sum_{\forall i} \min_{\mathbf{c}_i} \|\mathbf{x}_i - \mathbf{A}_v^T\mathbf{c}_i\mathbf{A}_h\|_2^2 + \|\mathbf{c}_i\|_0 \right) \tag{2.3.14}$$

This minimisation problem can be solved in a similar way to the non-separable version.

Optimal coefficients for a given transform

As in the non-separable problem, the optimal coefficients $\mathbf{c}_i$ are found by hard-thresholding the components of $\mathbf{X}_i = \mathbf{A}_v\mathbf{x}_i\mathbf{A}_h^T$ with $\Delta/2$.

Optimal vertical transform for given coefficients

The vertical transform needs to be updated accordingly:

$$\mathbf{A}_{v_{opt}} = \arg\min_{\mathbf{A}_v} \left(\sum_{\forall i} \|\mathbf{x}_i - \mathbf{A}_v^T \mathbf{c}_i \mathbf{A}_h\|_2^2 \right) \qquad \text{s.t. } \mathbf{A}_v \mathbf{A}_v^T = \mathbf{I} \tag{2.3.15}$$

Expanding the expression and grouping the terms as in the non-separable the covariance matrix $\mathbf{Y}$ can be defined as:

$$\mathbf{Y} = \sum_{\forall i} \mathbf{c}_i \mathbf{A}_h \mathbf{x}_i = \mathbf{U} \mathbf{\Lambda}^{1/2} \mathbf{V}^T \tag{2.3.16}$$

Then the optimal transform is given by:

$$\mathbf{A}_{v_{opt}} = \mathbf{U} \mathbf{V}^T \tag{2.3.17}$$

Optimal coefficients for updated vertical transform

As in the non-separable problem, the optimal coefficients $\mathbf{c}_i$ are determined by hard-thresholding the components of $\mathbf{X} = \mathbf{A}_{v_{opt}} \mathbf{x}_i \mathbf{A}_h^T$ with $\Delta/2$. However, this time the optimal vertical transform from the previous step is used.

Optimal horizontal transform for given coefficients

Similarly, the horizontal transform needs to be updated accordingly:

$$\mathbf{A}_{h_{opt}} = \arg\min_{\mathbf{A}_h} \left(\sum_{\forall i} \|\mathbf{x}_i - \mathbf{A}_{v_{opt}}^T \mathbf{c}_i \mathbf{A}_h\|_2^2 \right) \qquad \text{s.t. } \mathbf{A}_h \mathbf{A}_h^T = \mathbf{I} \tag{2.3.18}$$

Expanding the expression and grouping the terms as in the non-separable the covariance matrix $\mathbf{Y}$ can be defined as:

$$\mathbf{Y} = \sum_{\forall i} \mathbf{c}_i^T \mathbf{A}_{v_{opt}} \mathbf{x}_i = \mathbf{U} \mathbf{\Lambda}^{1/2} \mathbf{V}^T \tag{2.3.19}$$

The optimal transform is given by:

$$\mathbf{A}_{h_{opt}} = \mathbf{U} \mathbf{V}^T \tag{2.3.20}$$

Hence, the optimal transform are obtained in two iteratively alternated steps until convergence.

2.3.3 The Lagrange multiplier and the zero norm

Video coding residuals distribution can be modelled using a Generalised Gaussian Distribution (GGD), also known as generalised normal distribution or exponential power distribution [30, 63]. For this reason, in order to obtain the optimal Lagrange multiplier in a reasonably general case, a GGD is used to represent the residuals PDF.

Figure 2.3.1 illustrates the PDFs of a set of residuals transformed with the DCT and the same residuals transformed with an adapted RDOT. As expected, the RDOT, represented by the red curve, presents higher sparsity than the generic DCT, in black: its value is above the DCT on zero, meaning that the coefficients have more zeros when using the RDOT. Using different transforms, modifies the resulting PDF, but since the used transforms are orthogonal, the variance remains unaltered. Additionally, since residuals are computed as the difference between predicted and original blocks, they are prediction errors. These errors deviate evenly above and below from the actual value, evidenced by their zero-mean PDF, which is symmetrical and centred around zero. Figure 2.3.1 also includes the PDF of a Laplace distribution and a normal distribution. Those two distributions are particular cases of a GGD.

The centred GGD PDF can be expressed compactly as:

$$\mathrm{GGD}(\sigma, \gamma, x) = a \mathrm{e}^{-(b|x|)^\gamma} \tag{2.3.21}$$

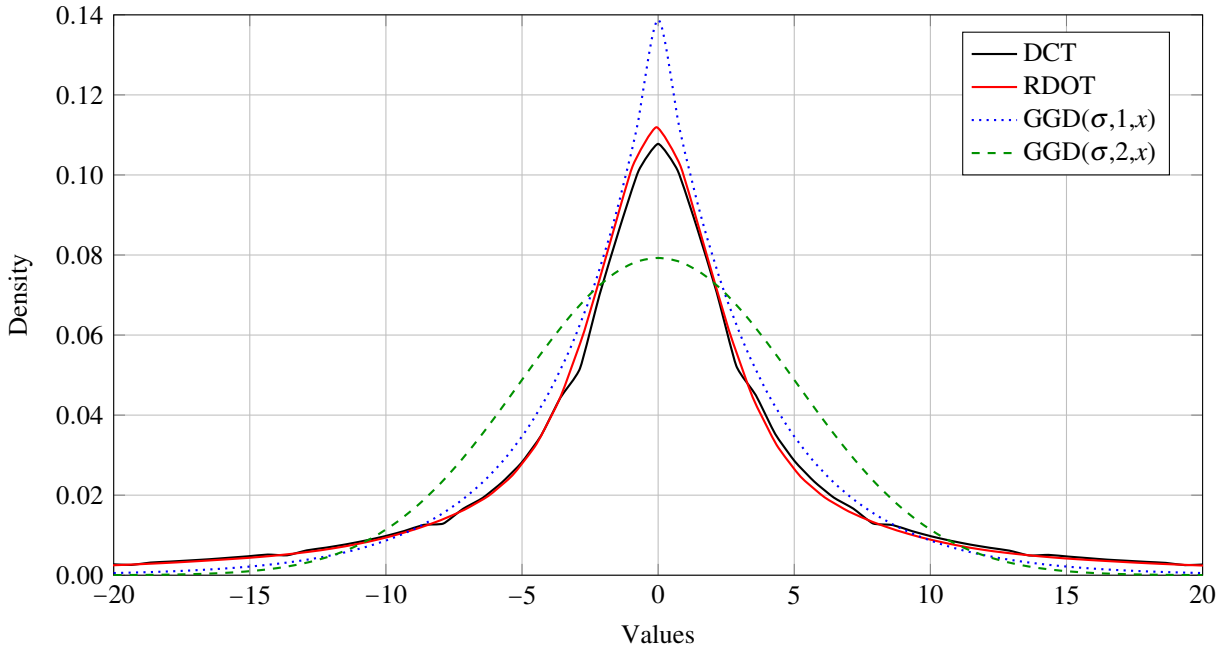


Figure 2.3.1 – PDFs of the residuals with different transforms compared to Laplace and normal distributions

Where:

$$b = \frac{1}{\sigma}\sqrt{\frac{\Gamma(3/\gamma)}{\Gamma(1/\gamma)}} \tag{2.3.22}$$

$$a = \frac{b\gamma}{2\Gamma(1/\gamma)} \tag{2.3.23}$$

and $\Gamma(z)$ is the gamma function, defined as:

$$\Gamma(z) = \int_0^\infty t^{z-1}\mathrm{e}^{-t}\mathrm{d}t \tag{2.3.24}$$

The Laplace and normal or Gaussian PDFs are achieved with $\gamma = 1$ and $\gamma = 2$, respectively. The uniform distribution can even be reached by making $\gamma \to \infty$. However, from figure 2.3.1 one could approximate the video coding residuals PDF with $1 \leq \gamma \leq 2$ [32].

In Sezer's work [46, 47], the optimal value of the Lagrange multiplier λ has been claimed to be straightforward to obtain. Nevertheless, no analytical way of proving its optimality has been found in the literature. Hence, a detailed study has been carried out below.

In order to find the optimal λ from (2.3.2), which describes the trade-off between the distortion and the rate, $J(\lambda)$ has to be derived. The problem will be tackled in two separate steps:

1. Compute the distortion analytically and derive it.
2. Compute the rate constraint and derive it.

It has been decided to normalise the (2.3.2) by N (the signal dimension) to simplify the equations. This scaling factor does not affect the solution.

Derivation of the distortion function

The distortion introduced by the hard-thresholding from (2.3.4) can be expressed as:

$$D = \frac{1}{N}\int_{-\infty}^{\infty} N(x-\hat{x})^2 \mathrm{P}_X(x)\mathrm{d}x = \int_{-\Delta/2}^{\Delta/2} x^2 \mathrm{P}_X(x)\mathrm{d}x \tag{2.3.25}$$

The integration intervals have been reduced to where the quantised values differ from the original ones, that is, the values that have been affected by the hard-thresholding from (2.3.4).

Substituting $P_X(x)$ with the residuals PDF:

$$D = \int_{-\Delta/2}^{\Delta/2} x^2 a \mathrm{e}^{-(b|x|)^\gamma} \mathrm{d}x \tag{2.3.26}$$

$$= 2\int_{0}^{\Delta/2} x^2 a \mathrm{e}^{-(bx)^\gamma} \mathrm{d}x \tag{2.3.27}$$

$$= 2a\frac{\Gamma\left(\frac{3}{\gamma}\right) - \Gamma\left(\frac{3}{\gamma}, \left(\frac{b\Delta}{2}\right)^\gamma\right)}{b^3\gamma} \tag{2.3.28}$$

Where $\Gamma(a,z)$ is the incomplete upper gamma function, defined as:

$$\Gamma(a,z) = \int_z^\infty t^{a-1}\mathrm{e}^{-t}\mathrm{d}t \tag{2.3.29}$$

Deriving (2.3.28) in Δ:

$$\frac{\mathrm{d}D}{\mathrm{d}\Delta} = \frac{\Delta^2 a \mathrm{e}^{(-b\Delta/2)^\gamma}}{4} \tag{2.3.30}$$

Derivation of the zero norm function

By definition, the ℓ_0 norm is the total number of non-zero elements in a vector. Consequently, the constraint can be expressed as:

$$R = \frac{1}{N} N \mathrm{P}_X\left(|X| \geq \frac{\Delta}{2}\right) = 1 - P_X\left(|X| < \frac{\Delta}{2}\right) = 1 - \int_{-\Delta/2}^{\Delta/2} \mathrm{P}_X(x)\mathrm{d}x \tag{2.3.31}$$

Substituting $P_X(x)$ with the residuals PDF:

$$R = 1 - \int_{-\Delta/2}^{\Delta/2} a \mathrm{e}^{-(b|x|)^\gamma} \mathrm{d}x \tag{2.3.32}$$

$$= 1 - 2\int_{0}^{\Delta/2} a \mathrm{e}^{-(bx)^\gamma} \mathrm{d}x \tag{2.3.33}$$

$$= 1 + 2a\frac{\Gamma\left(\frac{1}{\gamma}, \left(\frac{b\Delta}{2}\right)^\gamma\right) - \Gamma\left(\frac{1}{\gamma}\right)}{b\gamma} \tag{2.3.34}$$

Deriving (2.3.34) in Δ:

$$\frac{\mathrm{d}R}{\mathrm{d}\Delta} = -a\mathrm{e}^{(-b\Delta/2)^\gamma} \tag{2.3.35}$$

Optimal Lagrange multiplier

With both the distortion (2.3.30) and the constraint (2.3.35) derived, the optimal Lagrange multiplier can be found as:

$$\frac{\mathrm{d}J(\lambda)}{\mathrm{d}\Delta} = \frac{\mathrm{d}D}{\mathrm{d}\Delta} + \lambda\frac{\mathrm{d}R}{\mathrm{d}\Delta} = 0 \tag{2.3.36}$$

Substituting both derivatives:

$$\frac{\Delta^2 a \mathrm{e}^{(-b\Delta/2)^\gamma}}{4} - \lambda a \mathrm{e}^{(-b\Delta/2)^\gamma} = 0 \quad \Rightarrow \quad \boxed{\lambda = \frac{\Delta^2}{4}} \tag{2.3.37}$$

This proves how the Lagrange multiplier is only related to the quantisation step. In other words, once this value is fixed, so is the optimal balance between the distortion and the rate constraint.

An important consequence of using the ℓ_0 norm is that the optimal Lagrange multiplier is independent from the data's PDF (it does not depend on σ neither on γ), meaning that the optimal Lagrange multiplier remains the same, no matter which transform has been used.

In fact, these results can be generalised to any PDF, making the ℓ_0 norm a robust approximation of the rate. This property makes the ℓ_0 norm suitable for iterative learning methods where the transform changes at each iteration and so does the PDF of the training data in the transform domain.

2.3.4 Independence from the PDF

In the previous subsection, for a given quantisation step, the value of the optimal Lagrange multiplier λ has been found for the particular case of the residuals displaying PDF that can be modelled after a GGD. By using the fundamental theorem of calculus, which relates integrals and derivatives of a function, one can generalise that conclusion for any continuous PDF.

Let $f(x)$ be the residuals PDF. The distortion is computed reusing (2.3.25):

$$D = \int_{-\Delta/2}^{\Delta/2} x^2 f(x) \mathrm{d}x \tag{2.3.38}$$

Deriving the distortion with respect to Δ:

$$\frac{\mathrm{d}D}{\mathrm{d}\Delta} = \frac{\Delta^2}{8}\left[f\left(\frac{\Delta}{2}\right) + f\left(-\frac{\Delta}{2}\right)\right] \tag{2.3.39}$$

The rate constraint from (2.3.31) is computed as follows with the generic PDF $f(x)$:

$$R = 1 - \int_{-\Delta/2}^{\Delta/2} f(x) \mathrm{d}x \tag{2.3.40}$$

Again, deriving with respect to Δ:

$$\frac{\mathrm{d}R}{\mathrm{d}\Delta} = -\frac{1}{2}\left[f\left(\frac{\Delta}{2}\right) + f\left(-\frac{\Delta}{2}\right)\right] \tag{2.3.41}$$

Combining previous equations using (2.3.36), is made clear that the optimal Lagrange multiplier λ does not depend on the residuals PDF:

$$\frac{\Delta^2}{8}\left[f\left(\frac{\Delta}{2}\right) + f\left(-\frac{\Delta}{2}\right)\right] - \lambda\frac{1}{2}\left[f\left(\frac{\Delta}{2}\right) + f\left(-\frac{\Delta}{2}\right)\right] = 0 \tag{2.3.42}$$

$$\boxed{\lambda = \frac{\Delta^2}{4}} \tag{2.3.43}$$

This fact makes the ℓ_0 norm suitable even for distributions that cannot be modelled after a GGD. The independence from the PDF comforts the choice of the ℓ_0 norm over other models that might provide a more realistic and accurate approximation of the rate, such as the entropy.

An example of the ℓ_0 norm independence from the PDF is provided below. Consider the following scenario: some residuals PDFs that follow GGD with different exponents γ. Consider, as well, a quantisation step $\Delta = 14.256$, issued from a QP 27 in HEVC. It has been demonstrated previously, that $J(\lambda)$ from (2.3.2) achieves its minimum value at $\lambda = \frac{\Delta^2}{4} \approx 50.81$, independently from the residuals PDF when modelling the rate with the ℓ_0 norm.

However, if instead of using the ℓ_0 norm, entropy (H) is used, it can no longer be assumed that the optimal value of λ providing the best trade-off between distortion and rate does not depend on the residuals PDF. Due to the complexity of the calculations involved then using the entropy together with a uniform quantiser, the dependency to the PDF will be evidenced through the example from figure 2.3.2. It shows that the value $J(\lambda)$ is plotted for various GGDs at different QP values, which served as the hard-threshold value. When using the ℓ_0 norm, $J(\lambda)$ reaches its minimum value at QP 27, as expected. On the other hand, using the entropy leads to a minimum value of $J(\lambda)$ that is PDF-dependent. An unwanted consequence of this dependence is that, for an iterative learning algorithm, after updating the transform at each iteration, the new PDF would have to be estimated to find the new optimal Lagrange multiplier λ, thus complicating the whole learning process and leading to probable instabilities.

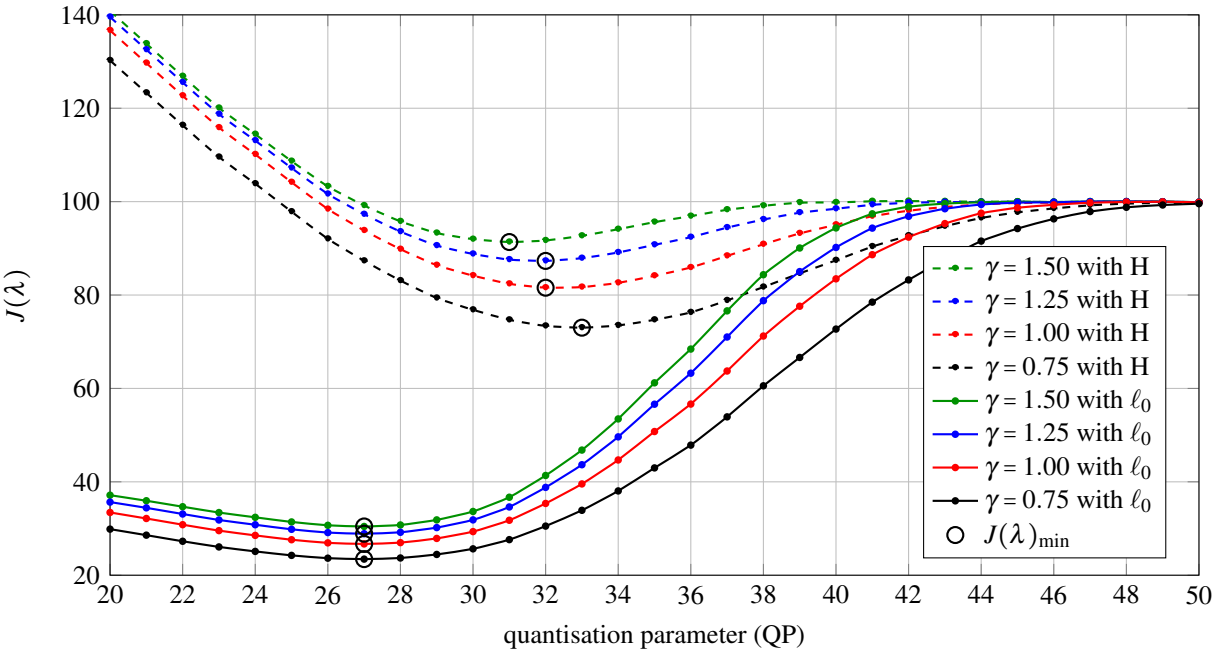


Figure 2.3.2 – $J(\lambda)$ for different GGDs modelling the rate with the ℓ_0 norm and the entropy (H)

2.3.5 Rate-distortion improvement through the learning

Assuming the source PDF can be modelled after a GGD with a given variance σ^2 and exponent γ, the impact of the learning in terms of the $J(\lambda)$ from (2.3.2) can be illustrated with the following example. By learning an adapted transform over a set of residuals, the amount of transformed coefficients mapped to zero increases, increasing its kurtosis (the distribution "looks more sharp"), hence reducing the exponent γ. Figure 2.3.3 illustrates the relationship between the distortion and the rate model, at different QPs. QPs 27 and 32 are highlighted, corresponding to quantisation steps $\Delta = 14.256$ and $\Delta = 25.504$ in HEVC, respectively.

As the exponent γ decreases, so do both terms of $J(\lambda)$, the distortion and the ℓ_0 norm. It can also be seen that since the Lagrange multiplier value λ does not change, neither does the slope of a tangent line to the circled points, corresponding to the optimal trade-off between the distortion and the rate constraint. The amount of improvement in each direction is weighted by λ: at QP 27 there is more room for sparsity improvement than there is for reducing the distortion, compared to QP 32.

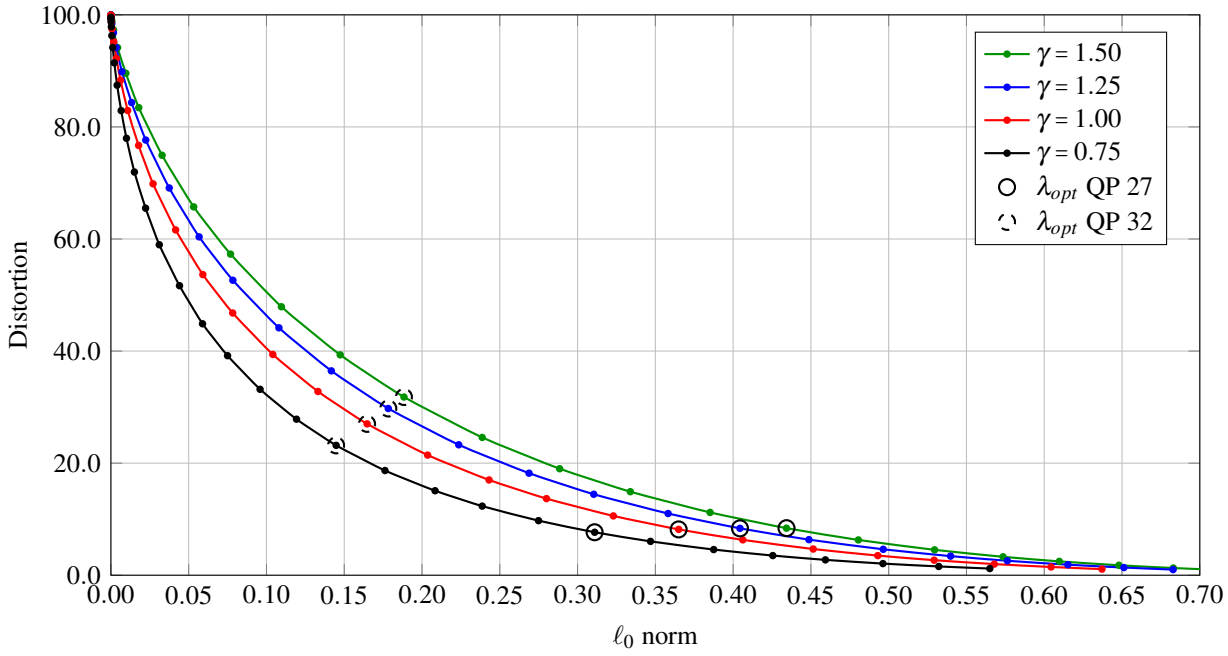


Figure 2.3.3 – Distortion (MSE) and ℓ_0 norm with λ_{opt} at QPs 27 and 32 for different PDFs

2.4 Conclusions

This chapter has introduced different types of transforms and some basic concepts, such as separability and its relation to computational complexity.

Two different approaches of transform design have been studied: the KLT and the RDOT. The KLT defines the transform providing optimal bit allocation under the high-resolution hypothesis. The DCT and DST used in HEVC have been obtained from the KLT for particular kinds of signals.

The RDOT finds the optimal balance between the distortion introduced by the quantiser and a rate constraint, expressed in terms of signal sparseness. A detailed study on its design, based on the ℓ_0 norm, has been carried out to justify the appropriateness of the approach, lacking in current literature.

In the next Chapter, both transform designs will be tested in a real scenario to evaluate their performances in video coding.

Chapter 3

The mode-dependent directional transforms

Contents

3.1 Introduction

The previous Chapter has introduced two different transform design approaches. In order to test their performances for video coding, a technique called Mode-Dependent Directional Transform (MDDT) is developed in this section. The principle of the MDDT lies on using an adapted transform learnt specifically for each Intra Prediction Mode (IPM).

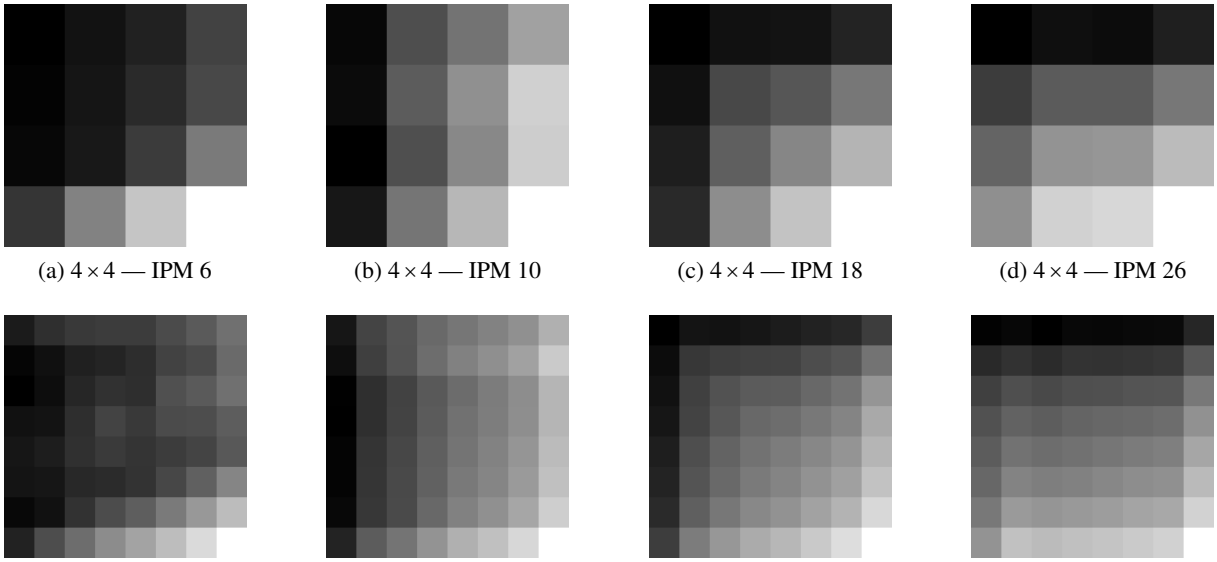
The MDDT technique was proposed during the HEVC standardisation phase, but it was finally discarded in favour of a new transform, the DST as explained in §3.1.2.

3.1.1 Motivation and principles of the MDDT

Depending on the selected prediction mode, transformed coefficients might present different patterns, making low and high values appear at different block positions. This heterogeneity can be harmful for the entropy coding, which is one of the reasons why different scanning patterns exist in HEVC [48]. These scanning patterns, presented in the lower part of figure 1.3.3 depend only on the IPM used to compute the residual.

The MDDT was motivated by the fact that residuals issued from different IPMs present notable differences. An example of these differences is presented in figure 3.1.1 for 4×4 and 8×8 prediction residuals. An adapted transform for each IPM can provide good signal compaction by specialisation, which means having an adapted transform for each one of the 35 IPMs.

A possible implementation of the MDDT inside a video coding scheme is illustrated in figure 3.1.2. This encoder works in the same way as the hybrid encoder from figure 1.3.1, with the exception that, in this case, each IPM will be tested in the RDO loop with its corresponding transform. Three examples of residuals are shown: for IPMs 10, 18 and 26, which are obtained as the differences between the predictions derived from their corresponding IPMs and the original image. The IPM 10 stands for a



(a) 4×4 — IPM 6 (b) 4×4 — IPM 10 (c) 4×4 — IPM 18 (d) 4×4 — IPM 26

(e) 8×8 — IPM 6 (f) 8×8 — IPM 10 (g) 8×8 — IPM 18 (h) 8×8 — IPM 26

Figure 3.1.1 – Average residual profiles for different IPMs

horizontal prediction from left blocks, 18 for the diagonal, from upper and left blocks and 26 for the vertical prediction from the upper blocks. When inside the RDO loop, each intra prediction residual is tested using the assigned transform. Then, the IPM offering the best rate-distortion trade-off is selected and signalled into the bitstream.

On the decoder side (figure 3.1.3), the only needed information to decode the block is represented in coloured blocks: the transformed coefficients and the used IPM. No additional information has to be sent to the decoder, with regards to the hybrid video coding scheme: the signalled IPM allows the decoder to know which transform to use to convert the transformed coefficients back to the pixel domain, without having to perform any extra calculations or take any decisions.

One of the positive aspects of the MDDT is the fact that, despite having more than one transform for a particular block size, only the transform corresponding to the selected IPM is used to transform that residual. This means that, the IPM conditions the transform, so there is no extra step in the RDO loop for the transform. However, there might be an increase of the complexity on both MDDT encoder and decoder due not to the number of transforms, but to the possible lack of fast algorithms for the adapted transforms implementation.

3.1.2 The DST as a simplification of the MDDT

The DST has been presented as the KLT for intra prediction residuals in §2.2.1. The way of generating the prediction residuals leads to a particular kind of signals with properties that differ from those of natural images. As a result, the DCT is no longer a good approximation of the KLT. However, it was not straightforward to realise that the KLT for those signals is the DST: many studies, detailed below, were carried out in order to find better performing transform for these signals.

Over the HEVC standardisation phase, various techniques to improve the performance of H.264/MPEG-4 AVC were explored and designated as Key Technical Areas (KTA) [54]. Amongst those techniques, in order to improve the coding performance of intra prediction residuals, an adapted transform for each IPM (at that time 9, which evolved into 35 in HEVC), was included in the KTA software. The adapted transforms were KLTs, and the resulting technique was called MDDT. It became a core component of the Test Model under Consideration (TMuC) [27] due to its performance improvements over the previous standard.

During the JCT on Video Coding (JCT-VC) meetings, many efforts were done to yield the first

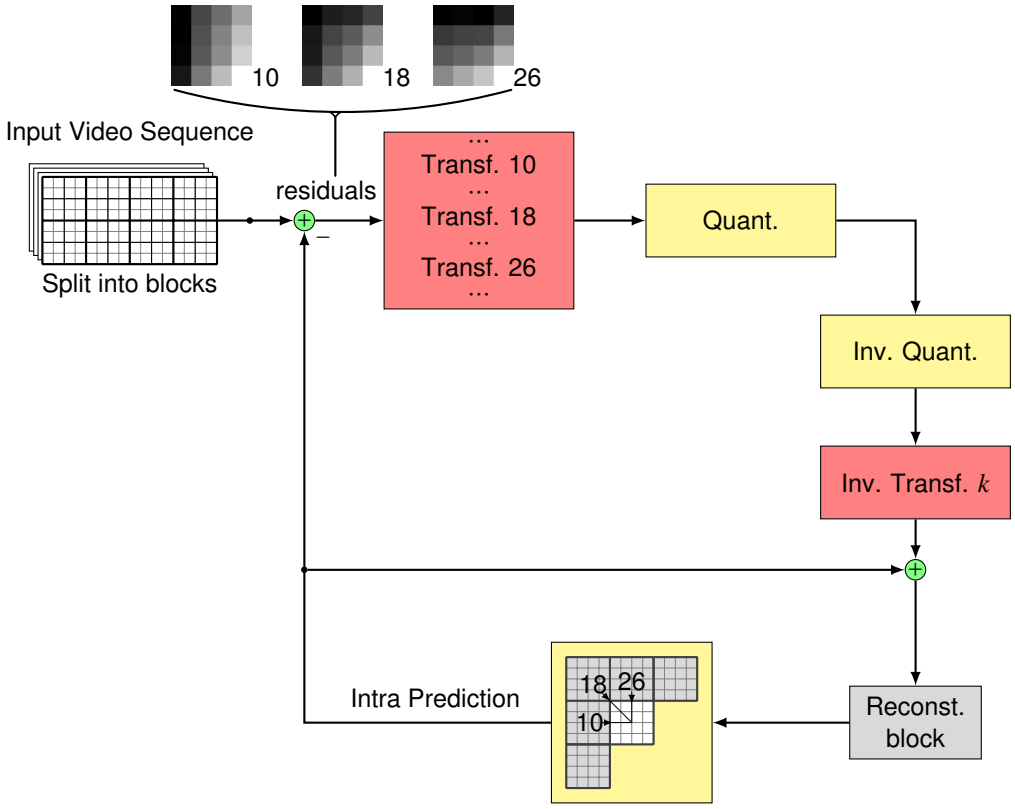


Figure 3.1.2 – Encoding scheme for the MDDT

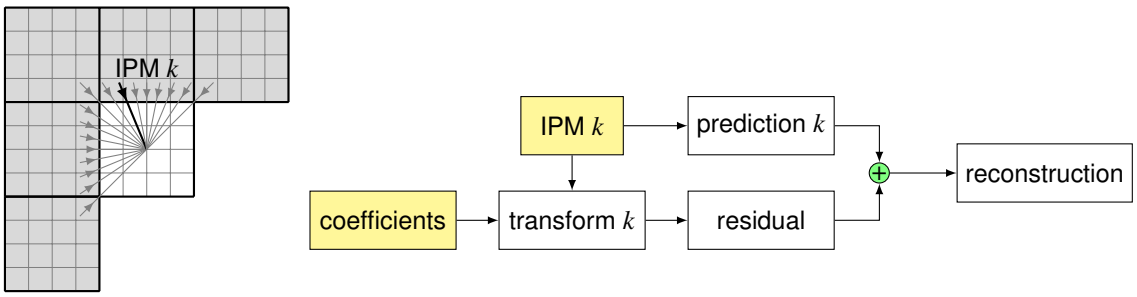


Figure 3.1.3 – Decoding scheme for the MDDT

implementation of the MDDT, which was non-separable, more lightweight [59].

One of the first attempts was to make the KLTs separable [62]. A fast algorithm was derived for the 4×4 transform by analysing the correlation matrix of the intra prediction residuals, which can be modelled using tridiagonal matrices [64]. In order to further reduce the complexity, rotational transforms were also considered in [16]. The main idea was to use the DCT followed by a secondary transform implemented in the form of Given's rotations [58] to improve the coding efficiency.

The DST was theoretically proved to be the optimal transform, in terms of KLT approximation, of the intra prediction residuals in [40, 41]. It was used together with the DCT as combinations of horizontal and vertical transforms, depending on the IPM. However, it was not adopted in the TMuC until the design of a fast algorithm of the DST [44], which introduced limited increase in the decoding time with regards to the DCT. It was only then when it fully replaced the MDDT [61, 42, 43, 45]. Finally, after many Core Experiments (CEs), cross-checks between companies and combinations of DST, DCT on different block sizes and luma and chroma channels, the whole system was simplified by using the DST for all 4×4 luma intra prediction residuals, and the DCT for the rest of the block sizes, due to the lack of fast algorithms and limited gains, in both luma and chroma [52].

In the following section, the MDDT is re-implemented using both transforms designs from §2.2 (KLT) and §2.3 (RDOT), in both separable and non-separable manners, to empirically compare its performance with regards to the current standard, HEVC and unveil the merits of the transform designs.

3.2 Design and implementation of MDDT systems

The preliminary work carried out for this section has been published in the form of an article [3].

3.2.1 MDDT system learning

The transform designs presented in the previous Chapter, namely the KLT and the RDOT, are tested in this Section, through the MDDT technique. This experiment reveals the ability of each design approach to adapt to a particular kind of signals and to fit the video coding demands.

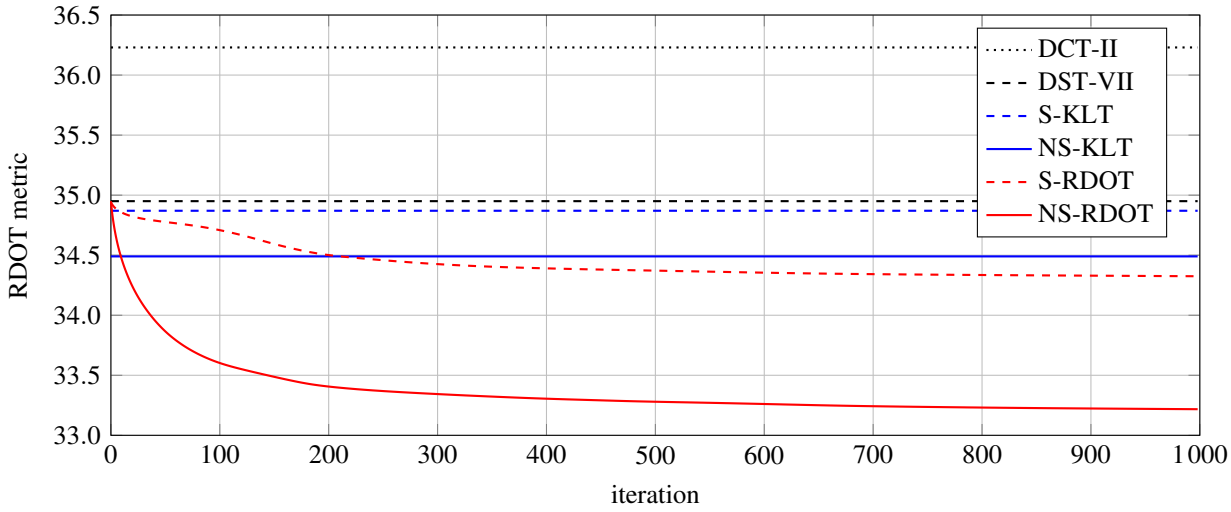
To design a transform, a learning set is needed. Since transforms are being designed to improve upon the MDDT, the learning set has been built with intra prediction residuals only, issued from HEVC encodings at four QPs (22, 27, 32, 37), coming from classes B and C from the HEVC test set, defined by the JCT-VC in the CTC [11]. These 2 classes contain sequences of 1920×1080 and 832×480, respectively, and cover various frame rates of 24, 30, 50 and 60 frames per second. As a result, the number of the residuals used for the learning set exceeds the 96 million for 4×4 TUs and 140 million for 8×8 TUs. Originally, the MDDT was designed to work with 4×4 blocks, but in this thesis, 8×8 blocks are also considered.

These residuals, grouped by the IPM they are issued from, are used to compute an adapted transform for each block size and for each IPM using both KLT and RDOT design approaches in separable and non-separable versions.

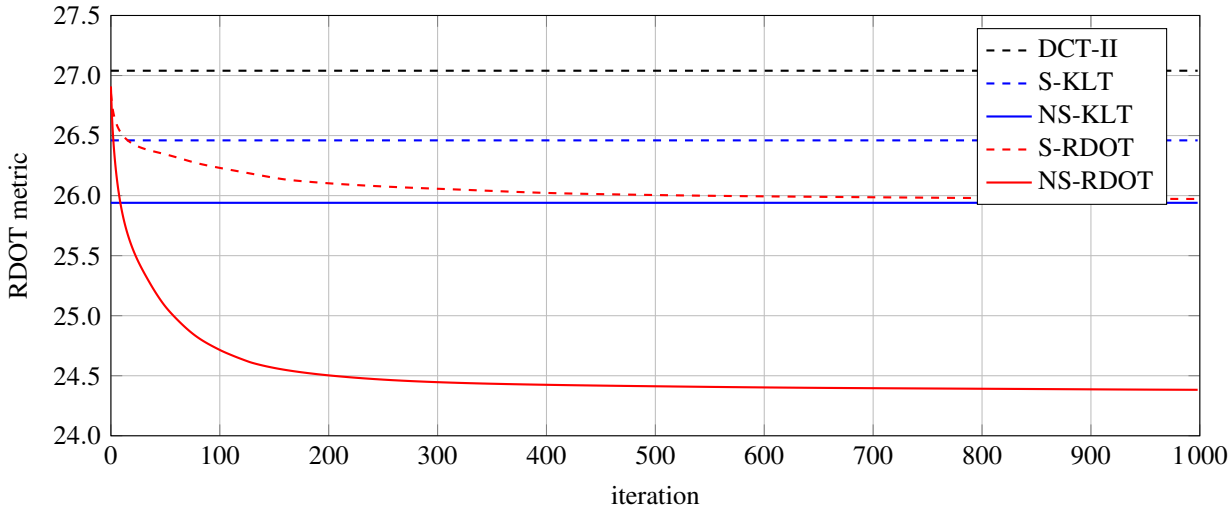
To illustrate the results of the iterative learning algorithm for the RDOT, presented in §2.3, figure 3.2.1 shows the value of the RDOT metric during the learning phase, averaged across all IPMs, for separable and non-separable designs.

The learning algorithm of the RDOT converges smoothly for both separable and non-separable designs. As expected, the non-separable RDOT presents a lower metric value than the separable version, since non-separable transforms are able to exploit linear correlations between any pair of pixels within a block, allowing for sparser signals in the transform domain.

Furthermore, figure 3.2.1 also contains the RDOT metric evaluated on the KLT, both separable and non-separable, and the default HEVC transforms. The use of this metric confirms some points stated in §2.2.2 in the BD-rate domain that justified the use of the DST-VII over the DCT-II for 4×4 blocks: the value of the metric using the DST-VII is notably lower than that of the DCT-II, and the fact that the DST for these blocks is close to the separable KLT is visible in terms of the RDOT metric.



(a) 4×4 transforms



(b) 8×8 transforms

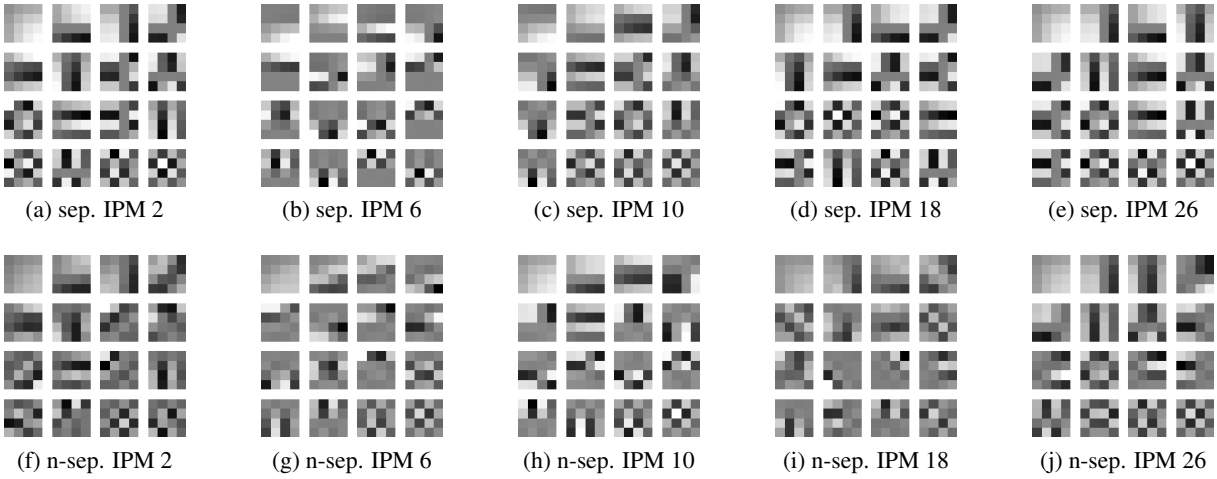
Figure 3.2.1 – Average RDOT metric evolution during different transform learnings: separable (S) and non-separable (NS)

Some examples of 4×4 learnt RDOTs are presented in figure 3.2.2. Although it is complicated to compare transform bases visually, non-separable transforms have captured the main direction for diagonal IPMs (2, 6, 18). These directions can be observed in the base vectors, especially in the first ones, as their patters exhibit the direction of the IPM. Separable transforms are not able to capture accurately those directions by construction, since they combine transforms operating row and column wise.

After having confirmed that the learning has output better performing transforms in terms of distortion and sparseness, their implementation on top of HEVC is tested and discussed in the next section.

3.2.2 MDDT results on video coding

The MDDT systems designed in the previous Section have been used in HEVC using the common test conditions, defined by the JCT-VC [11], for AI and RA coding configurations. The common test conditions consist in encoding each sequence at four different QP values (22, 27, 32, 37) and computing the average bitrate savings with regards to HEVC using the BD-rate metric.



(a) sep. IPM 2 (b) sep. IPM 6 (c) sep. IPM 10 (d) sep. IPM 18 (e) sep. IPM 26

(f) n-sep. IPM 2 (g) n-sep. IPM 6 (h) n-sep. IPM 10 (i) n-sep. IPM 18 (j) n-sep. IPM 26

Figure 3.2.2 – 4×4 separable and non-separable RDOT for different IPMs

	4×4				8×8				4×4 & 8×8			
	sep		non-sep		sep		non-sep		sep		non-sep	
Cl.	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT
A	0.08	0.12	-0.18	-0.17	-0.96	-0.98	-1.60	-1.56	-0.88	-0.90	-1.63	-1.60
B	-0.07	-0.20	-0.50	-0.80	-0.40	-0.76	-1.46	-2.47	-0.42	-0.93	-1.65	-2.77
C	-0.18	-0.60	-1.46	-2.47	-0.42	-0.82	-2.40	-4.03	-0.51	-1.33	-3.09	-5.15
D	-0.12	-0.58	-1.17	-2.04	-0.23	-0.56	-1.17	-2.08	-0.27	-1.06	-1.88	-3.41
E	0.39	0.35	-0.07	-0.45	-0.92	-1.20	-2.11	-2.91	-0.57	-0.95	-1.83	-3.00
F	0.10	-0.78	-0.57	-2.04	0.09	-0.22	-1.12	-2.37	0.25	-0.93	-1.20	-3.51
Av.	0.02	-0.31	-0.67	-1.34	-0.45	-0.74	-1.62	-2.55	-0.39	-1.02	-1.87	-3.23

Table 3.2.1 – Average bitrate savings (%) for each HEVC Class in AI

Bitrate savings

Tables 3.2.1 and 3.2.2 contain the performances of different MDDT systems for AI and RA coding configurations, respectively, for the different sequence classes of the HEVC test set using the BD-rate metric. The last line of the table represents the average bitrate savings for all sequences (it does not correspond to the averaged bitrate savings per class).

Three system configurations are reported: a MDDT system operating on 4×4 only TUs, 8×8 only TUs and jointly on both sizes. The first column of both tables refers to the separable KLT-based MDDT for 4×4 TUs, which, as demonstrated in [24, 25], corresponds to the use of a DST. The average bitrate savings with regards to HEVC for this system are insignificant since HEVC already uses the DST for those blocks.

The differences between both transform learning approaches (KLT and RDOT) can be observed by looking at each pair of columns. Results are very consistent with what was anticipated in the RDOT metric domain: the RDOT-based MDDT outperforms the KLT in every case (separability, TU size, class and coding configuration).

It is also worth-noticing the impact of separability: non-separable configurations provide systematically higher bitrate savings than the separable ones, especially for those systems involving 8×8 TUs.

For the combined 4×4 and 8×8 system in AI the bitrate savings are over 3% and almost 2% in RA. The KLT systems are about one point below. The detailed performances of the combined systems are included in table 3.2.3. Despite having used classes B and C for the transform learning, consistent bitrate savings are achieved across different resolutions among classes. Moreover, classes D and E (not included in the learning set) present higher bitrate savings than class B. It is also worth noticing that the RDOT systems do not present losses for any sequence, which is not the case for the KLT.

	4×4				8×8				4×4 & 8×8			
	sep		non-sep		sep		non-sep		sep		non-sep	
Cl.	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT
A	0.03	0.05	-0.03	-0.07	-0.29	-0.42	-0.71	-0.80	-0.27	-0.37	-0.70	-0.87
B	0.00	-0.10	-0.26	-0.51	-0.24	-0.46	-0.86	-1.42	-0.25	-0.55	-0.99	-1.63
C	-0.08	-0.36	-0.77	-1.41	-0.20	-0.39	-1.18	-2.22	-0.23	-0.70	-1.57	-2.88
D	-0.03	-0.25	-0.50	-1.05	-0.08	-0.25	-0.53	-1.05	-0.11	-0.50	-0.88	-1.73
E	0.10	-0.19	-0.36	-0.87	-0.31	-0.74	-1.02	-2.04	-0.27	-0.91	-1.14	-2.47
F	0.07	-0.59	-0.29	-1.26	0.08	-0.10	-0.50	-1.42	0.18	-0.65	-0.56	-2.21
Av.	0.01	-0.24	-0.37	-0.85	-0.17	-0.38	-0.80	-1.47	-0.16	-0.60	-0.97	-1.93

Table 3.2.2 – Average bitrate savings (%) for each HEVC Class in RA

A remarkable point stands out of the test set: the *BasketballDrill* sequence from class C, with bitrate savings of almost 12% using the non-separable RDOT in AI. This is due to the fact that this sequence presents strong directional patterns that cannot be dealt with separable transforms. Almost all the performance is lost when using separable systems. For illustrative purposes, figure 3.2.3 represents graphically the results from table 3.2.3. The behaviour in the RA coding configuration is very similar to the AI, with the bitrate savings in RA being around two thirds of those in AI.

As a reminder, all transforms learnt for the presented MDDT systems have used residuals issued from AI coding configurations and have only been enabled for intra coded residuals in HEVC. Nonetheless, around two thirds of the bitrate savings achieved for the AI coding configurations have been achieved in RA. This is due to the fact that I-frames, and intra coded blocks in RA are of higher quality, and serve, therefore, as better references to derive the temporal predictions.

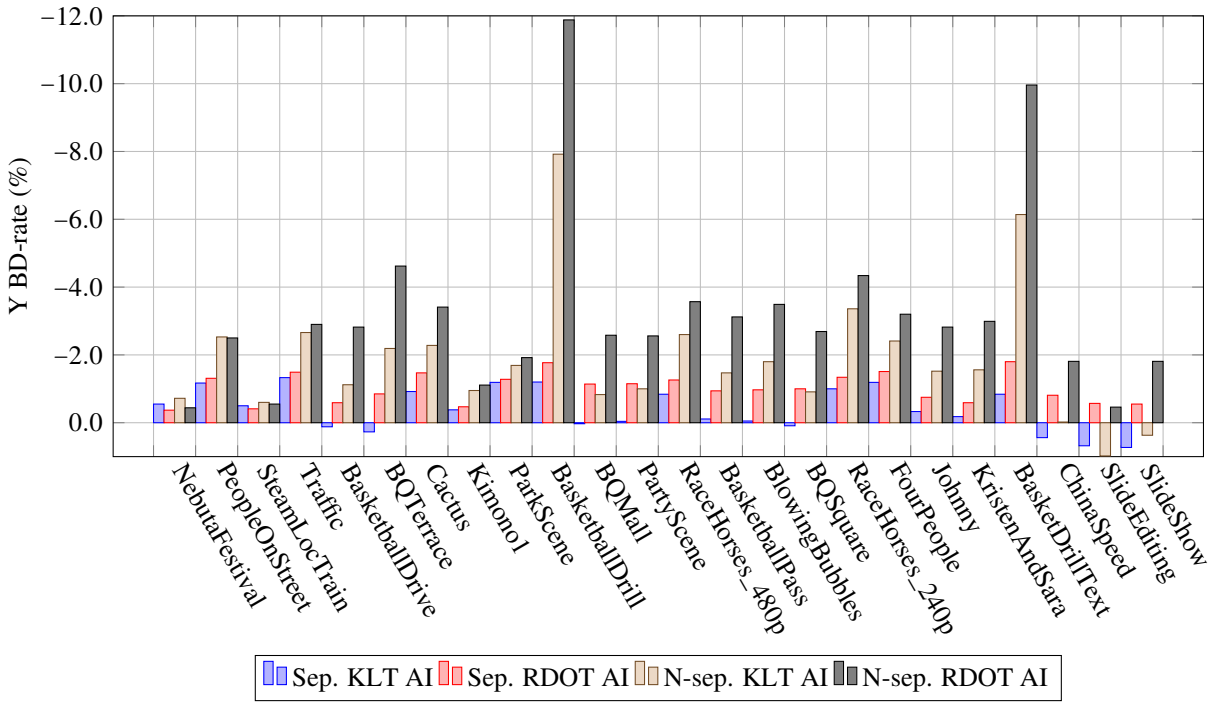
A final comment on the designed transforms regarding the scanning of the transformed coefficients: the transform base vectors for each IPM have been sorted taking into account the scanning that HEVC performs implicitly, illustrated in figure 1.3.3. For non-separable transforms, base vectors can be placed in any order, but separable transforms does not have this freedom, as such, an adapted scanning matrix is needed to make sure the transform coefficients are properly sorted.

Coding complexity

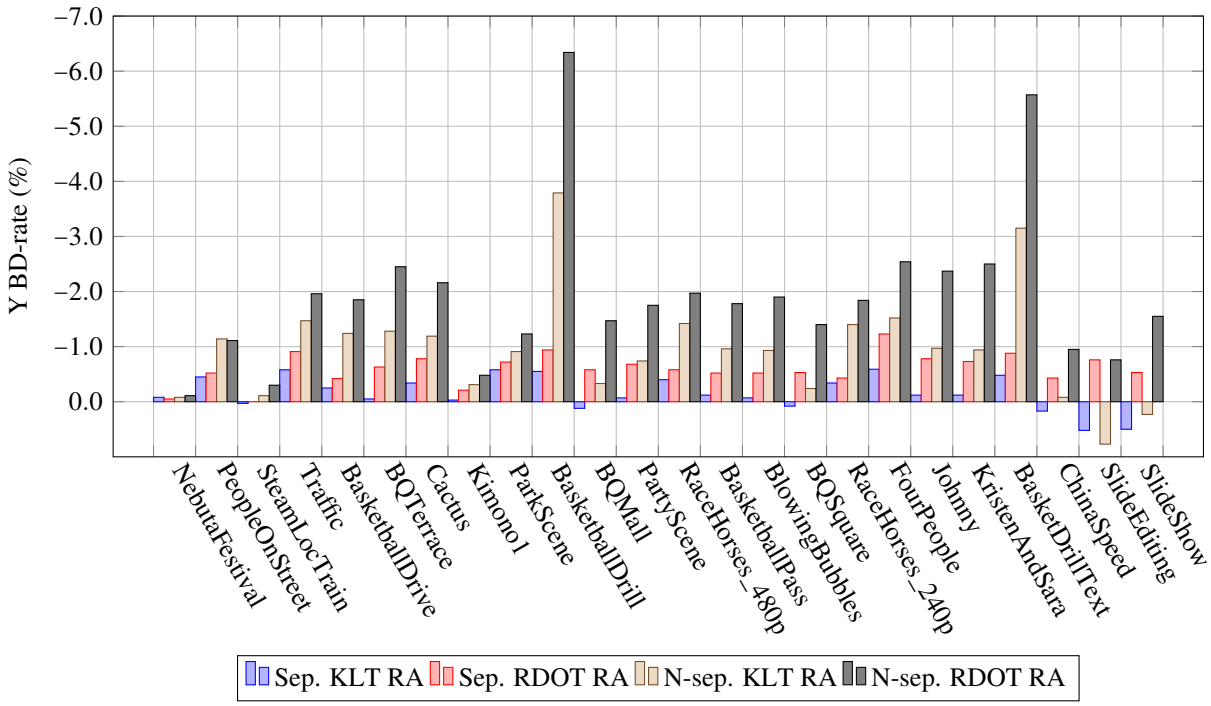
Regarding the complexity of the system, the increase in the encoding and decoding time is only due to the lack of fast algorithms for KLTs and RDOTs. Due to the way the quad-tree partitioning works in HEVC, if, for instance, a 16×16 TU provides a better rate-distortion trade-off than splitting it into 8×8 TUs, the 4×4 will not be explored. Since the MDDT system improves 4×4 and 8×8 TUs, it becomes more likely that splitting a 16×16 TU into four 8×8 TUs works better than not splitting. In that case, the 4×4 TUs will also be explored, which will lead to an increase in complexity with regards to HEVC, as more coding possibilities are being explored. Table 3.2.4 summarises the coding complexity for MDDT systems using separable and non-separable transforms for 4×4 and 8×8 TUs. The complexity for KLT and RDOT based systems is equivalent, since transforms are performed as generic matrix multiplications. As seen in §2.1.3, the number of operations required for a non-separable transform is about N times the amount for separable transforms, with N being the TU size. For 4×4 TUs, the complexity in both coding and decoding times remains approximately the same as that of HEVC. However, for larger TUs, the differences due to fast algorithms start becoming more noticeable.

MDDT storage requirements

The transforms used in the MDDT system are, not describable with a mathematical formula unlike the DCT and the DST. As such, additional memory is required to store these transforms and make them available to the encoder and the decoder. Each transform coefficient has been quantised to 1 byte by rounding it to the nearest integer. More advanced quantisation techniques could have been implemented



(a) AI coding configuration



(b) RA coding configuration

Figure 3.2.3 – Bitrate savings for combined 4×4 & 8×8 MDDT systems

		4×4 & 8×8							
		AI				RA			
		sep		non-sep		sep		non-sep	
	Sequence	KLT	RDOT	KLT	RDOT	KLT	RDOT	KLT	RDOT
Class A (2560×1600)	NebutaFestival	-0.55	-0.37	-0.72	-0.44	-0.08	-0.05	-0.08	-0.11
	PeopleOnStreet	-1.17	-1.31	-2.53	-2.50	-0.45	-0.52	-1.14	-1.11
	SteamLocTrain	-0.50	-0.41	-0.60	-0.55	0.03	0.00	-0.11	-0.30
	Traffic	-1.33	-1.49	-2.66	-2.90	-0.58	-0.91	-1.47	-1.96
	Average	-0.88	-0.90	-1.63	-1.60	-0.27	-0.37	-0.70	-0.87
Class B (1920×1080)	BasketballDrive	0.12	-0.59	-1.12	-2.82	-0.25	-0.42	-1.24	-1.85
	BQTerrace	0.27	-0.85	-2.19	-4.62	-0.05	-0.63	-1.28	-2.45
	Cactus	-0.92	-1.47	-2.28	-3.41	-0.34	-0.78	-1.19	-2.16
	Kimono1	-0.38	-0.47	-0.95	-1.11	-0.03	-0.21	-0.31	-0.48
	ParkScene	-1.19	-1.28	-1.69	-1.92	-0.58	-0.72	-0.91	-1.23
	Average	-0.42	-0.93	-1.65	-2.77	-0.25	-0.55	-0.99	-1.63
Class C (832×480)	BasketballDrill	-1.20	-1.77	-7.92	-11.88	-0.55	-0.94	-3.79	-6.34
	BQMall	0.03	-1.14	-0.83	-2.58	0.12	-0.58	-0.33	-1.47
	PartyScene	-0.04	-1.15	-1.00	-2.56	-0.07	-0.68	-0.74	-1.75
	RaceHorses	-0.84	-1.26	-2.60	-3.57	-0.40	-0.58	-1.42	-1.97
	Average	-0.51	-1.33	-3.09	-5.15	-0.23	-0.70	-1.57	-2.88
Class D (416×240)	BasketballPass	-0.11	-0.94	-1.47	-3.12	-0.12	-0.52	-0.96	-1.78
	BlowingBubbles	-0.05	-0.97	-1.80	-3.49	-0.07	-0.52	-0.93	-1.90
	BQSquare	0.09	-1.00	-0.91	-2.69	0.08	-0.53	-0.24	-1.40
	RaceHorses	-1.00	-1.34	-3.36	-4.34	-0.34	-0.43	-1.40	-1.84
	Average	-0.27	-1.06	-1.88	-3.41	-0.11	-0.50	-0.88	-1.73
Class E (1280×720)	FourPeople	-1.19	-1.51	-2.41	-3.20	-0.59	-1.23	-1.52	-2.54
	Johnny	-0.33	-0.75	-1.52	-2.82	-0.12	-0.78	-0.97	-2.37
	KristenAndSara	-0.18	-0.59	-1.56	-2.99	-0.12	-0.73	-0.94	-2.50
	Average	-0.57	-0.95	-1.83	-3.00	-0.27	-0.91	-1.14	-2.47
Class F (various resolutions)	BasketDrillText	-0.84	-1.80	-6.14	-9.96	-0.48	-0.88	-3.15	-5.57
	ChinaSpeed	0.44	-0.81	-0.02	-1.81	0.17	-0.43	-0.08	-0.95
	SlideEditing	0.68	-0.57	0.98	-0.46	0.52	-0.76	0.77	-0.76
	SlideShow	0.73	-0.55	0.37	-1.81	0.50	-0.53	0.23	-1.55
	Average	0.25	-0.93	-1.20	-3.51	0.18	-0.65	-0.56	-2.21
All Sequences	Overall	-0.39	-1.02	-1.87	-3.23	-0.16	-0.60	-0.97	-1.93

Table 3.2.3 – Detailed bitrate savings (%) for combined 4×4 & 8×8 MDDT systems

that would lead to lower degradations [12]. The rounding method does not guarantee that the transform remains orthogonal and that the energy preservation property in the RDOT is retained, although experiments done with this quantisation show that observed losses are in the order of 0.2 BD-rate points. Taking all this into account, the storage requirements for a non-separable transform for $N \times N$ TUs is:

$$\mathrm{ROM}_{\text{non-sep}} = N^2 \cdot N^2 \cdot \frac{1\ \mathrm{kB}}{1024\ \mathrm{B}} \cdot M \tag{3.2.1}$$

The needed ROM for separable transforms is:

$$\mathrm{ROM}_{\text{sep}} = 3 \cdot N \cdot N \cdot \frac{1\ \mathrm{kB}}{1024\ \mathrm{B}} \cdot M \tag{3.2.2}$$

Where the number of IPMs M is 35 for the MDDT system. Sorting the transform base vectors in the proper order is essential to assist the entropy coding [60]. For this reason, there is a factor 3 for the ROM requirements in the separable case: horizontal transform, vertical transform and scanning matrix for each pair of transforms. The non-separable transforms do not need a scanning matrix, since the base vectors can be sorted to output the signal with an appropriate coefficients order. Table 3.2.4 contains the ROM

	4×4		8×8		4×4 & 8×8	
	sep	non-sep	sep	non-sep	sep	non-sep
Y BD-rate	-0.31%	-1.34%	-0.74%	-2.55%	-1.02%	-3.23%
Encoding	101%	102%	107%	111%	108%	112%
Decoding	101%	102%	103%	115%	105%	120%
ROM	1.64 kB	8.75 kB	6.56 kB	140.00 kB	8.20 kB	148.75 kB

Table 3.2.4 – Summary of RDOT-based MDDT systems compared to HEVC in AI

requirements for the MDDT systems in both separable and non-separable designs. This table illustrates the drawback of using non-separable transforms regarding the storage. Even the MDDT system using only non-separable transforms for 4×4 TUs needs more ROM than the complete separable MDDT system making use of both sizes.

3.3 Conclusions

Transform designs presented in Chapter 2, namely the KLT and the RDOT, have been evaluated through the MDDT, a technique for intra coded blocks that consists in designing an adapted transform per IPM. During the learning phase, the RDOT metric has presented coherent results with the knowledge acquired from literature regarding the KLT, DCT and DST. These transforms have been evaluated using the RDOT metric and have been confirmed by the results obtained in the BD-rate domain: for intra prediction residuals, the DST-VII has a better BD-rate score than the DCT-II, and so does using the RDOT metric.

The designed transforms have been tested in a modified version of HEVC using the MDDT technique. Non-separable transforms present a significant improvement in terms of bitrate savings over their separable counterparts. Moreover, the RDOT design approach has provided better results in terms of BD-rate than the KLT all over the tested sequences, with no losses for any of them. Bitrate savings between around 2–3% are achieved when using the RDOT-based MDDT, which validate the transfom design over the KLT-based design, providing bitrate savings around 1 point below. Consequently, the KLT will no longer be considered in the upcoming sections, only systems designed using the RDOT metric will be considered.

The complexity of the MDDT systems remains comparable to that of HEVC, since the modification only implies that default transform is replaced with an adapted one. Therefore, the increase in complexity is only due to the fact that transforms are implemented as matrix multiplications due to the lack of fast algorithms for general orthogonal transforms.

The following Chapter presents an improvement of the MDDT, which will allow transforms to be more adapted to intra prediction residuals and provide new coding alternatives to HEVC.

Chapter 4

The mode-dependent transform competition system

Contents

4.1 Introduction

Chapter 3 revisited the existing Mode-Dependent Directional Transform (MDDT) technique, its origins and motivation, and summarised how it was discarded in the final HEVC standard in favour of a more simplified approach using the DST. However, the MDDT technique has good potential in bitrate savings when used with RDOTs, especially in their non-separable design.

This Chapter is focused on improving upon the MDDT technique by increasing the number of available transforms. The main idea is to provide a fixed number of transforms in each Intra Prediction Mode (IPM) that compete against each other in the RDO loop, in the same ways as block sizes and IPMs do. This implies that, for a given block size and IPM, there is no longer a unique transform, but a set of them, and the encoder selects the one that provides the best trade-off in terms of rate-distortion. This evolution of the MDDT system has been named Mode-Dependent Transform Competition (MDTC) system [5].

Transform competition is not an entirely new concept in HEVC. A rudimentary form of competition is present for 4×4 intra predicted luma residuals: the choice between using the DST-VII and not using a transform at all already exists. This behaviour is controlled by the `transform_skip_flag` [34, 35].

Some work on transform competition has been carried out in [4, 65]. The main difference is that, in this Chapter, each IPM has its own transform sets instead of a common set for all modes. The approach from [65] uses the same transforms for all IPMs. The block size and IPM decision is based on the default HEVC transforms, then the residual is consequently transformed with the transform that provides the best trade-off in terms of rate-distortion.

Further work on different mode-dependent tools for video coding have been explored in [33], notably ways of simplifying and improving the MDDT system by adding re-orderings in the residuals and additional transforms after the main transform.

Early work on MDTC systems closely related to this Chapter has been presented in [5].

4.2 Multiple transform design using the RDOT metric

4.2.1 HEVC reproducibility

Since the main purpose of the transforms in HEVC is to compact the energy of the signal in order to increase the bitrate savings, the design process is developed as follows: the default HEVC transforms for 4×4 and 8×8 luma blocks (the DST and DCT, respectively) are kept and a number of additional transforms are learnt to capture those residuals for which HEVC transforms are not adapted to. This conservative approach has been made to guarantee that the original HEVC coding choices are still available when using the MDTC system.

4.2.2 The learning algorithm

The chosen transform design is the RDOT, since, as demonstrated in the previous Chapter, it provides significant improvements in terms of bitrate savings with respect to the KLT. However, in this Chapter, more than one RDOT will be learnt per IPM, using the same learning set: residuals issued from an AI HEVC coding of classes B and C from the HEVC test set, grouped by IPM. For each IPM, 2^N transforms are learnt, in addition to the HEVC default transforms (for compatibility reasons). Algorithm 1 describes how the learning has been carried out in each IPM. Assuming the desired outputs are 2^N additional transforms, the performed steps are:

1. Initial random classification of the residuals into $1 + 2^N$ classes.
2. For the 2^N classes that are not assigned to the HEVC transform, learn a separable or non-separable transform, depending on the desired configuration.
3. Evaluate each residual using the RDOT metric and assign it to the transform that minimises the value.
4. Repeat steps 2 and 3 until convergence.

A conceptual example for $N = 1 \Rightarrow 2^N = 2$ additional transforms is provided in figure 4.2.1.This figure displays the residuals and how a they would be assigned to the transform that provides the lowest value of the RDOT metric. The algorithm is subject to improvements, since the initial conditions are difficult to define. Consequently, in order to increase the confidence level of not ending up into a local minimum, several runs per learning are done with different initialisations.

In order to confirm the coherence of the learnings, the RDOT metric has been evaluated for different learning configurations, depending on the N value for 4×4 and 8×8 blocks. Figure 4.2.2 presents the averaged RDOT metric value for all IPMs residuals when using an increasing number of transforms for 4×4 and 8×8 residuals. The starting point in 4.2.2.a and 4.2.2.b coincides with the RDOT metric evaluated on HEVC default transforms, which corresponds to the one shown in figure 3.2.1, from the previous Chapter.

The RDOT metric decreases with the number of transforms, but it stagnates when the number of transforms is high.

Furthermore, there is an important gap in the RDOT metric between separable and non-separable transforms. For 4×4 blocks, the value achieved with separable transforms, is achieved with half the number of non-separable transforms. The gap is even more important for 8×8 blocks.

input : Residuals **x** from a given Intra Prediction Mode
output: Set of 2^N RDOTs A_n

Initial random classification into $1+2^N$ classes

while *!convergence* **do**
  for $n = 1$ **to** 2^N **do**
    Learn a RDOT on Class$_n$ using (2.3.1) or (2.3.14), depending on separability
  end
  foreach *block* **x do**
    for $n = 0$ **to** 2^N **do**
      $\delta_n = \|\mathbf{x} - \mathbf{A}_n^T \mathbf{c}\|^2 + \lambda \|\mathbf{c}\|_0$
    end
    $n^* = \arg\min_n(\delta_n)$
    Class$_{n^*}$.append (**x**)
  end
end

Algorithm 1: Multiple transform design

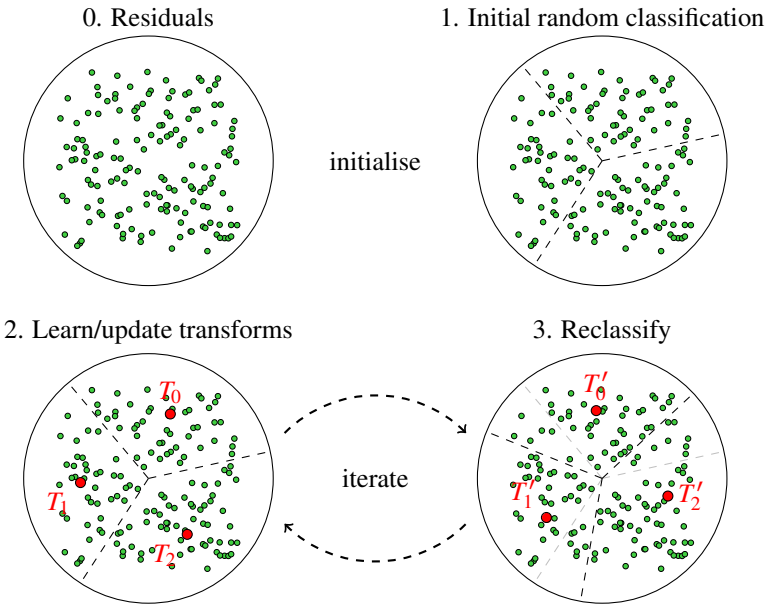
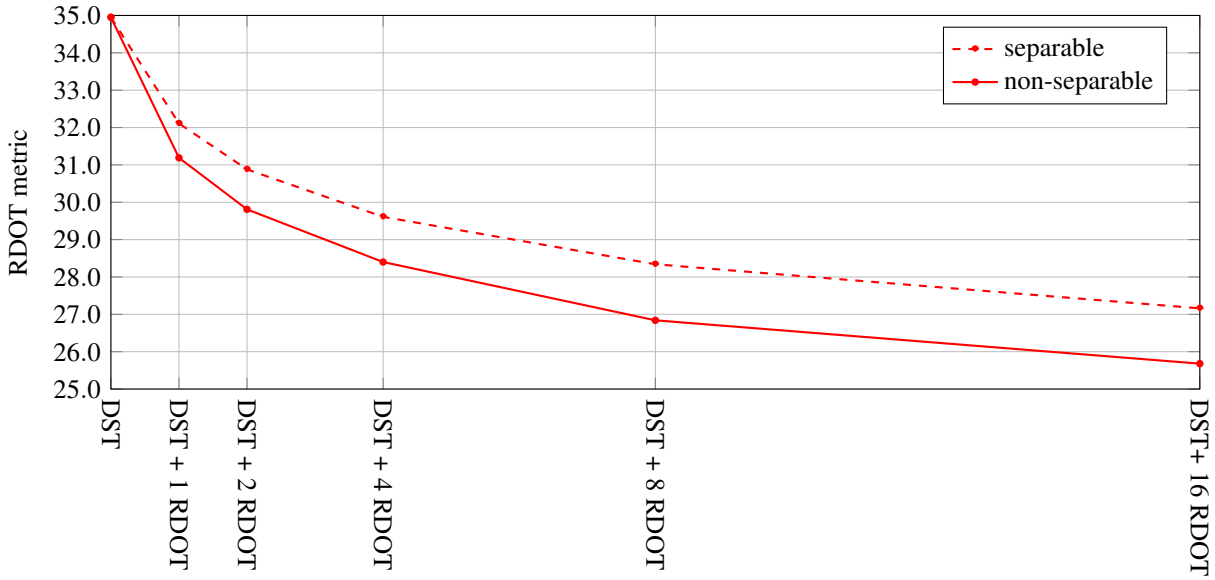
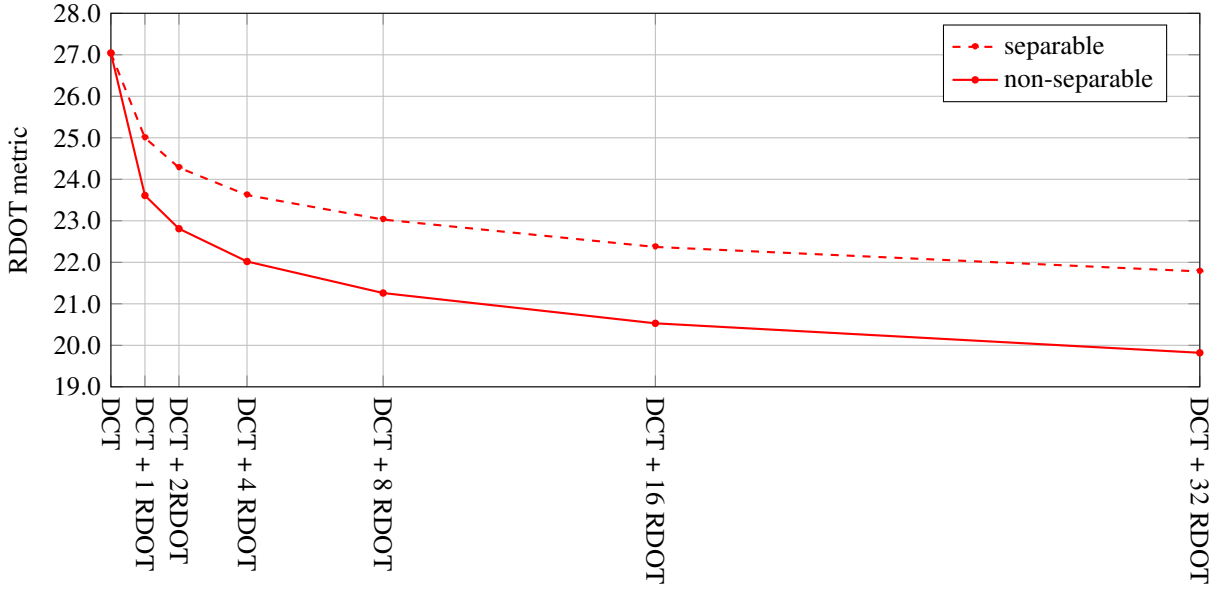


Figure 4.2.1 – Clustering and transform learning for a given set of residuals



(a) RDOT metric evolution with the number of 4×4 transforms



(b) RDOT metric evolution with the number of 8×8 transforms

Figure 4.2.2 – Average RDOT metric evolution for the learning set depending on the number of transforms

4.3 The MDTC system in video coding

This section presents the performances of the MDTC systems designed in the previous section on the full HEVC test set. Systems are evaluated in terms of bitrate savings, encoding complexity and decoding complexity.

4.3.1 Signalling the transforms in the bitstream

The main reason for learning $1+2^N$ transform configurations, with the first one being the default HEVC transform for the target block size, has been backward compatibility, as mentioned in the previous Section. Since the default transforms are kept, the encoder is able to choose them and reproduce the same coding choices as HEVC in case they provide the best trade-off. Moreover, this configuration also favours the signalling of the transforms. The chosen signalling strategy is to use a flag, which, when enabled, indicates that a different transform from the default HEVC is used for that TU size. The flag is conveyed though a CABAC encoder and uses a context for each TU size to help the entropy coding predict its value and to have negligible impact on performances in case it is never enabled. If the flag is enabled, a code word of fixed length (N bits) is used to signal the selected transform by the encoder among the 2^N remaining. This signalling strategy, apart from keeping backward compatibility with HEVC, favours a simple approach that matches with the fact that, during the learning phase, all additional transforms were used almost uniformly.

4.3.2 Performances of different configurations

Different MDTC systems have been considered and tested, corresponding to different number of additional transforms used for 4×4 and 8×8 TUs. All tests have been carried out following the common test conditions, stipulated in [11].

Figure 4.3.1 illustrates the relationship between the BD-rate and the number of transforms on the HEVC test set for the AI configuration. The BD-rate decreases approximately with a logarithmic law on the number of additional transforms. The behaviour in the BD-rate domain is close to the one observed in the RDOT metric domain in figure 4.2.2, despite the fact that signalling has not been taken into account during the learnings. Unlike during the transform learnings, additional transforms have been signalled during encoding tests to provide a decodable stream. This signalling counterbalances the sparseness increase in the transformed residuals and contributes to the saturation in BD-rate when the number of transforms is high.

For 4×4 transforms, bitrate savings of almost 2.5% can be achieved, whereas 6% are obtained when using only 8×8 transforms on the HEVC test set.

In order to find out the minimum and maximum performances of the combined MDTC systems, two systems have been evaluated on the full HEVC test set in AI and RA coding configurations, named:

— Low complexity system: one additional transform is set per IPM for both 4×4 and 8×8 TUs.

— High performance system: 16 additional transforms per IPM for 4×4 TUs and 32 for 8×8 TUs.

The number of transforms per block size has been chosen to guarantee that enough residuals are available for transform learning in each IPM and that the obtained performances are not counterbalanced by the transform signalling. The performances of the low complexity and high performance systems are summarised in table 4.3.1. The first thing to notice is that almost no losses are observed with regards to HEVC on any sequence: only *SteamLocomotiveTrain* presents small losses in the RA coding configuration. On the one hand, the low complexity system is able to achieve average bitrate savings of around 3.8% with non-separable transforms, and around 2.4% with separable transforms. On the other hand, the high complexity system provides average bitrate savings of over 4% for its separable version, and over 7% for the non-separable one. Around two thirds of the gain observed in the AI coding configuration are obtained in the RA.

Graphical representations of table 4.3.1 are shown in figures 4.3.2 and 4.3.3 for the low complexity and high performance MDTC systems, respectively.

		4×4: 1+1 — 8×8: 1+1				4×4: 1+16 — 8×8: 1+32			
		sep.		non-sep.		sep.		non-sep.	
	Sequence	AI	RA	AI	RA	AI	RA	AI	RA
Class A (2560×1600)	NebutaFestival	-0.34	-0.05	-0.52	-0.04	-1.06	-0.08	-1.15	-0.13
	PeopleOnStreet	-1.40	-0.48	-2.76	-1.16	-4.21	-1.49	-5.65	-2.27
	SteamLocTrain	-0.36	0.33	-0.46	-0.10	-0.60	0.23	-0.68	0.03
	Traffic	-1.68	-1.25	-3.07	-2.32	-4.52	-3.73	-6.06	-5.12
	Average	-0.94	-0.36	-1.70	-0.90	-2.60	-1.27	-3.38	-1.87
Class B (1920×1080)	BasketballDrive	-1.17	-0.35	-3.15	-1.76	-3.22	-0.72	-5.52	-2.36
	BQTerrace	-1.80	-1.08	-5.11	-2.83	-4.70	-2.65	-9.22	-4.93
	Cactus	-1.95	-1.15	-3.75	-2.42	-5.22	-3.08	-10.92	-7.68
	Kimono1	-0.46	-0.25	-1.06	-0.50	-1.10	-0.79	-1.80	-1.18
	ParkScene	-1.68	-1.09	-2.20	-1.50	-4.61	-3.22	-5.27	-3.69
	Average	-1.41	-0.78	-3.05	-1.80	-3.77	-2.09	-6.55	-3.97
Class C (832×480)	BasketballDrill	-2.09	-1.32	-12.83	-7.06	-5.92	-3.53	-25.06	-14.80
	BQMall	-2.04	-1.20	-3.27	-1.91	-4.79	-2.79	-6.21	-3.64
	PartyScene	-2.18	-1.40	-3.31	-2.29	-4.88	-3.21	-6.19	-4.30
	RaceHorses	-1.71	-0.70	-3.99	-2.16	-4.25	-1.59	-6.75	-3.24
	Average	-2.00	-1.15	-5.85	-3.36	-4.96	-2.78	-11.05	-6.49
Class D (416×240)	BasketballPass	-1.68	-0.82	-3.71	-1.97	-3.91	-1.82	-6.15	-3.14
	BlowingBubbles	-1.96	-1.21	-4.21	-2.35	-4.30	-2.69	-6.73	-3.95
	BQSquare	-2.26	-1.21	-3.72	-2.00	-4.58	-2.74	-6.12	-3.61
	RaceHorses	-1.57	-0.58	-4.61	-2.03	-3.85	-1.54	-7.13	-3.20
	Average	-1.87	-0.95	-4.06	-2.09	-4.16	-2.20	-6.53	-3.47
Class E (1280×720)	FourPeople	-1.83	-1.94	-3.42	-3.27	-4.56	-5.18	-6.33	-6.70
	Johnny	-1.29	-1.61	-2.96	-2.86	-3.34	-4.16	-5.21	-5.78
	KristenAndSara	-1.34	-1.75	-3.49	-3.26	-3.82	-4.47	-6.01	-6.17
	Average	-1.49	-1.77	-3.29	-3.13	-3.91	-4.60	-5.85	-6.22
Class F (various resolutions)	BasketDrillText	-2.37	-1.51	-11.01	-6.31	-6.24	-3.71	-21.64	-13.15
	ChinaSpeed	-2.02	-1.22	-2.87	-1.58	-4.25	-2.71	-4.86	-3.03
	SlideEditing	-2.04	-2.18	-1.82	-1.97	-4.80	-5.05	-3.67	-4.06
	SlideShow	-2.53	-2.32	-3.49	-2.96	-5.66	-5.61	-6.03	-5.91
	Average	-2.24	-1.81	-4.80	-3.21	-5.24	-4.27	-9.05	-6.54
All sequences	Overall	-1.66	-1.10	-3.78	-2.36	-4.10	-2.76	-7.10	-4.67

Table 4.3.1 – Y BD-rate (%) for low complexity and high performance MDTC systems

MDTC systems overcome the MDDT systems by a significant amount, even in their low complexity configuration, around 1 point of BD-rate is gained.

There are some sequences that stand out, notably the *BasketballDrill*, which already showed high gains using the MDDT technique (see §3.2.2). This sequence has the particularity of presenting a large amount of directional patterns, which lead to bitrate savings of over 25%. The BD-rate curves for this sequence are presented in figure 4.3.4. For low bitrates, corresponding to QPs of 32 and 37, the bitstream size is almost kept untouched, but the PSNR is substantially improved. For higher bitrates, the improvements are present in both axes. This behaviour has been observed for most of the sequences.

An example of visual improvements are provided in figure 4.3.5. The first sub-figure displays the original block, which has not been coded yet, and the two other sub-figures the result of coding that block using HEVC and the non-separable high performance MDTC, respectively. The comparison is pertinent, as both sequences have the same bitrate at QP 37 (around 3.30 Mbps, see figure 4.3.4). It is easy to see the improvements made by the MDTC system along the diagonal patterns of the image: The lines are cut when coding with HEVC but they remain continuous using MDTC thanks to the non-separability of the used transforms. For the rest of the sequences, since the bitrate savings are much lower, the visual improvements are, in general, not easy to spot. Nevertheless, no specific artefacts have been noticed when using the MDTC technique for either systems.

	4×4: 1+1 8×8: 1+1		4×4: 1+16 8×8: 1+32	
	sep.	non-sep.	sep.	non-sep.
Y BD-rate	-1.66%	-3.78%	-4.10%	-7.10%
Enc. Time	150%	200%	800%	2000%
Dec. Time	103%	110%	105%	120%
ROM	8.20 kB	148.75 kB	236.25 kB	4.51 MB

Table 4.3.2 – Summary of RDOT-based MDTC systems compared to HEVC

4.3.3 Coding complexity

In this section, the complexity of the MDTC systems is analysed. Contrary to MDDT systems, presented in Chapter 3, MDTC systems explore many more coding alternatives, which leads to significant increments in terms of encoding time: up to 20 times the complexity of the reference HEVC encodings for the high performance MDTC system. The separable version presents a much lower complexity, even if it is 8 times higher than the one of HEVC.

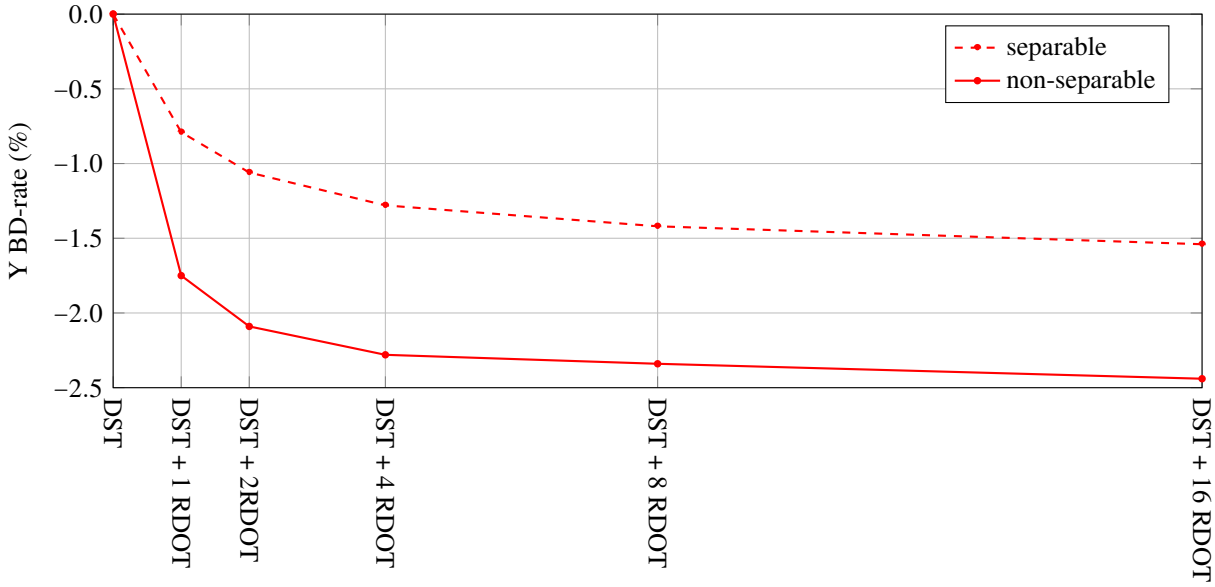
The decoding time is increased due to the lack of efficient fast algorithms for the designed RDOTs, as in the MDDT systems. Nevertheless, in this case, the default HEVC transforms are still available, making it possible that the MDTC decoder becomes less complex that the MDDT in some cases. On the other hand, the decoding complexity can increase slightly with the number of transforms, since it becomes less likely to use the default HEVC transform.

For the same decoding time, the low complexity MDTC system provides higher bitrate savings (around 0.6 BD-rate points) than the MDDT using 4×4 and 8×8 TUs, with the only burden being on the encoding time.

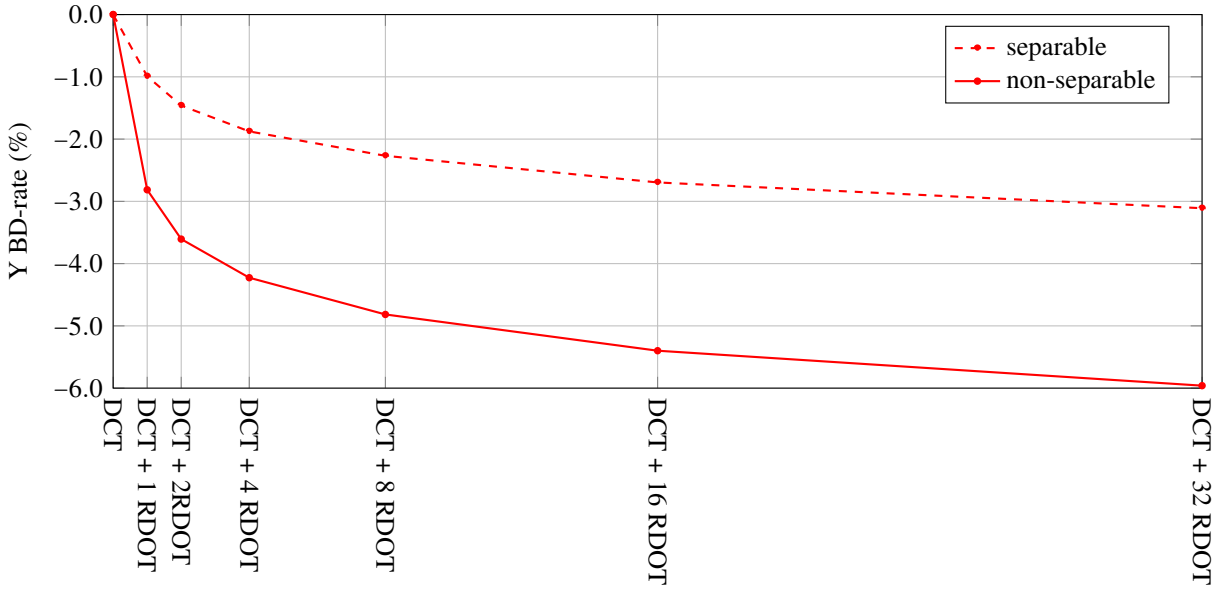
Table 4.3.2 contains the complexity figures for the low complexity and high performance MDTC systems. A graphical version of the table is presented in figure 4.3.6. All axes in the four diagrams have the same scaling factor in order to ease the comparison among them. Using this representation, HEVC would be a dot in the centre of the diagram, and an ideal system would only have one arrow downwards, meaning that the BD-rate has been improved without any additional coding complexity or ROM storage requirements.

4.3.4 Storage requirements

The storage requirements for the MDTC systems are computed as explained in (3.2.1) and (3.2.2), with the exception that, for the MDTC systems, the total ROM is affected by the number of transforms in each TU size. The quantisation of the transform coefficient has remained unchanged: each coefficient is rounded to the closest integer, quantised to 1 byte. The low complexity MDTC system has exactly the same ROM requirements as the MDDT, since only one transform is used per IPM. On the other hand, the high performance MDTC system has storage requirements largely superior: more than 4.5 MB are required for the non-separable version. The actual values of required ROM are presented in table 4.3.2 and in figure 4.3.6.



(a) BD-rate evolution with the number of additional 4×4 transforms



(b) BD-rate evolution with the number additional of 8×8 transforms

Figure 4.3.1 – Average BD-rate evolution for the learning set depending on the number of transforms

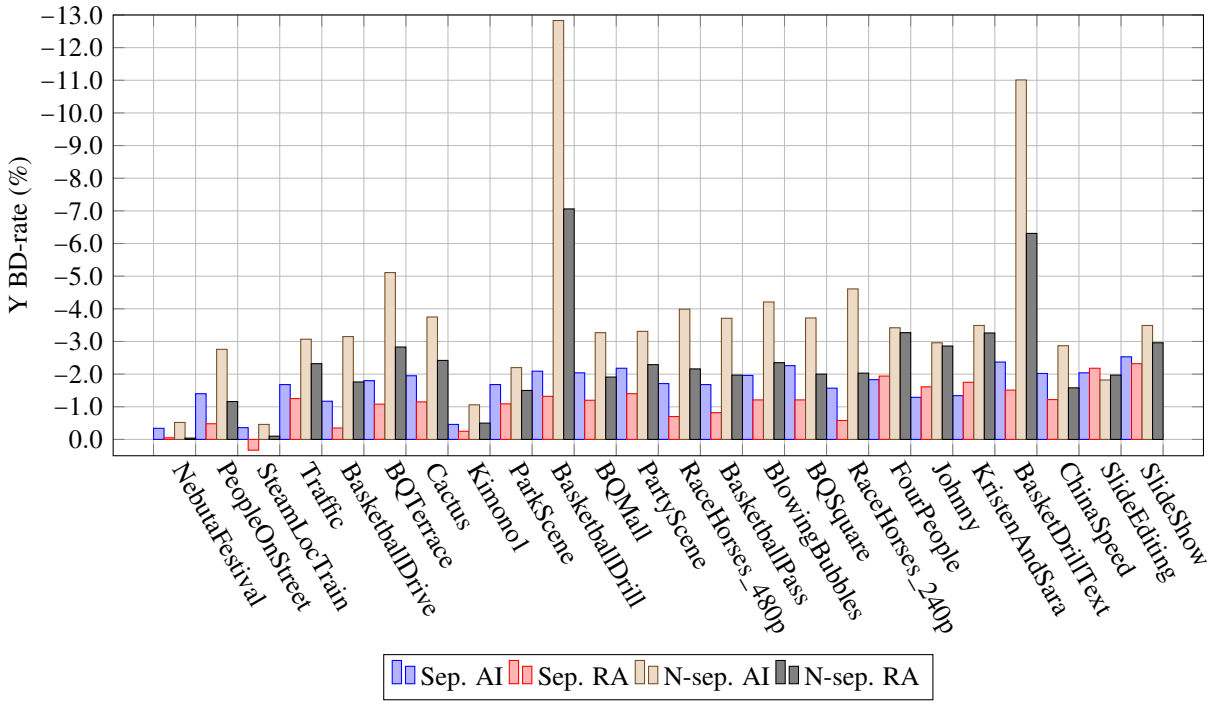


Figure 4.3.2 – BD-rate for low complexity MDTC system

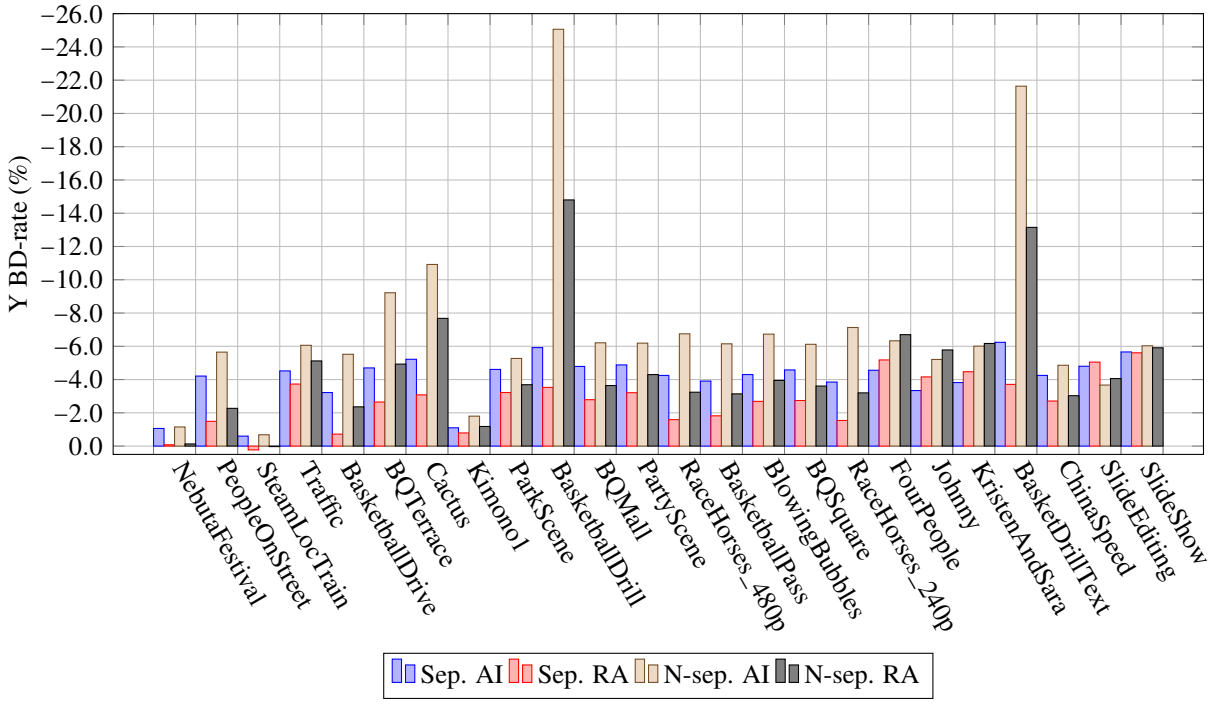


Figure 4.3.3 – BD-rate for high performance MDTC system

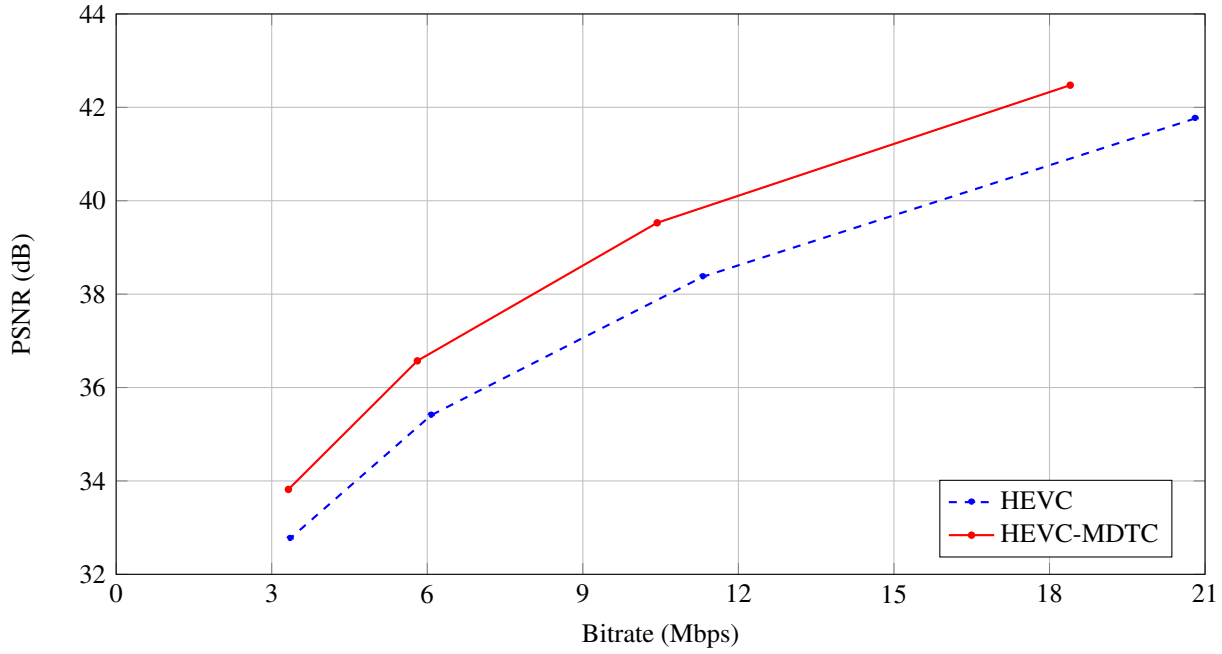
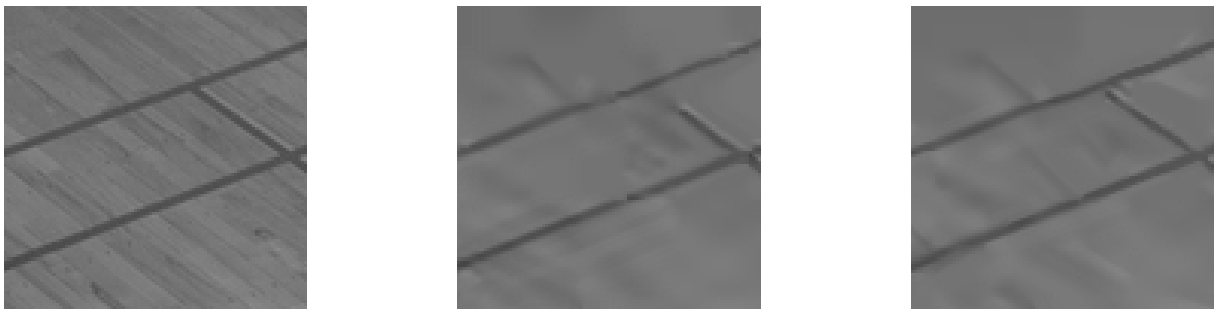
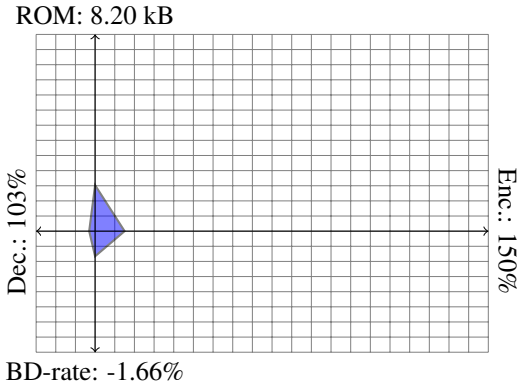


Figure 4.3.4 – BD-rate curves of *BasketballDrill* for the non-separable high performance MDTC system. This curve represents a bitrate improvement of 25.06% over HEVC.

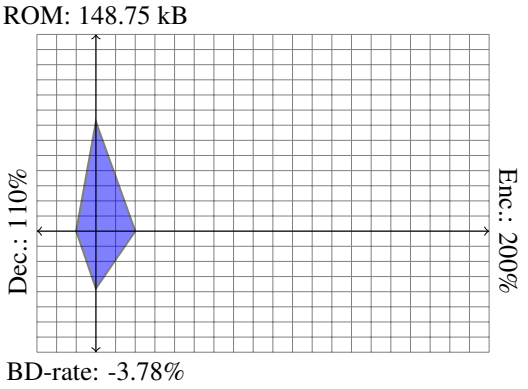


(a) Original crop (b) HEVC at QP 37 (c) MDTC at QP 37

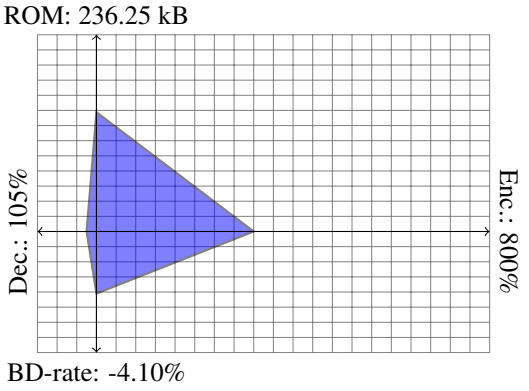
Figure 4.3.5 – A 100×100 block from *BasketballDrill* encoded at QP 37 with HEVC and the non-separable high performance MDTC system



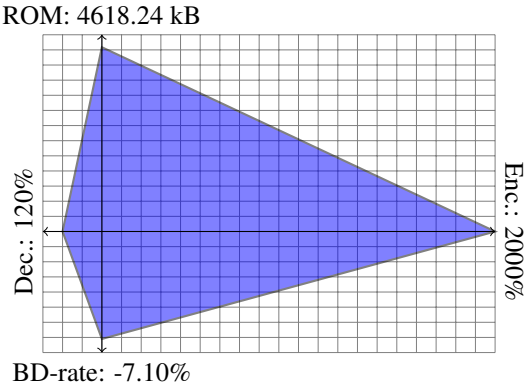
(a) Separable low complexity MDTC



(b) Non-separable low complexity MDTC



(c) Separable high performance MDTC



(d) Non-separable high performance MDTC

Figure 4.3.6 – Graphical comparison of the four proposed MDTC systems

4.4 Conclusions

A systematic procedure based on [47] to learn multiple transforms using the RDOT metric is described in this Chapter. Despite the learning algorithm being subject to some improvements, the potential of using multiple non-separable transforms in video coding is unveiled: average bitrate savings of more than 7% can be achieved in AI configurations, with peaks of up to 25% for some sequences containing large amounts of diagonal patterns. The interest of using multiple separable transforms has also been shown, with bitrate savings of around 4% on average in AI.

However, in order to achieve those gains, the encoder complexity is affected by a factor of 8–20, depending on separability. The decoding time using separable transforms remains reasonable, with only 5% increase, but up to 20% when using non-separable transforms. Storage requirements are also important, if not prohibitive, especially for non-separable transforms.

MDTC systems have also proved to be able to provide different trade-offs in terms of bitrate savings, complexity and storage requirements. Bitrate savings of 3.78% can be achieved with twice the encoding complexity, 10% added decoding complexity and around 300 kB of ROM. In the separable version, savings of 1.66% are obtained at 50% added complexity, 3% added decoding complexity and around 8 kB of ROM.

Next Chapters will explore different ways of simplifying the MDTC systems in three important dimensions of codec design: encoding complexity, decoding complexity and storage requirements while trying to keep most part of the bitrate savings.

Chapter 5

Incomplete transforms

Contents

5.1 Introduction

Chapters 3 and 4 have presented the potential of non-separable transforms in terms of bitrate savings for video coding. However, they come at a cost: the amount of algorithmic operations needed to transform a block is highly increased with respect to a transform having a fast implementation. Besides, the amount of ROM needed to store the transforms themselves is important, sometimes in the order of several MB.

This Chapter presents an attempt at making non-separable transforms usable for video coding applications. The work presented here is mainly based on the one published in [6].

5.2 Motivations of incomplete transforms

5.2.1 Forcing sparse data representation

Sparse data representation has been an important field of study in the last years thanks to its countless applications in many domains, in which, compression and feature extraction stand out. Sparse representation focuses on finding the most compact representation for a given signal [21]. Amongst them, the K-SVD is one way of designing overcomplete dictionaries to achieve sparse data representation [1]. Also, the RDOT systems presented in this thesis are part of this ensemble.

— The RDOT metric contains a rate constraint turned into a measure of the signal sparsity in the transform domain (the number of coefficients set to zero).

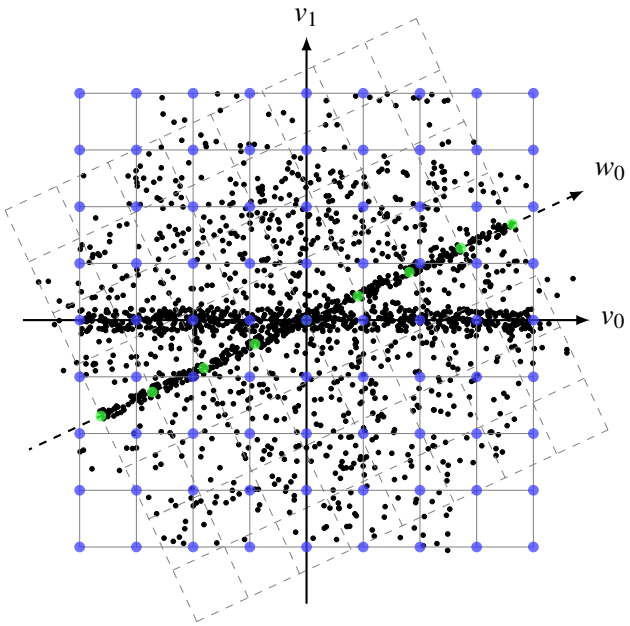


Figure 5.2.1 – Illustration of the incomplete transform concepts. An additional basis vector (w_0 is added to assist an orthogonal transform (v_0, v_1))

— The MDTC system provides an overcomplete representation, as HEVC core transforms are complemented with 2^N transforms. As such, $1+2^N$ sets of coefficients are computed on the encoding side. A selection amongst those is made through the transform signalling and the coefficient coding.

The usage of multiple complementary transforms to provide sparse representations has been addressed in Chapter 4, with the MDTC systems, where high computational requirements were pointed out. In this Chapter, a low complexity solution for sparse data representation is proposed. The approach is based on a standard orthogonal transform, the Discrete Cosine Transform (DCT), in competition with multiple elementary sparse transforms, named *incomplete transforms*. The competition exists in the sense that the encoder selects, for each TU, the transform that provides the best signal representation in the distortion-sparsity plane. The effectiveness of this approach is measured using the RDOT metric, defined in 2.3.1.

Incomplete transforms can be considered as a special case of Rate-Distortion Optimised Transforms (RDOTs), introduced in §2.3, in which only one base vector is retained and considered. Consequently, a signal transformed using an incomplete transform has only one coefficient different from zero in the transform domain. In order to be able to represent any signal within a given distortion, incomplete transforms are conceived to work as companions of a main orthogonal transform, such as the DCT for image coding.

To illustrate a case where incomplete transforms might be useful, figure 5.2.1 presents a 2D scenario, where the small dots symbolise the 2D signals to be transformed. The main transform, whose base vectors are v_0 and v_1, is able to represent the signal efficiently, as v_0 follows the main direction of the dark dots. By construction, v_1 is orthogonal to v_0.

However, there exists a secondary direction that cannot be represented compactly using the (v_0, v_1) base vectors: both axes are required to describe the coordinates of those points. Just by adding an extra axis, w_0, adapted to this secondary direction, an effective and sparser representation of those dots can be achieved.

Therefore, the dots plotted in this space can be efficiently represented thanks to the union of two transforms. One transform is complete, the second one, which can be conceived as a complete transform, is restricted to only one base vector, the principal component: this way, the compactness is guaranteed, as only one transform coefficient needs to be transmitted.

If only one adapted transform had been used in figure 5.2.1 to adapt to all those points, such as the Karhunen-Loève Transform, the main axis would have been placed somewhere between v_0 and w_0.

5.2.2 Complexity analysis

Incomplete transforms are proposed in this work to provide a low complexity approach to the non-separable MDTC systems. A remarkable consequence of using incomplete transforms is a decrease in complexity when decoding a signal, since there will be one coefficient different from zero, decoding implies multiplying the base vector by the transformed coefficient.

In image coding, a separable two-dimensional transform is expressed as in (2.1.4). Assuming the image is composed of 8×8 blocks, $\mathbf{x}$ stands for the 8×8 pixels, and $\mathbf{X}$ their representation in the transform domain. $\mathbf{A}$ is the 8×8 1D transform. The usual transform used in image coding is the DCT-II, whose fast algorithm requires in the order of 12 multiplications and 29 additions per 8×1 vector. As 8 vectors per block need to be processed both for the vertical and horizontal transform, processing an 8×8 block requires a total of 192 multiplications and 464 additions. This number of operations is identical for the inverse transform.

For an incomplete transform, only one axis needs to be processed: each axis being formed by 64 values in this example. Consequently, only 64 multiplications and 63 additions are also needed to transform the input block $\mathbf{x}$ into the transform domain, that is, a simple correlation of the first base vector with the block $\mathbf{x}$. For the inverse transformation, only 64 multiplications are needed, since the first and only transform coefficient multiplies the first basis vector to recover the spatial domain samples.

As a result, in this case, the incomplete transforms can be applied with a number of operations of approximately one third of the cost of the fastest separable transforms. This complexity reduction benefits both the encoder and the decoder.

It is also worth noticing that those incomplete transforms are chosen non-separable and, therefore, able to exploit any linear correlation amongst pixels within a block.

5.3 Design of incomplete transforms

The incomplete transform design is based upon the RDOT model proposed in [47] and detailed in §2.3. The original method describes a way of iteratively deriving one optimal transform for some given training data and an initial transform by using a metric that includes a sparsity constraint. This section adapts the learning method for incomplete transforms.

5.3.1 Incomplete transform learning

The learning phase of incomplete transforms uses the same method presented in §2.3, particularly in (2.3.1). The design of an incomplete transform takes one extra step apart from those required in the RDOT learning: after the hard-thresholding of the coefficients, only the first coefficient is kept. Consequently, the ℓ_0 norm of the transform coefficients is always equal to one for any incomplete transform. Therefore, for a given set of training signals, one obtains a transform consisting of one meaningful base vector. The remaining vectors, albeit constituting a complete transform with the first base vector, are useless for the aim of the application.

For consistency with previous Chapters, the learning set consists of prediction residuals issued from an HEVC AI coding for 4×4 and 8×8 blocks, the same one used in Chapters 3 and 4.

Multiple incomplete transforms

In Chapter 4 the use of multiple transforms inside each IPM was introduced. A decision has been made to keep the default HEVC transform in each TU size for compatibility reasons. Besides, since incomplete transforms are designed with a strong sparsity constraint on the quantised coefficients, a complete transform is needed to guarantee that all signals can be expressed in the transform domain

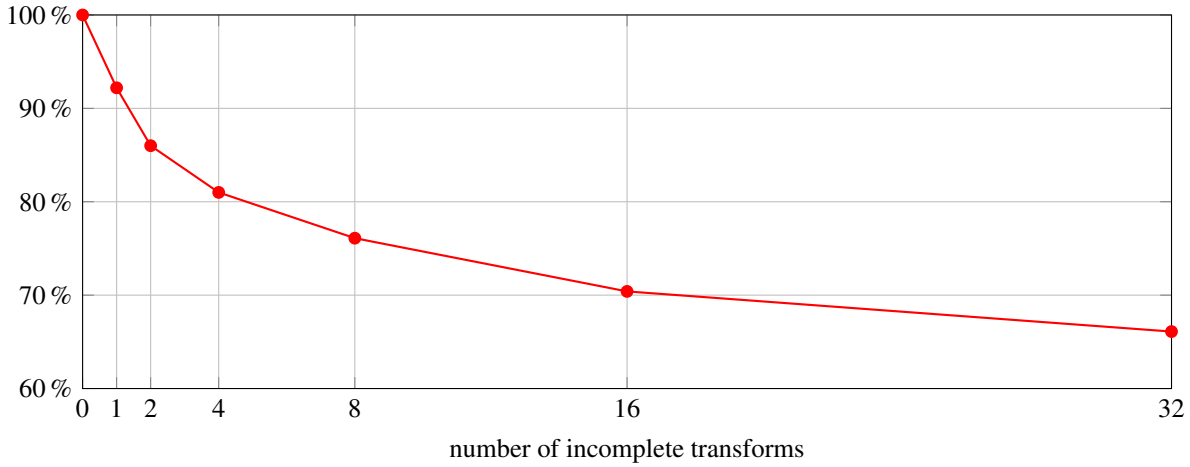


Figure 5.3.1 – Percentage of non-zero coefficients referred to the DCT

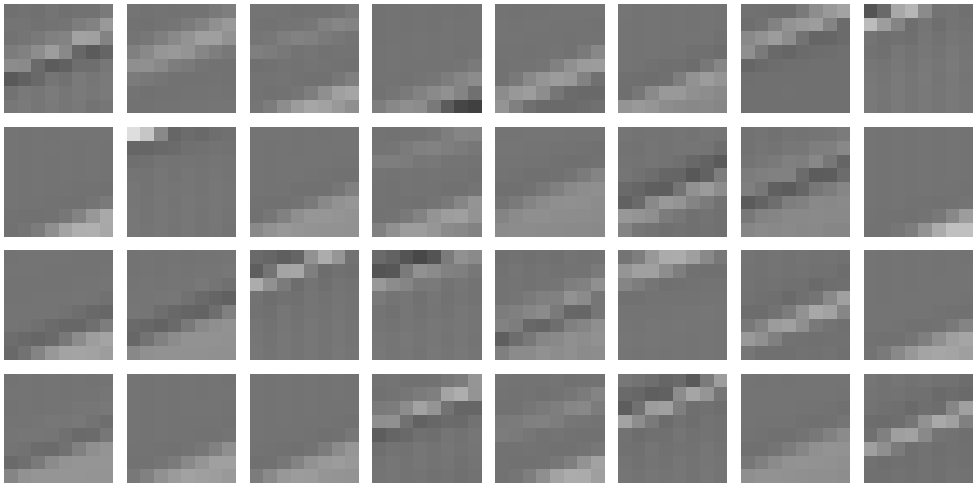


Figure 5.3.2 – List of 32 incomplete transforms for 8×8 blocks and IPM 6

for a chosen level of fidelity. By using the same design as in the previous Chapter (one main transform plus 2^N incomplete transforms), signals would be able to have a fall-back transform when incomplete transforms do not provide the desired level of fidelity in terms of distortion.

As a result, the learning algorithm for non-separable transforms has remained almost untouched from algorithm 1, described in §4.2.2, only the design of the transform has been modified by just keeping the first coefficient after the hard-thresholding. After that, residuals are evaluated using the RDOT metric to assign them to the best transform.

To illustrate the effectiveness of the learning process, figure 5.3.1 presents how an increase on the number of incomplete transforms is able to provide a more sparse representation of the signal in the transform domain. This illustration is based on the IPM 6 for 8×8 blocks.

To evaluate this, the average number of significant coefficients is computed for different coding configurations (from 1 to 32 incomplete transforms used in conjunction with the DCT). The results are presented relative to the reference system, which consists of a coding system using the DCT alone, such as the HEVC. As the number of incomplete transforms is increased, the proportion of significant coefficients is decreased to 66% of its original value for a similar distortion: this validates the fact that more sparse representations can be achieved with the joining of incomplete transforms to the traditional DCT.

The result of a learning experiment is shown in figure 5.3.2, where 32 8×8 incomplete transforms are presented. For each one, only the first base vector is displayed, since it is the only one delivering

significant transformed coefficients. In the case of two-dimensional signals, such as images, incomplete transforms can be directly interpreted as texture patterns against whom the input signal is matched. The learning set in this experiment is made of prediction residuals extracted from a directional mode from the HEVC coding scheme. The selected mode (IPM 6) corresponds to an angular prediction of approximately +26°. Accordingly, the blocks selected by HEVC for this mode mostly present a directional pattern following that prediction. It can be observed how the incomplete transforms have patterns containing that particular direction, each exhibiting a particular band-shaped pattern. Note that the DCT requires a significant number of coefficients to represent such directional, and inherently non-separable, patterns because none of the DCT bases expresses that direction.

5.4 Incomplete transforms in video coding

To evaluate the performance of the approach, a set of 4×4 and 8×8 incomplete transforms is designed for each HEVC IPM. At the encoder, for each block, the best IPM/transform pair is selected, exactly in the same way as the MDTC system from Chapter 4.

5.4.1 Signalling of incomplete transforms in the bitstream

The signalling scheme for incomplete transforms is inherited from the MDTC system: a flag indicates whether the default HEVC transform is used. If that is not the case a fixed length codeword indicates which incomplete transform has been chosen by the encoder for that IPM.

Incomplete transforms have a special feature that generic RDOTs used in the MDTC system do not have: they always output a signal that has only one coefficient different from zero. Consequently, the residual signal bitstream syntax can be reduced: the significance map (i.e. the location of the significant coefficients) does not need to be transmitted, since the position of the last element different from zero is known when using incomplete transforms. This fact can be taken advantage of by removing the `last_sig_coeff_x` and `last_sig_coeff_y` prefix and suffix syntax elements from the conveyed bitstream [55].

5.4.2 Performances of different configurations

Experiments have been performed following the common test conditions described in [11] for the HEVC test set. Incomplete transforms have been enabled for 4×4 and 8×8 TUs in order to make the system comparable to the MDTC systems from the previous section. In particular, 8 additional transforms have been designed for 4×4 TUs and 32 for 8×8 TUs.

The results of encoding the HEVC test set using incomplete transforms are detailed in table 5.4.1. Performances are much lower than those presented in the previous Chapter for the MDTC system. However, as explained in §5.4.3 and §5.4.4, incomplete transforms requirements are also much lower than a MDTC system.

When incomplete transforms are enabled only for 4×4 TUs, the system presents modest performances (around 0.4% of bitrate savings) for both AI and RA coding configurations. The system starts presenting more substantial performances when enabling incomplete transforms for 8×8 TUs, providing 1% of average bitrate savings. If incomplete transforms are enabled for both TU sizes, the average bitrate savings achieve 1.4% for the AI coding configuration and 1% for RA.

Sequences that performed exceptionally well using the MDTC system due to their non-separable nature, such as *BasketballDrill* present bitrate savings of more that 8% using this approach.

Figure 5.4.2 shows the BD-rate curves of standard HEVC and the MDTC system using incomplete transforms on 4×4 and 8×8 on the *BasketballDrill* sequence. The bitrate savings are over 8% and it can be seen that the system behaves virtually in the same way as the MDTC systems presented in the previous Chapter for all bitrate ranges.

		4×4		8×8		4×4 & 8×8	
	Sequence	AI	RA	AI	RA	AI	RA
Class A (2560×1600)	NebutaFestival	0.01	0.00	0.00	-0.04	0.01	-0.03
	PeopleOnStreet	0.00	0.02	-0.64	-0.32	-0.86	-0.37
	SteamLocomotiveTrain	0.04	-0.12	0.03	0.16	0.03	0.19
	Traffic	0.02	-0.24	-0.68	-0.62	-0.88	-1.04
	Average	0.02	-0.08	-0.32	-0.20	-0.43	-0.31
Class B (1920×1080)	BasketballDrive	-0.10	0.02	-1.07	-0.72	-1.31	-0.78
	BQTerrace	-0.36	-0.39	-1.06	-0.76	-1.41	-1.10
	Cactus	-0.18	-0.32	-1.38	-0.87	-1.73	-1.25
	Kimono1	0.07	0.02	-0.31	-0.17	-0.32	-0.23
	ParkScene	0.13	0.08	-0.15	-0.16	-0.19	-0.24
	Average	-0.09	-0.12	-0.80	-0.54	-0.99	-0.72
Class C (832×480)	BasketballDrill	-3.21	-1.95	-6.22	-2.91	-8.22	-4.35
	BQMall	-0.10	-0.19	-0.66	-0.44	-1.03	-0.77
	PartyScene	-0.10	-0.29	-0.18	-0.15	-0.40	-0.54
	RaceHorses	-0.31	-0.23	-0.44	-0.37	-0.85	-0.51
	Average	-0.93	-0.67	-1.88	-0.97	-2.63	-1.54
Class D (416×240)	BasketballPass	-0.18	-0.06	-0.47	-0.31	-0.88	-0.55
	BlowingBubbles	-0.15	-0.37	-0.21	-0.07	-0.52	-0.49
	BQSquare	-0.32	-0.37	-0.25	-0.19	-0.59	-0.51
	RaceHorses	-0.37	-0.09	-0.32	-0.12	-0.86	-0.37
	Average	-0.26	-0.22	-0.31	-0.17	-0.71	-0.48
Class E (1280×720)	FourPeople	-0.03	-0.46	-1.07	-1.03	-1.36	-1.58
	Johnny	-0.13	-0.75	-1.25	-1.16	-1.56	-1.84
	KristenAndSara	-0.31	-0.59	-1.15	-0.77	-1.56	-1.44
	Average	-0.16	-0.60	-1.16	-0.99	-1.49	-1.62
Class F (various resolutions)	BasketDrillText	-2.54	-1.67	-4.83	-2.33	-6.49	-3.66
	ChinaSpeed	-0.40	-0.29	-0.45	-0.26	-0.90	-0.61
	SlideEditing	-0.23	-0.24	-0.51	-0.61	-0.71	-0.81
	SlideShow	-0.73	-0.76	-0.81	-0.83	-1.51	-1.64
	Average	-0.97	-0.74	-1.65	-1.01	-2.40	-1.68
All sequences	Overall	-0.40	-0.38	-1.00	-0.63	-1.42	-1.02

Table 5.4.1 – Y BD-rate (%) for MDTC systems using incomplete transforms

5.4.3 Coding complexity

Despite using non-separable incomplete transforms, which are notably less complex than regular non-separable and even separable transforms, the encoding complexity is increased. This is due to the fact that incomplete transforms are used as a simplified MDTC system, where many transforms are tested for a same block size. This fact makes the encoder explore many coding possibilities that are unavailable to HEVC. That being said, the increase in encoding complexity is lower than for a full-fledged MDTC system, since testing of incomplete transforms is much faster.

On the decoder side, the situation changes in favour of the incomplete transforms with regards to HEVC. Since all coding choices are made by the encoder and the decoder only has to apply the signalled transform. Therefore, the decoder is of equal complexity, whenever the DCT/DST is chosen, or lower, if incomplete transforms are used, than that of HEVC.

The actual results for encoding and decoding complexity are contained in table 5.4.2.

5.4.4 Storage requirements

Due to the similarities to the MDTC system, the non-separable incomplete transforms also need to be stored. However, in this case, the storage requirements are divided by N^2, with N being the TU size

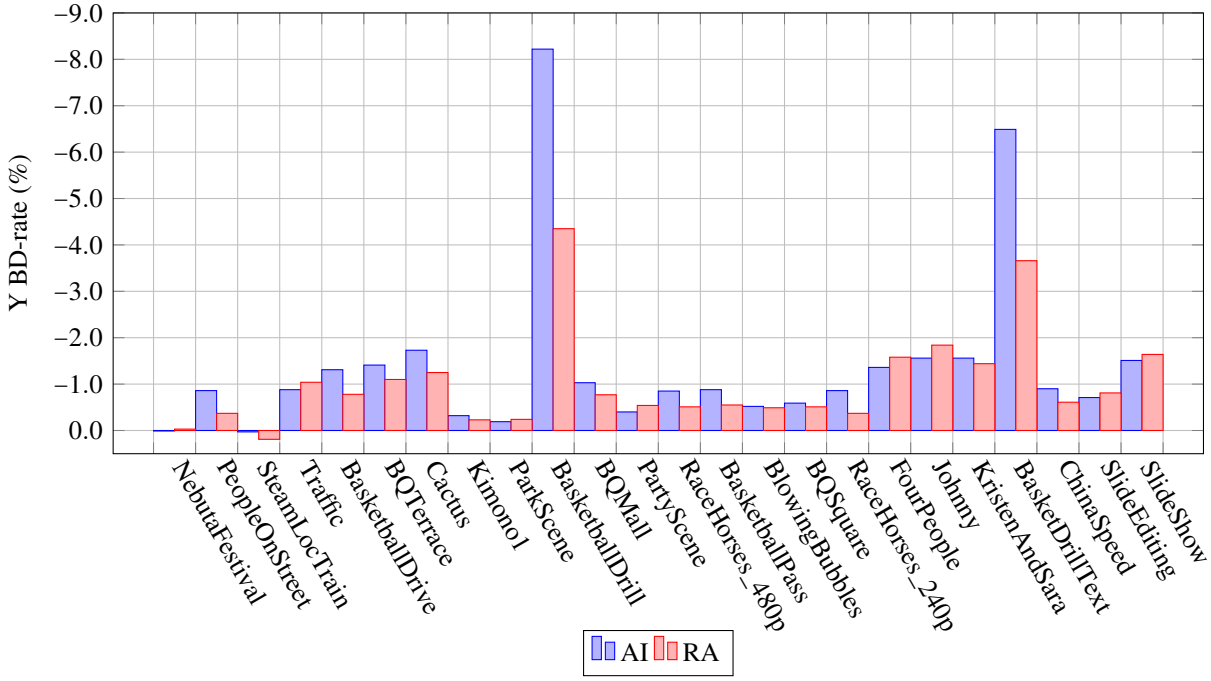


Figure 5.4.1 – BD-rate for incomplete transforms enabled for 4×4 and 8×8 TUs

	4×4	8×8	4×4 & 8×8
Encoding	210%	250%	340%
Decoding	97%	99%	100%

Table 5.4.2 – Relative average complexity of incomplete transform systems to HEVC

for which the transforms are conceived, since only one base vector is stored. Assuming each transform coefficient is quantised to 1 byte and rounded to the nearest integer, as in the MDDT and MDTC systems, this leads to storage requirements of:

— 4×4 transforms: 16 B per transform.

— 8×8 transforms: 64 B per transform.

For the system using 8 4×4 transforms and 32 8×8 transforms, the global storage requirements are 74.38 kB, as there are 35 IPMs.

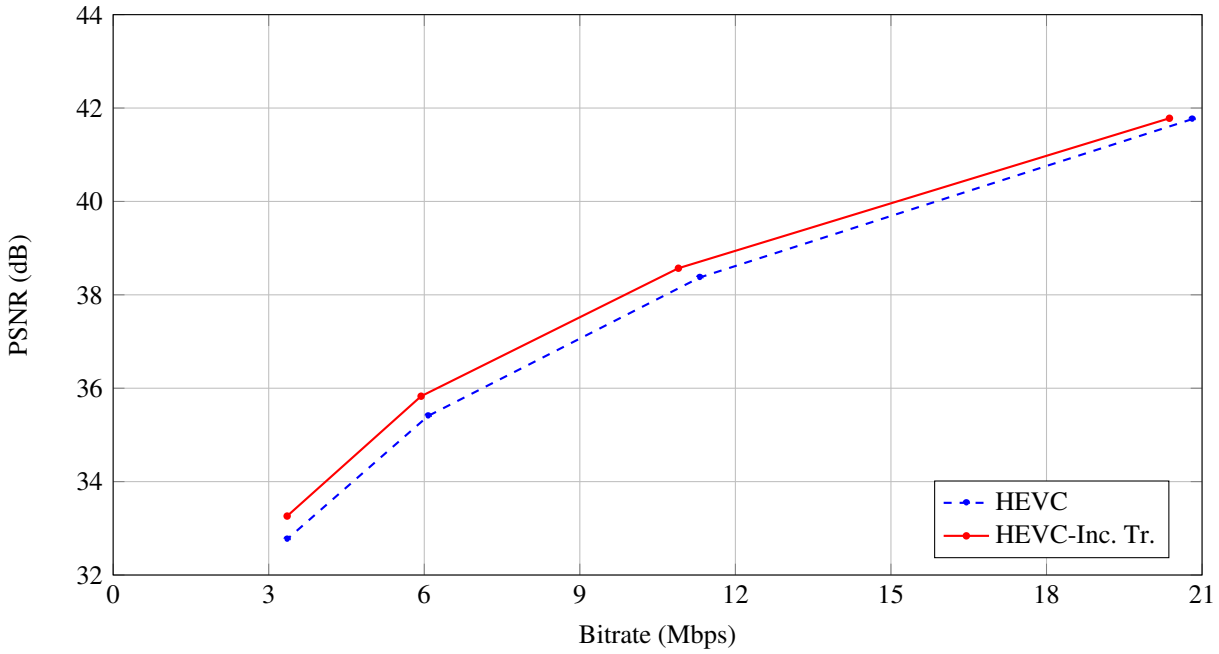


Figure 5.4.2 – BD-rate curves of *BasketballDrill* for the MDTC system using incomplete transforms. This curve represents a bitrate improvement of 8.22% over HEVC.

5.5 Conclusions

Incomplete transforms provide a simplified version of the MDTC system, presented in Chapter 4. These complementary transforms are able to increment the sparsity of the signal in the transform domain: the number of non-zero coefficients can be reduced to around two thirds of its original value.

Experimental results on HEVC prove that the sparsity improvements can lead to bitrate savings of 1.4% with peaks of up to 8%, while keeping the decoding complexity lower than that of HEVC. As a result, incomplete transforms constitute an example of application for non-separable transforms that can lead to bitrate savings for reasonable encoding complexity and storage amounts. On the decoder side, incomplete transforms require fewer operations per block than HEVC transforms, leading to a decrease on the decoding complexity.

Summing up, MDTC systems using incomplete transforms represent a more lightweight alternative to RDOT-based MDTC systems presented in Chapter 4 that are able to provide improvements over HEVC across a wide range of bitrate with low decoding complexity.

Chapter 6

Simplifications of MDTC systems for storage requirements

Contents

6.1 Introduction

Chapter 4 introduced the Mode-Dependent Transform Competition (MDTC) system for video coding schemes. Significant bitrate savings were obtained with regards to HEVC, especially when using the non-separable high-performance system, where many non-separable transforms were provided in each Intra Prediction Mode (IPM): up to 7% on average for the AI coding configuration.

This Chapter takes a look at different approaches to simplify the system presented in Chapter 4 in terms of storage requirements while trying to have minimum impact on its bitrate savings.

6.1.1 Discarding non-separable transforms

In order to conceive a low complexity system that makes use of transform competition, some concessions are required. This Chapter proposes simplifications to the areas leading to significant complexity increases in MDTC systems, especially, to the storage requirements.

The first approach for simplification is to suppress the use of non-separable transforms, even though they provide notably better performances than their separable counterparts, as seen in Chapters 3 and 4. Non-separable transforms can definitively not accommodate a system with neither storage restrictions, nor coding complexity limitations. For this reason, this Chapter will only consider systems composed by separable transforms. As a consequence of ditching non-separable transforms, the average bitrate savings

achievable on the HEVC using separable high performance MDTC systems are around 4%, which will be used as a reference when comparing the simplified systems.

Furthermore, separable transform present an increase of around 5% in decoding complexity, instead of the 20% added by non-separable transforms.

6.2 Source of the learning set

6.2.1 Completely independent sets

A criticism that could be made to systems presented in Chapters 3, 4 and 5 is that the testing set includes the learning set: classes B and C have served as the learning set and the resulting transforms to build the MDDT and MDTC systems have been tested against those classes. That being said, performances on classes not included in the learning set (A, D, E and F) have similar levels of performances, as described previously on their respective system tables (see tables 3.2.3, 4.3.1 and 5.4.1).

To remove any doubt about the effect of the learning data set, a completely independent data set has been chosen and used to learn the transforms. In order to make all results available and reproducible, the sequence of choice from which residuals have been output is *Tears of Steel*, a Creative Commons short film [10]. The film features computed generated imagery and at a resolution of 1920×800 at a frame rate of 24 Hz and a length of 12 min and 14 s. Although it is composed of more that 130 sub-sequences, all of them share the same video post-processing, frame rate, resolution, making a source of residuals less heterogeneous than classes B and C from the HEVC test set, but providing a larger number of residuals (around as twice as many, in total):

— 4×4 residuals: over 142 million.

— 8×8 residuals: over 343 million.

The resulting transforms have served to build new MDTC systems that have been compared to the previous ones on different sequence sets.

Additionally, another independent set of sequences has been used to test the performances of both learning sets (classes B and C, and *Tears of Steel*). This set consists of 59 sequences at different resolutions (1920×1080, 1280×720, 832×480) and various frame rates (25, 30, 50, 60, 100, 120, 240 frames per second). Each sequence has been limited to 5 frames and all the tests are performed in the AI configuration.

From now on, the different sets will be referred to as:

— BC: classes B and C from the HEVC test set, used as the learning set in the previous Chapters.

— ADEF: classes A, D, E and F from the HEVC test set.

— 59seq: 5 frames of 59 heterogeneous (private) sequences for testing.

— ToS: the *Tears of Steel* publicly available short film.

6.2.2 Impact on the coding performances over the learning sets

This section presents the results of different MDTC systems learnt with BC and ToS, and then tested against BC, ADEF, 59seq and ToS.

Various additional transform configurations have been learnt for 4×4 and 8×8 TUs. The resulting performances are contained in tables 6.2.1 and 6.2.2 for 4×4 and 8×8 separable transforms, respectively. The table headers read as follows: Test set | Learning set. The value before the pipe sign corresponds to the set of sequences that have been tested and the value after the pipe to the learning set that has been used. Moreover, since the ToS learning set provides a larger number of residuals, more transforms have been learnt for the 8×8 blocks. Bitrate savings are present for all transform configurations from both residuals sources on all test sets, and they coherently increase with the number of transforms.

To better understand the impact of the learning set on the different test sets, figure 6.2.1 shows the results from tables 6.2.1 and 6.2.2. For the 4×4 case, there are few differences in performances between

# tr.	BC | BC	BC | ToS	ADEF | BC	ADEF | ToS	59seq | BC	59seq | ToS	ToS | BC	ToS | ToS
1	-0.69	-0.61	-0.85	-0.95	-0.48	-0.48	-0.13	-0.17
2	-0.93	-0.85	-1.16	-1.21	-0.66	-0.66	-0.21	-0.25
4	-1.08	-1.05	-1.32	-1.54	-0.76	-0.80	-0.23	-0.31
8	-1.23	-1.14	-1.48	-1.68	-0.86	-0.91	-0.27	-0.34
16	-1.28	-1.22	-1.65	-1.85	-0.93	-0.96	-0.30	-0.34
32	-1.28	-1.25	-1.76	-1.89	-0.93	-0.99	-0.27	-0.35

Table 6.2.1 – BD-rate (%) for different testing and learning sets for 4×4 blocks

# tr.	BC | BC	BC | ToS	ADEF | BC	ADEF | ToS	59seq | BC	59seq | ToS	ToS | BC	ToS | ToS
1	-0.97	-0.68	-0.82	-0.74	-0.93	-0.83	-0.75	-1.00
2	-1.43	-1.11	-1.20	-1.13	-1.38	-1.32	-1.01	-1.33
4	-1.91	-1.44	-1.55	-1.47	-1.84	-1.70	-1.30	-1.59
8	-2.35	-1.81	-1.91	-1.83	-2.15	-2.08	-1.48	-1.84
16	-2.81	-2.19	-2.27	-2.12	-2.50	-2.46	-1.69	-2.05
32	-3.32	-2.51	-2.63	-2.43	-2.84	-2.76	-1.84	-2.17
64	-3.88	-2.79	-3.01	-2.66	-3.10	-3.04	-1.93	-2.28
128	—	-3.07	—	-2.92	—	-3.30	—	-2.38

Table 6.2.2 – BD-rate (%) for different testing and learning sets for 8×8 blocks

both learning sets, BC and ToS. Only classes ADEF (in blue) present some preference for the Tears of Steel data set. For the 8×8 case, there seems to be some over-learning for the sequences used during the learning phase (BC and ToS), since they present notable differences in performances when using their own learning set. The rest of test sets present similar performance levels for both learning sets.

This comes as a reassuring fact, since the residuals from ToS have less variability in terms of camera filters, frame rate, resolution and there are some computer generated textures, whereas the rest of sequences are exclusively composed of natural images issued from different sources, frame rates and resolutions.

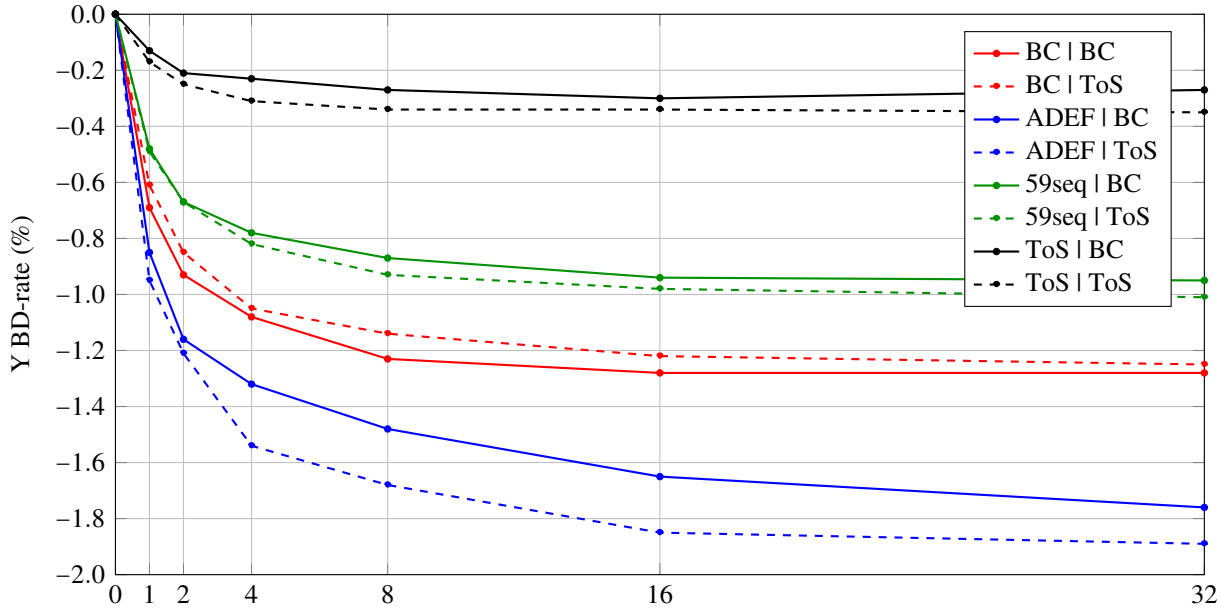
Table 6.2.3 shows the impact of changing the learning set to design the high performance MDTC system: losses of 0.45 % points are observed for the AI coding configuration for the HEVC test set, leading to bitrate savings of 3.65% instead of 4.1%.

Finally, as the 59seq set will be serving as the new reference system, against which new improvements will be measured, the separable high performance MDTC system has been evaluated on this test set. The results are also presented in table 6.2.3. Performances of the high performance MDTC system are almost equivalent on the 59seq testing set, independently of the used learning set.

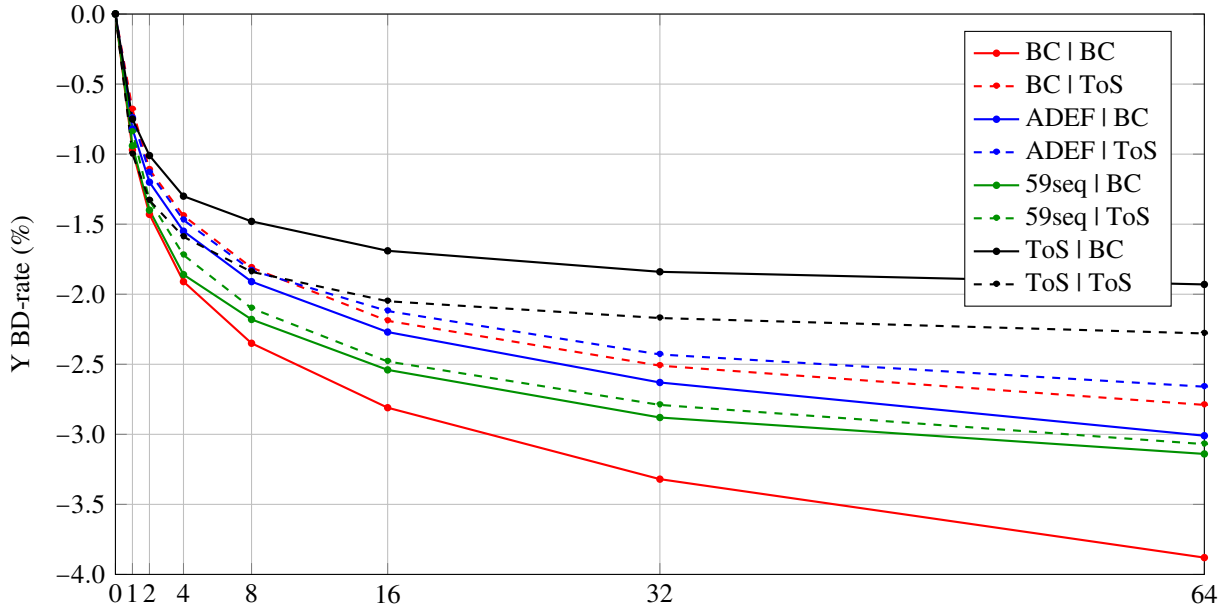
In this section, it has been proved that the transforms learnt and tested in previous Chapters were subject to some over-learning, leading to slightly better performances than when using a learning set completely independent from the test set. This over-learning reassures the fact that the designed transforms are able to adapt to the learning set, while still showing that there is little dependence to it. As a matter of fact, it can be concluded that changing the learning set does not affect the overall performance: the order of improvement is learning set independent although some small local differences appear.

Learn \ Test	HEVC test set AI	HEVC test set RA	59seq AI
BC	-4.10%	-2.76%	-3.54%
ToS	-3.65%	-2.62%	-3.36%

Table 6.2.3 – Bitrate savings for separable high performance MDTC using different test sets. This system uses 16 additional 4×4 transforms and 32 additional 8×8 transforms.



(a) Additional 4 × 4 transforms



(b) Additional 8 × 8 transforms

Figure 6.2.1 – BD-rate for different testing and learning sets

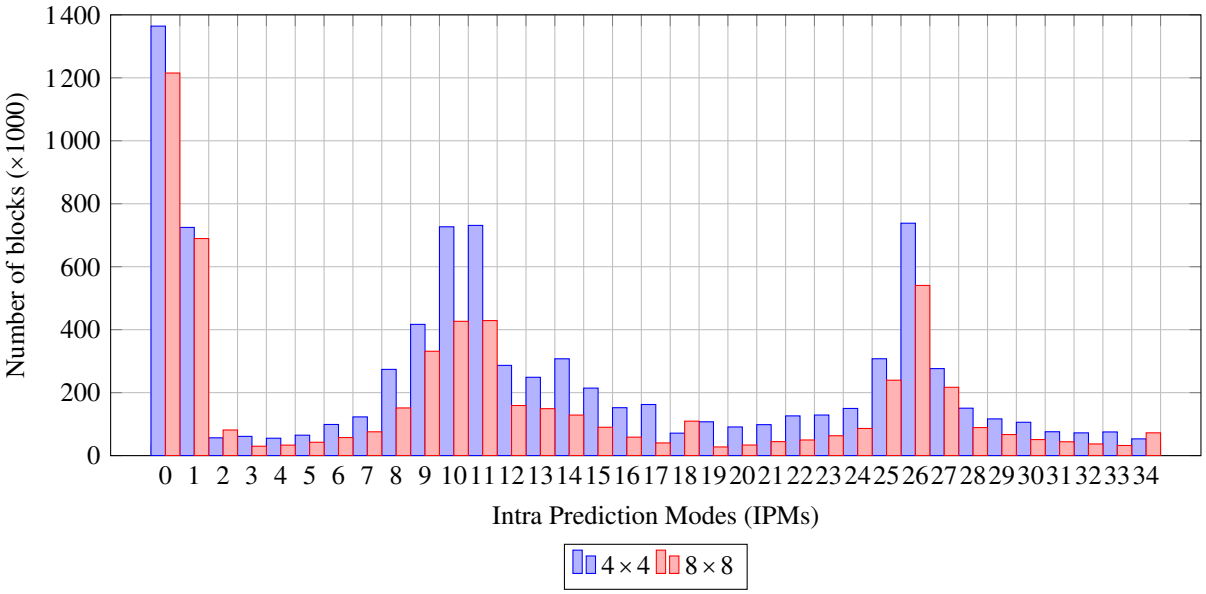


Figure 6.3.1 – IPM usage in HEVC on *BasketballDrive* for QP 22

6.3 Non-homogeneous MDTC systems

In Chapter 4, the MDTC system was introduced. The MDTC system was conceived as an evolved MDDT system, in which, instead of only one transform, a fixed number of transforms was provided inside each IPM. However, as exemplified in figure 6.3.1, HEVC does not use all IPMs equally: HEVC favours the signalling of the IPMs that are most often used during the encoding of a sequence in terms of signalling. This technique is named Most Probable Modes (MPMs) and it is explained in detail in [56]. As a result, it might be counter productive to add a high number of transforms to those modes that are hardly chosen, making them even less cost efficient, since the extra transforms need to be signalled, as well. Using a different number of transforms inside each IPM does not require any additional signalling to the transform index, since the encoder will know how many transforms are available for each IPM.

Consequently, next Section proposes a systematic way of generating MDTC systems taking into account the storage requirements. Since the ROM required to store the transforms is directly related to the number of transforms, and the number of transforms to the encoding complexity, by limiting the amount of ROM, the system might also simplified in terms of encoding time.

This section proposes a gradient descent algorithm to generate MDTC with ROM constraints and a non-homogeneous number of transforms in its IPMs.

6.3.1 Preparation phase

In order to build a system with a different number of transforms in each IPM, additional transforms for 4×4 and 8×8 TUs have been learnt. For the 4×4 TU, up to 32 additional transforms are available, and up to 128 for 8×8 TU.

As a reference point, tables 6.3.1 and 6.3.2 contain different separable homogeneous MDTC systems using 4×4 and 8×8 transforms, respectively. Systems have been labelled using the following convention: 4tN–8tM. Where 4 and 8 stand for the transform sizes, t for the type, in this case s, meaning separable transforms, and N and M represent the number of additional transforms added in each IPM for the respective transform size. The tables have information for each system about its ROM requirements, the bitrate savings and the encoding complexity with regards to HEVC. Consistent bitrate savings are observed when increasing the number of transforms for both TU sizes, even if the 4×4 transform performances seem to saturate with 16 and 32 additional transforms.

System	ROM	Y BD-rate	Complexity
4s1–8s0	1.64 kB	-0.49%	127%
4s2–8s0	3.28 kB	-0.67%	141%
4s4–8s0	6.56 kB	-0.82%	167%
4s8–8s0	13.13 kB	-0.93%	219%
4s16–8s0	26.25 kB	-0.98%	322%
4s32–8s0	52.50 kB	-1.01%	527%

Table 6.3.1 – Summary of 4×4 homogeneous MDTC systems on 59seq | ToS

System	ROM	Y BD-rate	Complexity
4s0–8s1	6.56 kB	-0.84%	127%
4s0–8s2	13.13 kB	-1.33%	142%
4s0–8s4	26.25 kB	-1.72%	171%
4s0–8s8	52.50 kB	-2.10%	229%
4s0–8s16	105.00 kB	-2.48%	345%
4s0–8s32	210.00 kB	-2.79%	577%
4s0–8s64	420.00 kB	-3.07%	1041%
4s0–8s128	840.00 kB	-3.33%	1970%

Table 6.3.2 – Summary of 8×8 homogeneous MDTC systems on 59seq | ToS

6.3.2 Determination of number of transforms per IPM

In order to improve current homogeneous MDTC systems in the ROM — Y BD-rate plane, algorithm 2 presents a systematic way of designing non-homogeneous MDTC systems, using a gradient approach. The algorithm starts with an unmodified version of HEVC. Then, it encodes the 59seq test set enabling one transform by IPM at a time. After that, it chooses the IPM providing the best BD-rate/ROM trade-off, computed as the quotient between the differences in BD-rate and ROM from the configuration chosen in the previous iteration. That configuration is then used as the new base system and the transforms are doubled each time for the IPM that gives the best ratio.

Figure 6.3.2 displays different MDTC systems on the ROM — Y BD-rate plane. The main systems that are compared are the homogeneous and non-homogeneous (issued from the iterative algorithm) systems using 4×4 and 8×8 transforms. Each point in this figure requires encoding 59 sequences at 4 QPs for each one of the 35 IPMs, leading to a total of 8260 encodings.

Systems that combine both transform sizes are also presented in the figure. Iterative non-homogeneous MDTC systems have more flexibility in terms of the number of transforms enabled in each IPM and, therefore, in their storage requirements. For this reason, iterative systems provide a lot more granularity than homogeneous ones, allowing for better ROM–BD-rate trade-offs and more selectable ROM operating points.

Red lines in figure 6.3.2 represent the 4×4 only MDTC systems. Homogeneous 4×4 systems start saturating in terms of bitrate savings after 8 additional transform in each IPM. The non-homogeneous 4×4 system is able to achieve the same performances as the homogeneous system, with less ROM.

The interest of the non-homogeneous systems is clearer when using 8×8 transforms (blue lines), since the storage requirements are higher. In this case, a factor of almost two can be observed in terms of ROM for the same BD-rate in homogeneous and non-homogeneous systems. Moreover, it can be noticed that, for the 8×8 systems, the bitrate savings are proportional, in a logarithmic scale, to the ROM requirements and, consequently, to the number of transforms.

Finally, some interesting configurations have been built combining 4×4 and 8×8 systems, in both homogeneous and non-homogeneous ways, represented using green lines. Again, for these systems, especially on low ROM configurations, a factor of two in terms of storage requirements is observed.

Each one of the points from the learnt systems has two coordinates: ROM and BD-rate. This means

```
testSet      : 59seq
nTransf[35]: 0

tmpTransf[35]: 0

/* Initial encoding of the test set using HEVC */

encode (nTransf, testSet)

while 1 do
    /* increment the transforms per IPM */

    for i = 0 to 34 do
        if nTransf[i] == 0 then
            tmpTransf[i] = 1
        else
            tmpTransf[i] = 2 * nTransf[i]
        end
    end

    /* encode the test set using the 35 configurations */

    results = encode (tmpTransf, testSet)

    /* find the IPM providing the best BD-rate/ROM ratio */

    bestIPM = computeBestRatio (results)

    /* update the transform configuration */

    nTransf[bestIPM] = tmpTransf[bestIPM]
end
```

Algorithm 2: Iterative non-homogeneous MDTC design

that, for a given ROM constraint, one can find the combination of the 4×4 and 8×8 systems that offers the best trade-off in terms of storage requirements and bitrate savings. Consequently, the optimal combinations of non-homogeneous MDTC systems have been found by setting a ROM specification and selecting the combination of heterogeneous 4×4 and 8×8 systems that provide the best BD-rate without exceeding the specified ROM. The results of all system combinations can be found in the scatter plot of figure 6.3.3. It can bee seen that the points present a lower boundary, which corresponds to the best combinations for a given ROM.

The selected ROM requirements for the systems presented in figure 6.3.2 are detailed in table 6.3.3. Information about their actual ROM requirements, BD-rate and encoding complexity are provided. There are some remarkable points, such as the system at 8 kB, which can achieve bitrate savings of 1.8% with an encoding complexity slightly higher than twice of HEVC's. Another interesting point is the 64 kB system, being able to provide bitrate savings of 3% at roughly seven times the complexity of HEVC.

Nonetheless, learning the 4×4 and 8×8 non-homogeneous MDTC systems independently and then combining them to design the complete system is a suboptimal approach. The suboptimality can be spotted by comparing figures 6.3.2 and 6.3.3: the lower boundary of figure 6.3.3 is lower than the actual measurements from figure 6.3.2. This is due to the fact that, when combining both transform sizes, there is an overlap between them in the coded blocks, making the performances of the combined 4×4 and 8×8 systems to be a bit lower than the sum of the corresponding separate systems.

Moreover, presented iterated MDTC systems have only the choice to use 1 to 32 additional transforms for 4×4 TU and 1 to 128 for 8×8 TUs. This is another source of suboptimality once the maximum number of transforms has been enabled in one IPM, since the system lacks the choice to double the number of transforms for that IPM.

As a final observation, non-homogeneous systems learnt iteratively with ROM constraints present a pattern in the number of transforms enabled in each IPM. Those IPMs that are most used in HEVC (0, 1, 10 and 26) tend to make use of a higher number of transforms than the rest of them, which matches closely the IPM use in HEVC from figure 6.3.1: systems tend to use more transforms on IPMs 0, 1, 10, 26 and around them. A detailed table breaking down each MDTC system listed in table 6.3.3 is presented

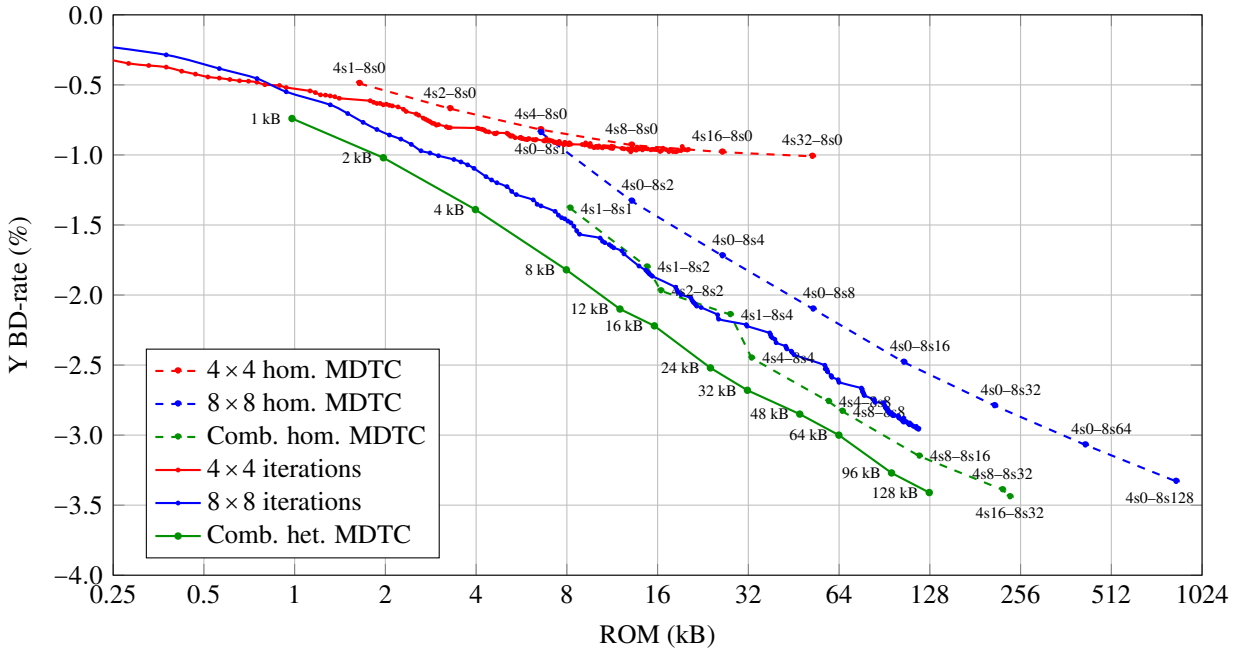


Figure 6.3.2 – Different MDTC systems on the ROM — Y BD-rate plane

System	ROM	Y BD-rate	Complexity
1 kB	0.98 kB	-0.74%	122%
2 kB	1.97 kB	-1.02%	131%
4 kB	3.98 kB	-1.39%	157%
8 kB	7.97 kB	-1.82%	200%
12 kB	12.00 kB	-2.10%	221%
16 kB	15.61 kB	-2.22%	245%
24 kB	23.95 kB	-2.52%	345%
32 kB	31.78 kB	-2.68%	387%
48 kB	47.39 kB	-2.85%	520%
64 kB	63.89 kB	-3.00%	660%
96 kB	95.53 kB	-3.27%	851%
128 kB	127.41 kB	-3.41%	931%

Table 6.3.3 – Non-homogeneous MDTC systems with ROM constraints

in appendix B, particularly tables B.1 and B.2, for 4×4 and 8×8 TUs, respectively.

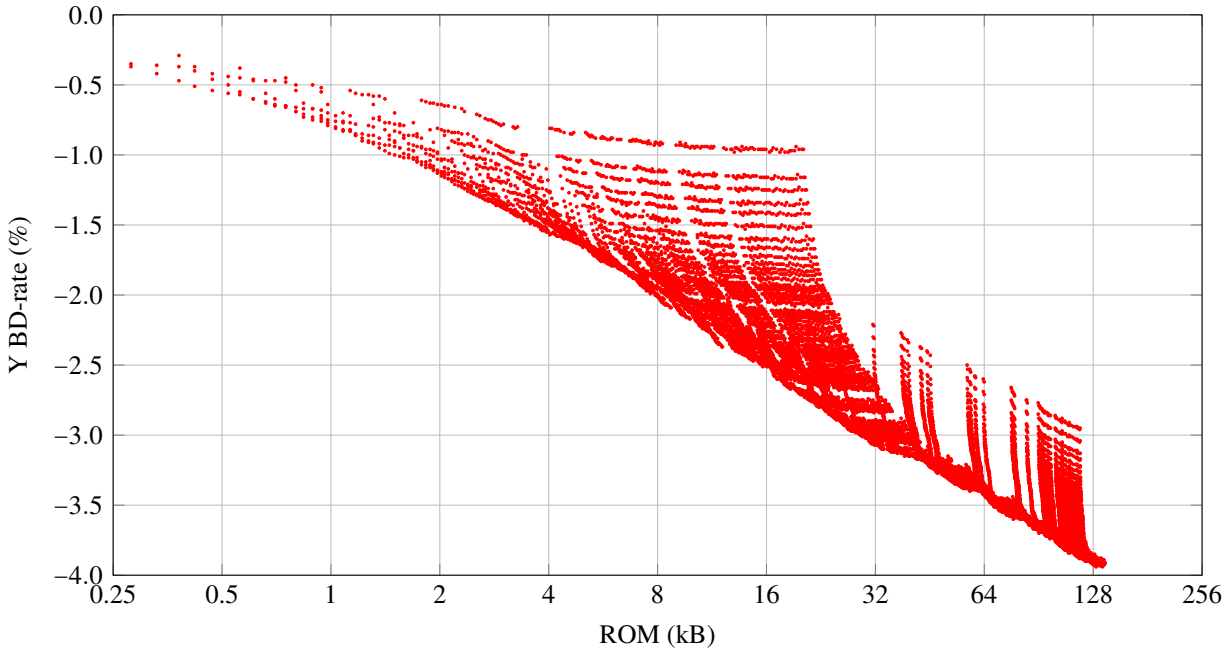


Figure 6.3.3 – All possible combinations of 4×4 and 8×8 non-homogeneous systems based on independent selection of the appropriate number of 4×4 and 8×8 transforms. The lower boundary shows the hypothetical bitrate savings that could be obtained if there were no penalty when combining independently designed 4×4 and 8×8 transforms.

6.4 Symmetrical MDTC systems

6.4.1 Simplifications taking advantage of IPM symmetries

In order to reduce the ROM impact of the transforms, one can take advantage of geometrical symmetries existing amongst HEVC IPMs: within the 35 IPMs in HEVC, symmetries in prediction residuals can be observed for directional modes (IPMs from 2 to 34). Figure 6.4.1 contains a schematic version of the HEVC IPMs that illustrates the proposed symmetrical relations.

It can be stated that residuals issued from the first half (2–17) are closely related to the transposed version of the second half (19–34). Figure 6.4.2 illustrates the average 4×4 residuals profile and how the first half of the directional IPMs relates to the second one.

These symmetries can be taken one step further and be applied inside the first half. Although it might seem counter-intuitive at first, this decision can be justified geometrically: around IPM 10, which is purely horizontal, it seems natural to think that residuals from IPM 9 (slightly diagonal up) and IPM 11 (slightly diagonal down) can be related via a horizontal mirroring operation (top-bottom) or reflection. Since the second half is related to the first half transposed, this property applies around IPM 26, through a vertical mirroring (left-right). Figures 6.4.3 and 6.4.4 contain illustrative examples of average profiles and they relations using these imposed symmetries.

When both symmetries are exploited, the number of transforms of an MDTC system is no longer affected by a factor of 35 (the number of IPMs), but by a factor of 11 (the number of basic IPMs), as presented in table 6.4.1.

Using these symmetries between IPMs implies manipulating the residual before the transform stage.

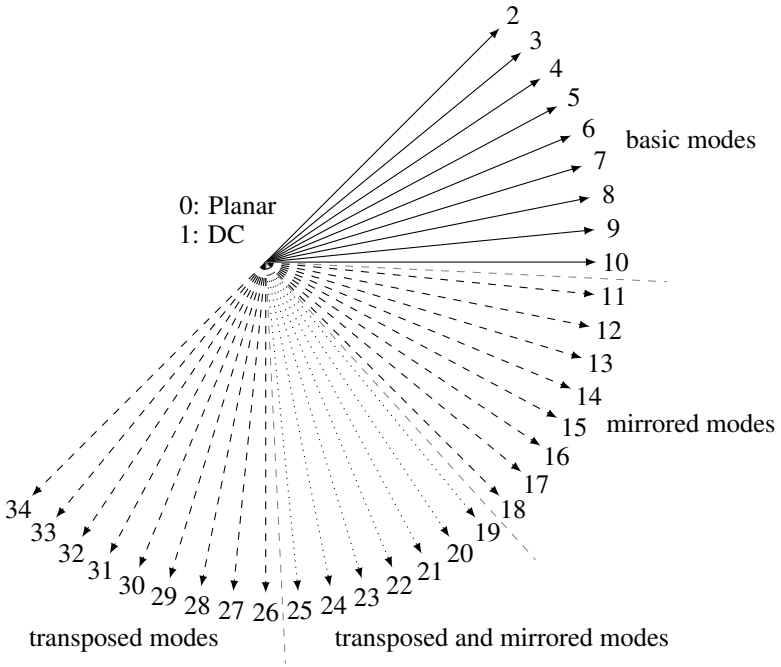


Figure 6.4.1 – The 35 IPMs in HEVC and their symmetries

IPM	$(\cdot)^T$	$\zeta(\cdot)$	$\zeta(\cdot)^T$
0	—	—	—
1	—	—	—
2	34	18	—
3	33	17	19
4	32	16	20
5	31	15	21
6	30	14	22
7	29	13	23
8	28	12	24
9	27	11	25
10	26	—	—

Table 6.4.1 – Symmetrical relations among IPM residuals

Before transmitting a residual, the encoder will consider the following cases, depending on the IPM:

$$\mathbf{X} = \begin{cases} \mathbf{A}\,\mathbf{x} & 0 \le \text{IPM} \le 10 \\ \mathbf{A}\,\zeta\,\mathbf{x} & 11 \le \text{IPM} \le 18 \\ \mathbf{A}\,\zeta\,\mathbf{x}^T & 19 \le \text{IPM} \le 25 \\ \mathbf{A}\,\mathbf{x}^T & 26 \le \text{IPM} \le 34 \end{cases} \tag{6.4.1}$$

Where $\mathbf{A}$ is a transform designed for the basic IPM set, $\mathbf{x}$ is the current residual and ζ represents the horizontal mirroring operator (top-bottom). Mirroring and transposing operations are used to make residuals compatible with the transforms learnt for the basic IPM set. These operations are only different ways of re-arranging the residual pixels consistently, which come at no computational cost.

6.4.2 Symmetries performances on video coding

Tables 6.4.2 and 6.4.3 show the impact of symmetries on 4×4 and 8×8 MDTC systems, respectively in the ROM and BD-rate plane. As demonstrated before, for the full-symmetrical MDTC systems, ROM

	non-symmetrical		symmetrical	
# tr	ROM	Y BD-rate	ROM	Y BD-rate
1	1.64 kB	-0.49%	0.52 kB	-0.50%
2	3.28 kB	-0.67%	1.03 kB	-0.63%
4	6.56 kB	-0.82%	2.06 kB	-0.72%
8	13.13 kB	-0.93%	4.13 kB	-0.84%
16	26.25 kB	-0.98%	8.25 kB	-0.91%
32	52.50 kB	-1.01%	16.50 kB	-0.93%

Table 6.4.2 – Symmetries impact on 4×4 transforms on the 59seq test set

	non-symmetrical		symmetrical	
# tr	ROM	Y BD-rate	ROM	Y BD-rate
1	6.56 kB	-0.84%	2.06 kB	-0.76%
2	13.13 kB	-1.33%	4.13 kB	-1.24%
4	26.25 kB	-1.72%	8.25 kB	-1.48%
8	52.50 kB	-2.10%	16.50 kB	-1.97%
16	105.00 kB	-2.48%	33.00 kB	-2.22%
32	210.00 kB	-2.79%	66.00 kB	-2.69%
64	420.00 kB	-3.07%	132.00 kB	-2.99%
128	840.00 kB	-3.33%	264.00 kB	-3.25%

Table 6.4.3 – Symmetries impact on 8×8 transforms on the 59seq test set

is decreased to around one third of its original value. Bitrate savings also experiment some losses, tending to decrease as the number of transforms increases, since symmetrical systems are more constrained than non-symmetrical systems. The encoding complexity for symmetrical system is equivalent to the one for non-symmetrical ones, detailed in tables 6.3.1 and 6.3.2, as the number of transforms per IPM remains unchanged.

Table 6.4.4 unveils impact of symmetries for the high performance MDTC system, defined in §4.3.2, on the 59seq test set. As a reminder, the high performance MDTC system uses 16 additional transforms for the 4×4 TUs and 32 for the 8×8 TUs. The symmetrical MDTC system presents slightly better performances for the 59seq test set than the non-symmetrical one. Consequently, exploiting intra prediction residual symmetries can be considered to have no global impact on bitrate savings.

non-symmetrical		symmetrical	
ROM	Y BD-rate	ROM	Y BD-rate
236.25 kB	-3.36%	74.25 kB	-3.50%

Table 6.4.4 – Symmetries impact of the high performance MDTC system on the 59seq test set. This system uses 16 additional transform for 4×4 TUs and 32 for 8×8 TUs

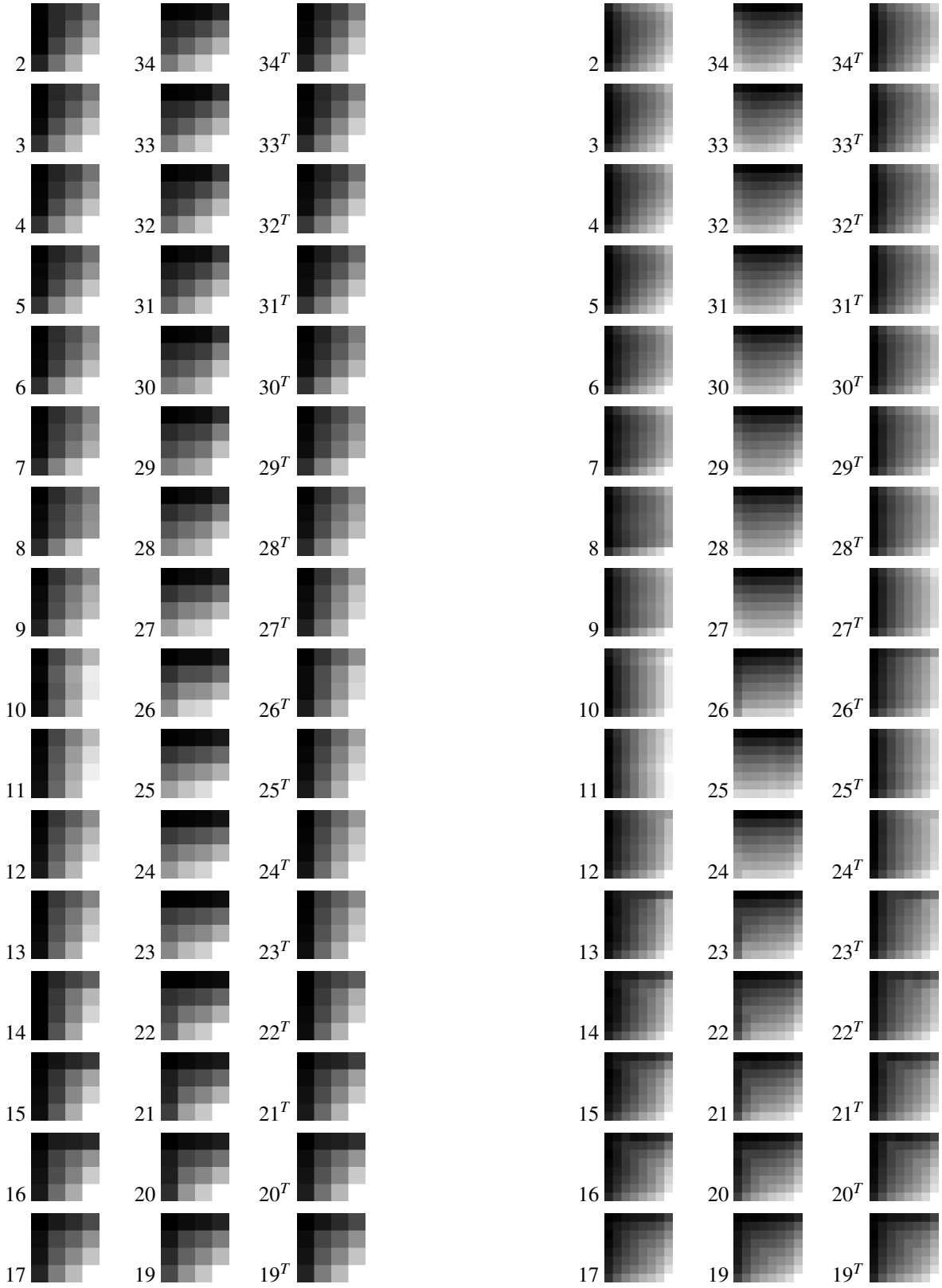


Figure 6.4.2 – Average 4×4 and 8×8 residual profiles showing the symmetries around IPM 18

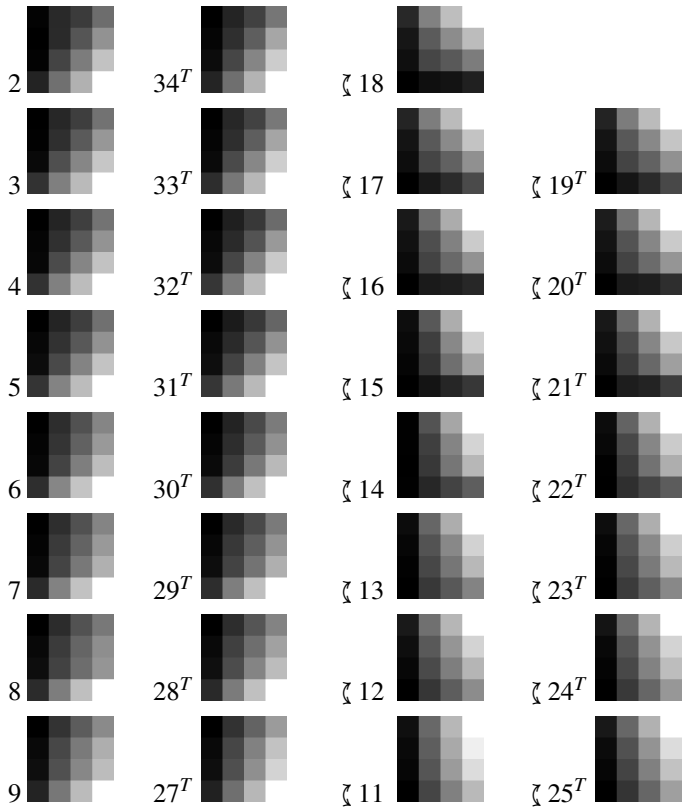


Figure 6.4.3 – Average 4×4 residual profiles showing imposed symmetries with transposition and vertical mirroring

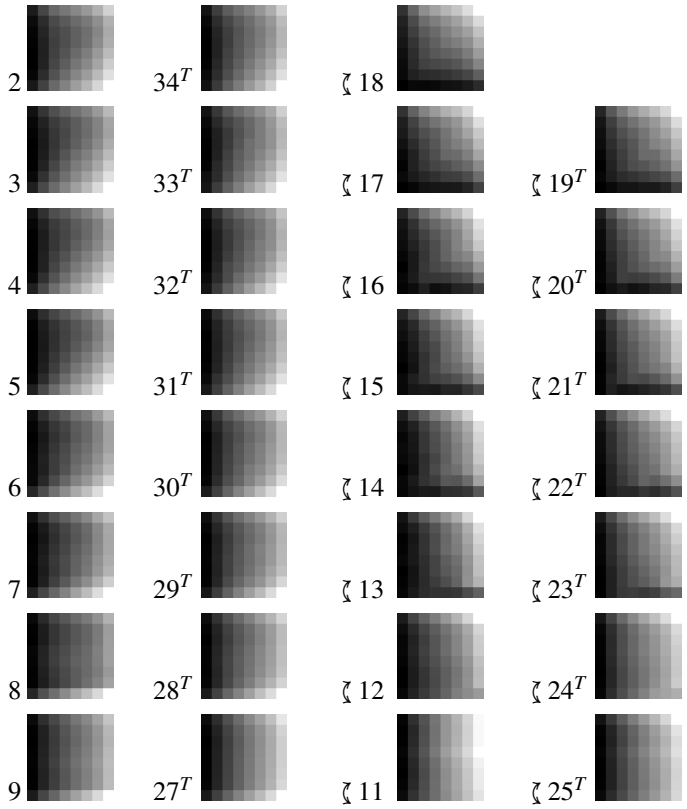


Figure 6.4.4 – Average 8×8 residual profiles showing imposed symmetries with transposition and horizontal mirroring

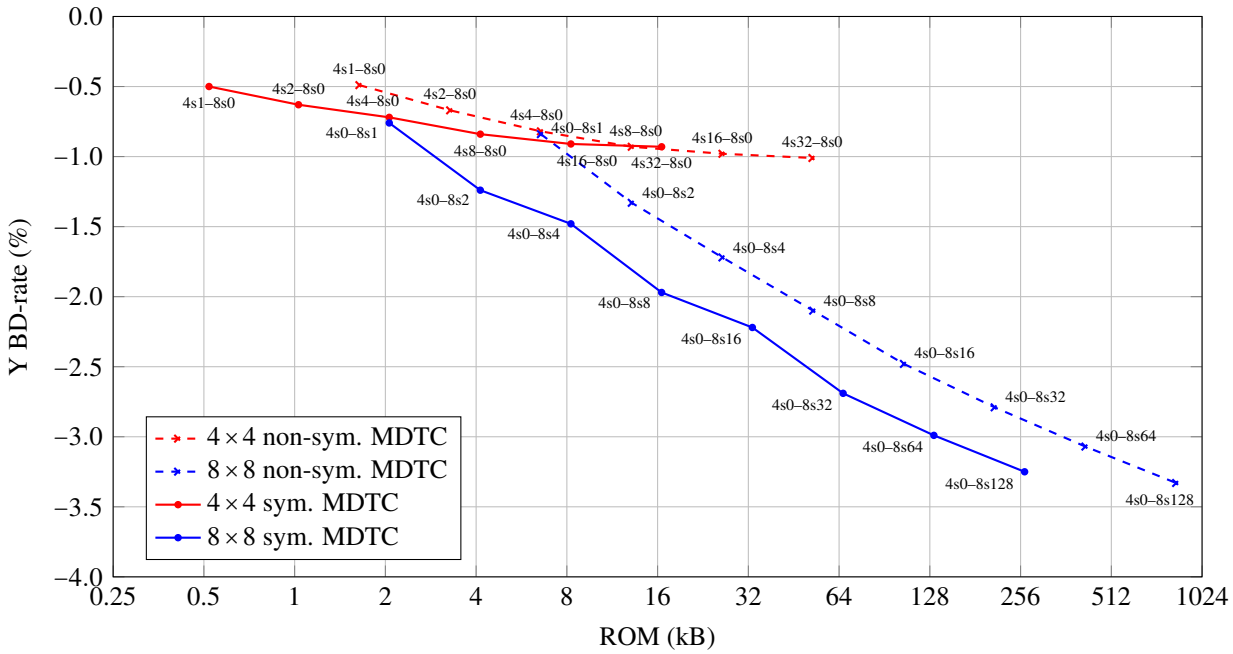


Figure 6.4.5 – Symmetry impact on the ROM — Y BD-rate plane for different MDTC systems

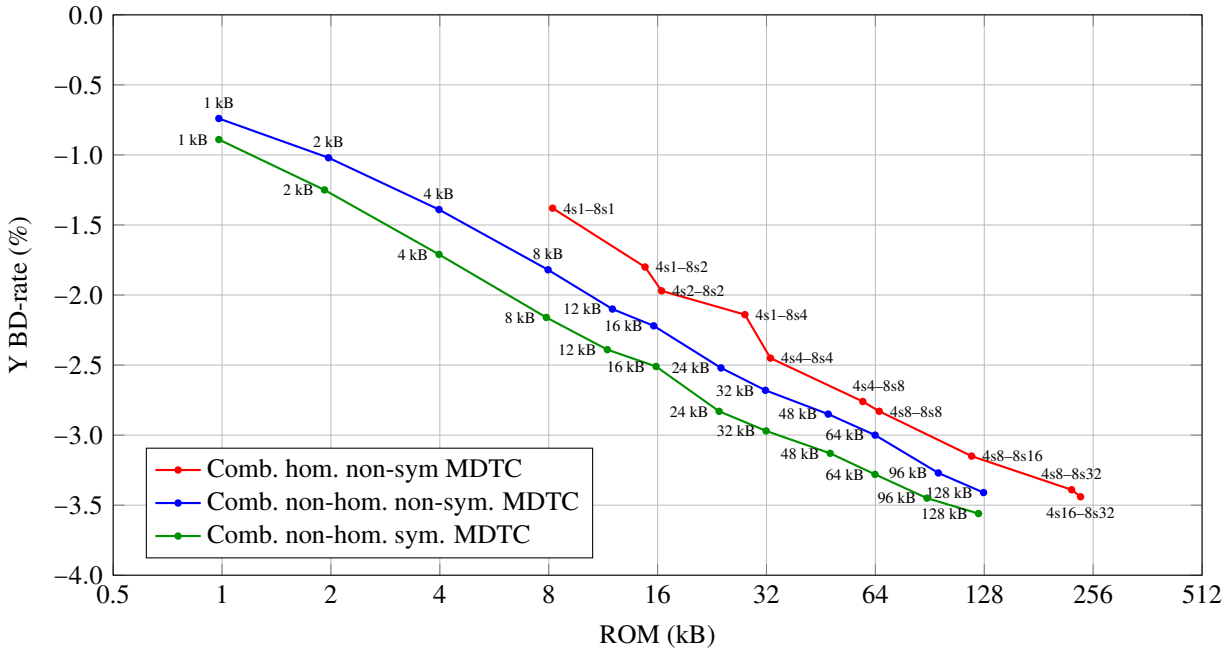


Figure 6.5.1 – MDTC systems making use of non-homogeneous transform repartition and symmetries

6.5 Performances of non-homogeneous symmetrical MDTC systems

The previous two Sections have explored new methods to reduce the storage requirements of MDTC systems. The approaches make use of:

— Heterogeneous repartition of the number of transforms per IPM.

— Symmetries in prediction residuals.

These techniques are not mutually exclusive, as such, this section presents MDTC systems that combine both of them, resulting in non-homogeneous symmetrical MDTC systems. These systems use a different number of transforms per IPM but guaranteeing the symmetrical relations established in §6.4. Building this kind of systems requires the iterative approach presented in §6.3, described in algorithm 2, except that, instead of enabling transforms for one IPM at a time, transforms will be enabled for one of the basic IPMs (0–10) and their symmetrical ones.

Results of the iterations of these non-homogeneous symmetrical systems are presented in figures 6.5.1 and 6.5.2. The first figure compares the systems in the ROM — Y BD-rate plane with the designed systems from previous sections making use of one technique only. The second figure places the systems in the complexity — Y BD-rate plane to ensure that systems remain comparable in terms of encoding times.

As in the non-symmetrical MDTC systems, appendix B details all the symmetrical MDTC systems from table 6.5.1 in tables B.3 and B.4 for 4×4 and 8×8 TUs, respectively. The same pattern is observed: more transforms are used in Planar, DC, horizontal and vertical modes.

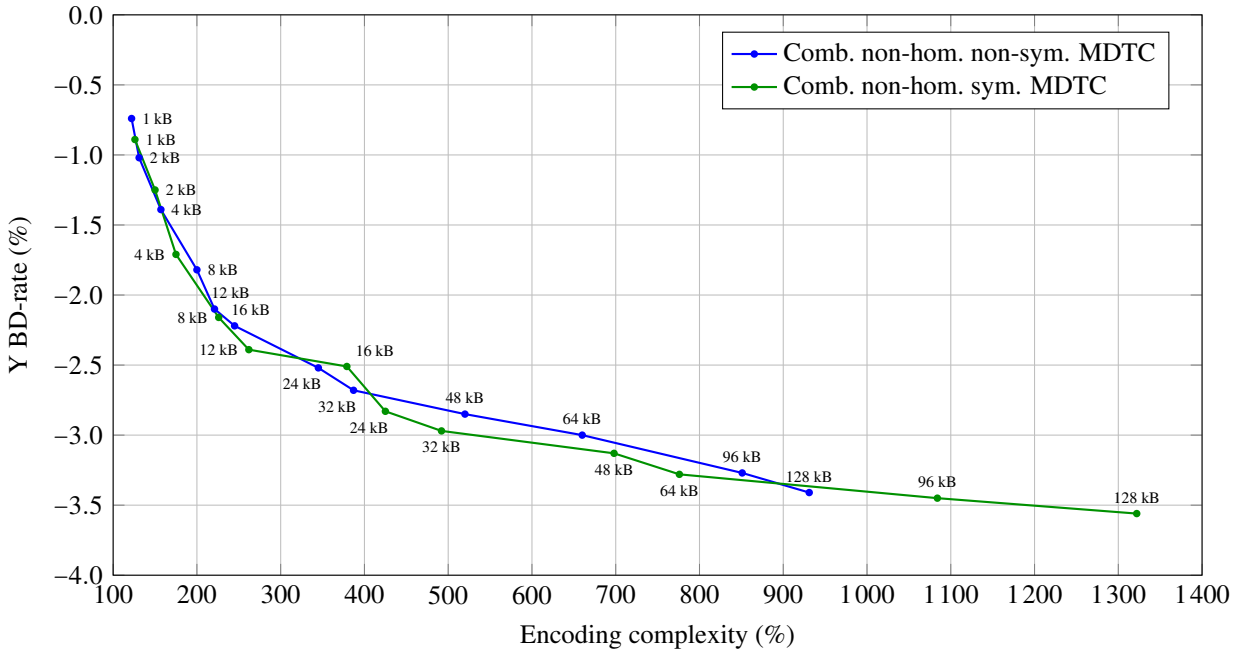


Figure 6.5.2 – Symmetry impact on the coding complexity and BD-rate of non-homogeneous MDTC systems

6.6 Conclusions

MDTC systems presented in Chapter 4 proved the fact that using multiple transforms competing against each other inside every IPM is able to provide bitrate savings of up to 7% when using non-separable transforms, and up to 4% when using separable transforms.

Nonetheless, the storage requirements for those transforms is not negligible. Therefore, this Chapter has presented two approaches to reduce the needed ROM for the transforms.

The first technique proposes a non-homogeneous distribution of the number of transforms per IPM. In Chapter 4 all IPMs were using the same number of transforms, which could be sub-optimal, since not all IPMs are used equally in HEVC. Around a factor of two in terms of ROM has been saved when using non-homogeneous MDTC systems for the same bitrate savings.

The second technique exploited existing symmetries to be able to re-use transforms from different IPM by pre-processing residuals with transposition and mirroring operations. This technique allows

System	ROM	Y BD-rate	Complexity
1 kB	0.98 kB	-0.89%	126%
2 kB	1.92 kB	-1.25%	150%
4 kB	3.98 kB	-1.71%	175%
8 kB	7.88 kB	-2.16%	226%
12 kB	11.62 kB	-2.39%	262%
16 kB	15.84 kB	-2.51%	379%
24 kB	23.67 kB	-2.83%	425%
32 kB	31.92 kB	-2.97%	492%
48 kB	48.00 kB	-3.13%	698%
64 kB	63.84 kB	-3.28%	776%
96 kB	88.88 kB	-3.45%	1084%
128 kB	123.38 kB	-3.56%	1322%

Table 6.5.1 – Non-homogeneous symmetrical MDTC systems with ROM constraints

saving around two thirds of the ROM presented for the MDTC systems from Chapter 4.

Finally, the techniques summarised above have been combined to provide better ROM — Y BD-rate trade-offs. When combining both techniques, storage requirements can be reduced up to 75% of the values presented in Chapter 4 for some systems.

Regarding complexity, using a non-homogeneous number of transforms per IPM decreases the decoding time, since fewer transforms need to be tested. On the other hand, symmetries allow putting more transforms without having an impact to the ROM requirements. As a result, the encoding complexity remains comparable to the one presented in Chapter 4.

Chapter 7

MDTC using discrete trigonometric transforms

Contents

7.1 Introduction

Chapter 4 has unveiled the potential of gains achievable through the MDTC technique. Nevertheless, the presented system has a level of complexity which is too high to be used in commercial applications. As a result, the Chapters 5 and 6 propose two different approaches to make the system less complex.

Chapter 5 simplifies the system by designing transforms that make use of only one base vector. This makes transforming a block less complex than with a regular transform, even if the incomplete transform is non-separable. Benefits of this approach are observed on the decoder, being less complex than that of HEVC, while still providing bitrate savings.

Chapter 6 proposes two ways of reducing the complexity of separable MDTC systems, especially the storage requirements:

— Using a different number of transforms in each IPM.

— Taking advantage of the symmetries that might exist within different IPMs.

Both approaches provide notable savings in the ROM required to store the transforms. Besides, the encoder complexity is slightly decreased using non-homogeneous systems, since there are fewer transforms to test in each IPM.

However, even if the main motivation of the approaches above is to reduce the complexity of the high performance MDTC system presented in Chapter 4, the improvements have only been focused on the ROM axis, leaving the encoder and decoder complexities decreases to be a side effect of the storage constraint.

This Chapter proposes a simplification approach that takes into account the storage requirements, the encoding complexity and the decoding complexity at the same time. It is also worth-noticing that the systems described in this Chapter are in early stages of development and, therefore, not as mature as the systems presented previously.

The experiments presented in this Chapter served as a base for the work published in [7].

Size \ Type	Reg. sep. transf.		DTT	
	Operations	ROM	Operations	ROM
4×4	128	48 B	64	16 B
8×8	1024	192 B	192	64 B

Table 7.2.1 – Comparison of ROM and complexity between DTTs and regular separable transforms

7.2 The discrete trigonometric transform family (DCTs & DSTs)

Chapter 3 has highlighted the appropriateness of the RDOT design method over the KLT for video coding. However, for the sake of simplicity, a different family of transforms is considered in this Chapter: the Discrete Trigonometric Transforms (DTTs).

DTTs are orthogonal transforms based on trigonometric functions. This family of transforms consists of 8 types (I to VIII) of Discrete Cosine Transforms (DCTs) and Discrete Sine Transforms (DSTs) [38, 37].

Historically, the DCT-II has been the *de facto* standard transform for image and video coding applications. Recently, other transforms from the DTT family are starting to arise the interest in video coding applications:

— The DST-VII is used in HEVC for 4×4 intra prediction luma residuals.

— The DST-III has been proposed for inter-layer prediction residuals in scalable video coding [19].

The interest of the DTTs in this Chapter is motivated by the existing fast algorithms for transform implementation, which are notably less complex than a full matrix multiplication required by generic block transforms. The algorithmic complexity for a 1D DTT is in the order of $N\log_2(N)$ instead of the N^2, where N stands for the transform size [37]. Table 7.2.1 compares the number of operations required for a regular separable 2D transform and a DTT using a fast algorithm. Results are presented for both 4×4 and 8×8 TUs.

Since DTTs coefficients can be computed using an analytical formula, their storage requirement is negligible. However, a scanning matrix is still required to sort the transformed coefficients in a globally decreasing order to ease the entropy coding stage. As a result, the necessary storage amount per transform is N^2 bytes. For RDOTs, the required memory is $3N^2$ bytes per transform, as the horizontal and vertical transforms are stored along with a dedicated scanning pattern. Storage values are compared in table 7.2.1.

Nevertheless, these transforms are a restrained subset of the orthogonal transform class, as such, their performance in coding gains is expected to be lower than that of RDOTs.

To sum up, DTTs are able to divide the storage requirements of separable transforms by a factor of 3 and present a lower computational complexity due to fast algorithms. This Chapter explores whether the ROM and complexity reductions balance the expected losses in bitrate savings regarding separable RDOTs.

The formal definition of the 8 types of DCT and DST can be found in tables 7.2.3 and 7.2.4, respectively. Some of them have scaling factors, namely $\varepsilon_n, \varepsilon_n$, that vary depending on the position in the matrix to ensure the energy preservation property.

Transforms in the DTT family present strong relations amongst them and their inverse transforms [39, 44], detailed in table 7.2.2. Since all inverse DCTs and DSTs can be expressed in terms of a direct transform, there are 16 unique 1D transforms in total. Combining them as a horizontal and vertical transforms leads to a total of 256 unique 2D transforms, which are able to capture different kinds of residual patterns. Some of these transforms are displayed in figure 7.2.1.

As an observation, using an orthogonal transform $\mathbf{A}$ for the rows of a block and another orthogonal transform $\mathbf{B}$ for the resulting columns does not result in the same signal in the transform domain as if the operations are carried out the other way around if $\mathbf{B} \neq \mathbf{A}$. Nonetheless, both resulting signals can be expressed in terms of the other. Assuming $\mathbf{x}$ is the input signal and $\mathbf{X}$ stands for its representation in the

Direct transform	Inverse transform
DCT-I	DCT-I
DCT-II	DCT-III
DCT-III	DCT-II
DCT-IV	DCT-IV
DCT-V	DCT-V
DCT-VI	DCT-VII
DCT-VII	DCT-VI
DCT-VIII	DCT-VIII
DST-I	DST-I
DST-II	DST-III
DST-III	DST-II
DST-IV	DST-IV
DST-V	IDST-V
DST-VI	DST-VII
DST-VII	DST-VI
DST-VIII	DST-VIII

Table 7.2.2 – Relationships between the different members of the DTT family

transform domain:

$$\mathbf{X} = \mathbf{B}\left(\mathbf{A}\mathbf{x}^T\right)^T = \mathbf{B}\mathbf{x}\mathbf{A}^T \tag{7.2.1}$$

$$\mathbf{X} = \mathbf{A}\left(\mathbf{B}\mathbf{x}^T\right)^T = \mathbf{A}\mathbf{x}\mathbf{B}^T \tag{7.2.2}$$

If instead of transforming $\mathbf{x}$ directly, a transposition is applied before:

$$\mathbf{X} = \mathbf{B}\left(\mathbf{A}\left(\mathbf{x}^T\right)^T\right)^T = \mathbf{B}\mathbf{x}^T\mathbf{A}^T = \left(\mathbf{A}\mathbf{x}\mathbf{B}^T\right)^T \tag{7.2.3}$$

$$\mathbf{X} = \mathbf{A}\left(\mathbf{B}\left(\mathbf{x}^T\right)^T\right)^T = \mathbf{A}\mathbf{x}^T\mathbf{B}^T = \left(\mathbf{B}\mathbf{x}\mathbf{A}^T\right)^T \tag{7.2.4}$$

Meaning that a transposition in the spatial domain is equivalent to invert the order in which the row and column transforms are applied and then transpose the result.

Additionally, if a different scanning is allowed per transform, transposing or not a signal in the spatial domain become equivalent operations, since the only difference is the order in which the transform coefficients are output, but their values remain the same, resulting in fewer combinations than 256.

In order to expand the number of transforms available, three geometrical operations have been considered:

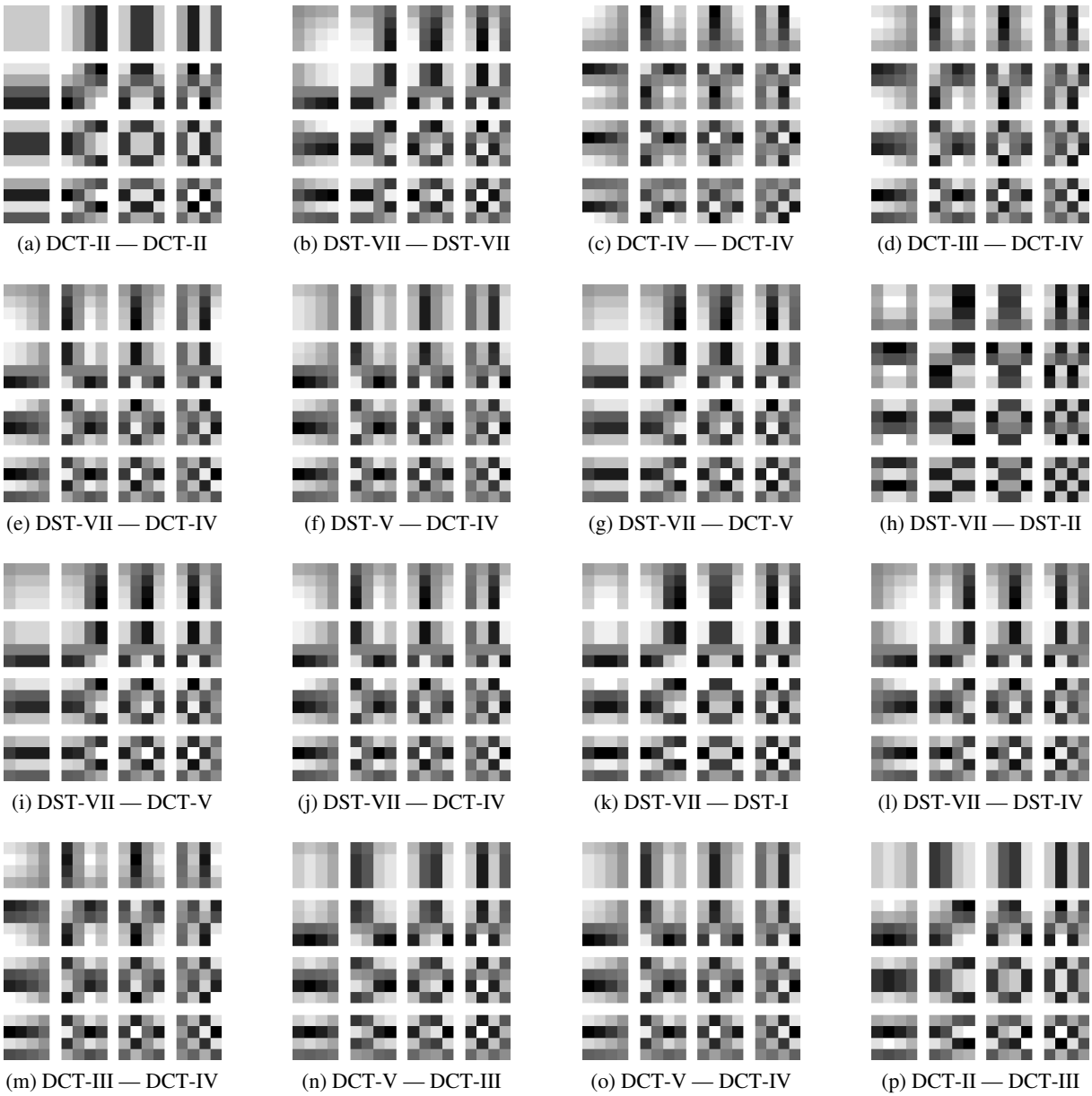
— Residual transposition: $\mathbf{x}^T$

— Horizontal mirroring: $\overleftrightarrow{\mathbf{x}}$

— Vertical mirroring: $\overleftrightarrow{\mathbf{x}}$

These operators can be combined, resulting into $2^3 = 8$ different operations. By applying these pixel permutations to the residual before the transform stage, the number of 2D transforms can be increased from 256 to 2048.

This kind of relations will be taken into account when designing DTT-based MDTC systems in the next Section.



(a) DCT-II — DCT-II (b) DST-VII — DST-VII (c) DCT-IV — DCT-IV (d) DCT-III — DCT-IV

(e) DST-VII — DCT-IV (f) DST-V — DCT-IV (g) DST-VII — DCT-V (h) DST-VII — DST-II

(i) DST-VII — DCT-V (j) DST-VII — DCT-IV (k) DST-VII — DST-I (l) DST-VII — DST-IV

(m) DCT-III — DCT-IV (n) DCT-V — DCT-III (o) DCT-V — DCT-IV (p) DCT-II — DCT-III

Figure 7.2.1 – 4×4 DTT combination examples

$$\text{DCT-I:}\quad \left[C_N^I\right]_{n,k} = \sqrt{\frac{2}{N-1}}\varepsilon_n\varepsilon_k\cos\left(\frac{\pi nk}{N-1}\right) \qquad \varepsilon_n,\varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & n,k=0\\ \frac{1}{\sqrt{2}} & n,k=N-1\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-II:}\quad \left[C_N^{II}\right]_{n,k} = \sqrt{\frac{2}{N}}\varepsilon_k\cos\left(\frac{\pi(2n+1)k}{2N}\right) \qquad \varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & k=0\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-III:}\quad \left[C_N^{III}\right]_{n,k} = \sqrt{\frac{2}{N}}\varepsilon_n\cos\left(\frac{\pi(2k+1)n}{2N}\right) \qquad \varepsilon_n = \begin{cases}\frac{1}{\sqrt{2}} & n=0\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-IV:}\quad \left[C_N^{IV}\right]_{n,k} = \sqrt{\frac{2}{N}}\cos\left(\frac{\pi(2n+1)(2k+1)}{4N}\right)$$

$$\text{DCT-V:}\quad \left[C_N^{V}\right]_{n,k} = \frac{2}{\sqrt{2(N-1)+1}}\varepsilon_n\varepsilon_k\cos\left(\frac{2\pi nk}{2N-1}\right) \qquad \varepsilon_n,\varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & n,k=0\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-VI:}\quad \left[C_N^{VI}\right]_{n,k} = \frac{2}{\sqrt{2(N-1)}}\varepsilon_n\varepsilon_k\cos\left(\frac{\pi(2n+1)k}{2N-1}\right) \qquad \varepsilon_n = \begin{cases}\frac{1}{\sqrt{2}} & n=N-1\\ 1 & \text{otherwise}\end{cases} \quad \varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & k=0\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-VII:}\quad \left[C_N^{VII}\right]_{n,k} = \frac{2}{\sqrt{2(N-1)}}\varepsilon_n\varepsilon_k\cos\left(\frac{\pi(2k+1)n}{2N-1}\right) \qquad \varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & k=N-1\\ 1 & \text{otherwise}\end{cases} \quad \varepsilon_n = \begin{cases}\frac{1}{\sqrt{2}} & n=0\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DCT-VIII:}\quad \left[C_N^{VIII}\right]_{n,k} = \frac{2}{\sqrt{2(N-1)}}\cos\left(\frac{\pi(2n+1)(2k+1)}{4N-2}\right)$$

Table 7.2.3 – DCT definitions of size N, where $n,k = 0,\ldots,N-1$

$$\text{DST-I:}\quad \left[S_N^I\right]_{n,k} = \sqrt{\frac{2}{2N+1}}\sin\left(\frac{\pi(n+1)(k+1)}{N+1}\right)$$

$$\text{DST-II:}\quad \left[S_N^{II}\right]_{n,k} = \sqrt{\frac{2}{2N}}\varepsilon_k\sin\left(\frac{\pi(2n+1)(k+1)}{2N}\right) \qquad \varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & k=N-1\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DST-III:}\quad \left[S_N^{III}\right]_{n,k} = \sqrt{\tfrac{2}{2N}}\varepsilon_n\sin\left(\tfrac{\pi(2n+1)(k+1)}{2N}\right) \qquad \varepsilon_n = \begin{cases}\frac{1}{\sqrt{2}} & n=N-1\\ 1 & \text{otherwise}\end{cases}$$

$$\text{DST-IV:}\quad \left[S_N^{IV}\right]_{n,k} = \frac{2}{\sqrt{2N}}\sin\left(\frac{\pi(2n+1)(2k+1)}{4N}\right)$$

$$\text{DST-V:}\quad \left[S_N^{V}\right]_{n,k} = \frac{2}{\sqrt{2N+1}}\sin\left(\frac{2\pi(n+1)(k+1)}{2N+1}\right)$$

$$\text{DST-VI:}\quad \left[S_N^{VI}\right]_{n,k} = \frac{2}{\sqrt{2N+1}}\sin\left(\frac{\pi(2n+1)(k+1)}{2N+1}\right)$$

$$\text{DST-VII:}\quad \left[S_N^{VII}\right]_{n,k} = \frac{2}{\sqrt{2N+1}}\sin\left(\frac{\pi(2k+1)(n+1)}{2N+1}\right)$$

$$\text{DST-VIII:}\quad \left[S_N^{VIII}\right]_{n,k} = \frac{2}{\sqrt{2(N-1)+1}}\varepsilon_n\varepsilon_k\sin\left(\frac{\pi(2n+1)(2k+1)}{2((2N-1)+1)}\right) \qquad \varepsilon_n,\varepsilon_k = \begin{cases}\frac{1}{\sqrt{2}} & n,k=N-1\\ 1 & \text{otherwise}\end{cases}$$

Table 7.2.4 – DST definitions of size N, where $n,k = 0,\ldots,N-1$

7.3 Design of DTT-based MDTC systems

The RDOT metric, defined in §2.3.2, has proved to be a good way of designing transforms that offer a balance between the distortion introduced by the quantisation and the sparsity of transformed coefficients. Nonetheless, the RDOT metric has also been used in Chapters 3 and 4 to measure the appropriateness of a given transform to compactly represent a residual in the rate-distortion plane. By using this metric, one is able to evaluate the performance of a transform on a residual and define a set of residuals for which a transform gives the best rate-distortion trade-off. In practice, this metric is able to rate a transform, i.e. its appropriateness with respect to a set of residuals.

Due to the fact that MDTC systems learnt using the RDOT metric lead to significant bitrate savings over HEVC, it is also used in this Chapter to design a DTT-based MDTC system. The learning method is based on Algorithm 1 for the classification part: residuals from the learning set are assigned to the transform that provide the lowest RDOT metric value. The main difference is that, when using DTTs, transforms are already learnt, so the learning phase of the algorithm can be skipped. Instead, all transforms will be tested against the learning set, and only the set of transforms that provide the lowest RDOT metric will be retained.

As explained in the previous Section, in order to expand the transform space when using the DTT family, which consists of 8 types of DCT, 8 types of DST and their inverse versions, 8 spatial pixel permutations have also been considered. The chosen permutations are combinations of transposed and mirrored (vertically and horizontally) versions of residuals so as not to break pixel correlations and relative pixel distances within a block. Using DTTs together with the proposed residual modifications leads to a total of 2048 possible transform combinations. In order to find the optimal DTT-based MDTC system, one would have to compute the best set of N transforms from the 2048 available giving the lowest value of the RDOT metric. This results into

$$\binom{2048}{N} \tag{7.3.1}$$

possible transform combinations. Fortunately, due to the nature of the DTT family, there are many redundancies within the transform combinations. For instance, for 4×4 transforms, the 2048 list reduced to 256 unique transforms, if a re-ordering is allowed in the transform domain. As an example, for $N = 4$ additional transforms, the number of combinations that need to be tested is:

$$\binom{256}{4} = 174\,792\,640 \approx 1.75\times10^{8} \tag{7.3.2}$$

instead of:

$$\binom{2048}{4} = 730\,862\,190\,080 \approx 7.31\times10^{11} \tag{7.3.3}$$

Meaning that the number of possible combinations is reduced by more than a factor of 4000.

In order to find a good combination of transforms and permutations, all combinations are put into a list, which is iterated through. Then, the first N transforms are used to classify the residuals into N groups, that is, each residual is assigned to the transform that gives the lowest value of the RDOT metric defined in (2.3.2). Afterwards, the transform $N+1$ tentatively replaces each transform and, if it makes the global RDOT metric lower, it replaces the outperformed transform, otherwise it is discarded. The same step is carried out for the next transform until the end of the list, using a different number of transforms per IPM and exploiting prediction.

This approach does not provide the global optimal and the classification algorithm is subject to be improved. Nevertheless, this simple method allows determining the validity of the approach. Furthermore, the classification algorithm becomes even less optimal when the number of additional transforms increases.

Since DTTs have a lower complexity than regular separable transforms, the main objective in this Chapter is to build low complexity MDTC systems, with high emphasis in the decoding time and ROM requirements.

System	ROM	Y BD-rate	Complexity
1 kB	0.98 kB	-1.28%	164%
2 kB	1.97 kB	-1.54%	177%
4 kB	3.97 kB	-1.83%	230%

Table 7.4.1 – Non-homogeneous symmetrical DTT-based MDTC systems with ROM constraints

7.4 Performances of DTT-based MDTC systems

In the previous Chapter, two approaches are proposed to improve the existing separable MDTC systems in the ROM — Y BD-rate plane: using a different number of transforms per IPM and exploiting prediction residuals symmetries to re-use the same transforms amongst symmetrical IPMs. MDTC systems that make use of both techniques at the same time present the best trade-off in the three axes: ROM, BD-rate and even encoding complexity.

As in Chapter 6, in order to build iterated non-homogeneous symmetrical MDTC systems, several configurations have been previously learnt per IPM in both 4×4 and 8×8 TU sizes. These configurations enable from 1 to 16 additional transforms for 4×4 TUs and from 1 to 32 for 8×8 TUs in steps of powers of 2.

Due to the intention of designing low complexity systems, only configurations of 1, 2 and 4 kB have been assembled. Table 7.4.1 contains the iterated non-homogeneous DTT-based MDTC systems for 1, 2 and 4 kB. A detailed description of the number of transforms used in each IPM for 4×4 and 8×8 TUs can be found in appendix B, more precisely in table B.5.

A comparison with previously designed systems (the symmetrical and non-symmetrical versions of the iterated non-homogeneous MDTC systems) is presented in figures 7.4.1 and 7.4.2 for ROM and encoding complexity, respectively. The first figure reveals how the DTT-based MDTC systems outperform the RDOT-based MDTC systems in the ROM — Y BD-rate plane. The sub-optimality of the learning algorithm can be observed as the number of additional DTTs increases: the DTT-based MDTC system gets closer to the RDOT-based MDTC systems. Due to the low storage requirements of the DTTs, more transforms can be fit for a given ROM constraint, leading to more coding possibilities, and thus to more encoding complexity. Moreover, the DTTs have been implemented as separable transforms inside the encoder, therefore, there is room for improvement in terms of encoding complexity once they make use of fast algorithms.

The system that offers the best performances in terms of BD-rate, labelled as 4 kB, presents a particular usage of transforms from the DTT family. Tables 7.4.2 and 7.4.3 detail the different transform combinations used for 4×4 and 8×8 TU sizes, respectively, as well as the number of occurrences.

For 4×4 blocks, the 4 kB system uses a total of 38 additional transforms (including all IPMs), which are combinations of different DTTs. Despite having 256 DTTs from where to choose, only 15 different transforms are used with different levels of repetition. The most used transforms are the DCT-IV–DCT-IV, DCT-III–DCT-IV, the DST-VII–DCT-IV and the DCT-V–DCT-IV pairs. For 8×8 blocks, the system uses 55 additional transforms, of which only 20 are unique. From these unique transforms, there are two combinations that stand out: the DST-VII–DST-VII and the DST-VII–DCT-V pairs. Moreover, the DST-VII and the DCT-V appear frequently in the other combinations. The usage of these transforms is summarised in tables 7.4.2 and 7.4.3 for 4×4 and 8×8 TUs, respectively.

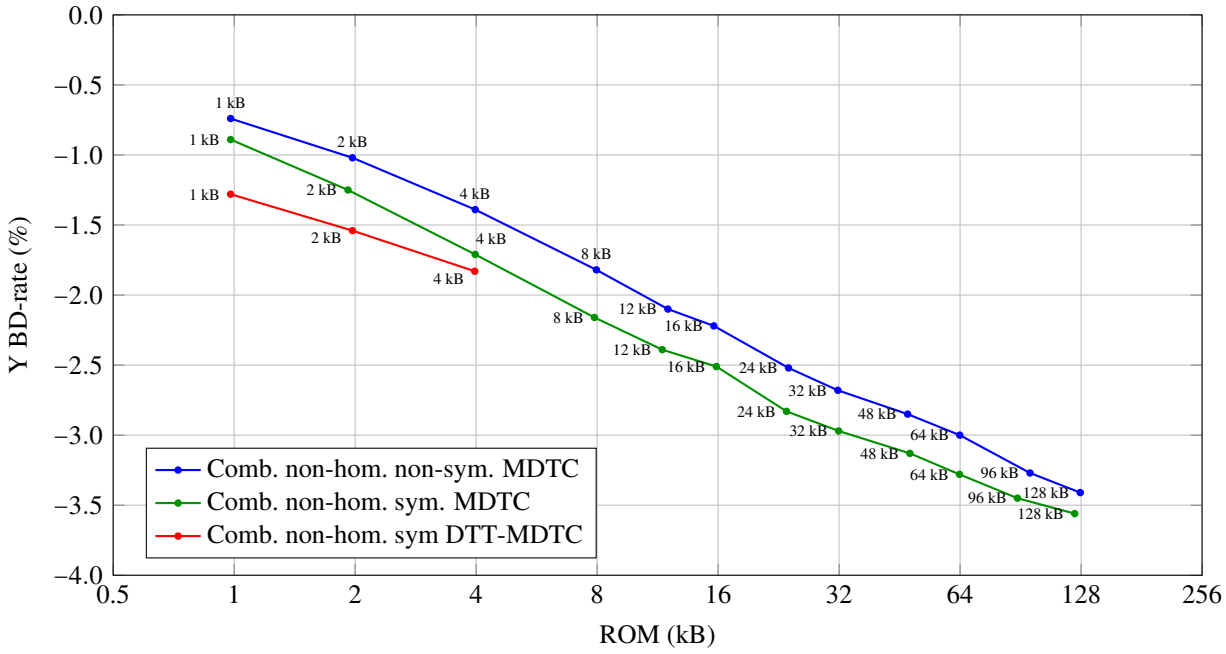


Figure 7.4.1 – DTT-based MDDT systems in the ROM — Y BD-rate plane compared to other non-homogeneous MDTC systems

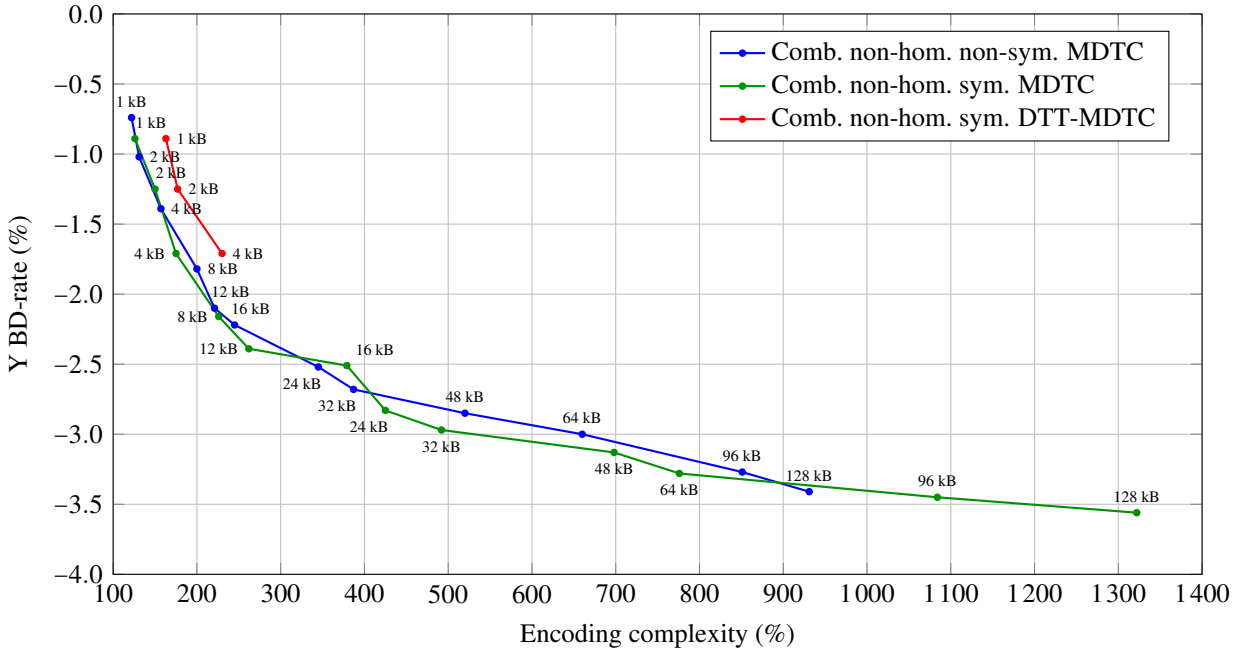


Figure 7.4.2 – DTT-based MDDT systems in the Complexity — Y BD-rate plane compared to other non-homogeneous MDTC systems

Horizontal transform	Vertical transform	Number
DCT-I	DCT-III	1
DCT-II	DCT-IV	1
DCT-III	DCT-III	1
DCT-III	DCT-IV	6
DCT-IV	DCT-IV	8
DCT-V	DCT-III	1
DCT-V	DCT-IV	5
DCT-V	DST-II	1
DST-I	DCT-IV	1
DST-V	DST-V	1
DST-VII	DCT-III	1
DST-VII	DCT-IV	6
DST-VII	DCT-V	2
DST-VII	DST-II	2
DST-VII	DST-V	1

Table 7.4.2 – The different 4×4 transform combinations for the 4 kB DTT-based MDTC system

Horizontal transform	Vertical transform	Number
DCT-II	DCT-IV	1
DCT-II	DST-V	1
DCT-III	DCT-IV	2
DCT-III	DST-VII	1
DCT-IV	DCT-IV	2
DCT-V	DCT-III	2
DCT-V	DCT-VII	1
DCT-V	DST-II	1
DCT-V	DST-III	1
DCT-V	DST-IV	2
DCT-V	DST-V	1
DCT-V	DST-VI	1
DST-I	DCT-IV	1
DST-VII	DCT-III	3
DST-VII	DCT-IV	4
DST-VII	DCT-V	10
DST-VII	DST-I	4
DST-VII	DST-III	1
DST-VII	DST-IV	3
DST-VII	DST-VII	12

Table 7.4.3 – The different 8×8 transform combinations for the 4 kB DTT-based MDTC system

7.5 Conclusions

Chapter 6 presented some ways of reducing the storage requirements by using a different number of transforms in each IPM and by exploiting intra prediction residuals symmetries.

This Chapter goes one step further on simplifying MDTC systems: instead of using RDOT-based MDTC systems, RDOTs are replaced by DTTs. The coefficients of these trigonometric transforms can be deduced analytically, hence the only storage requirement is the scanning pattern. Although fast algorithms exist for DTTs, this Chapter has implemented them as regular separable transforms. As a result, the encoding and decoding times presented here do not represent the values that an actual implementation would have.

Finally, the DCT-based MDTC systems explored in this Chapter are preliminary, but still very competitive and promising with regards to the more complex approaches using separable RDOTs. As such, coding complexity is subject to further improvements once DTTs are implemented using fast algorithms. Moreover, revisiting the learning algorithm to fix its sub-optimality can lead to even better performances of the DTT-based MDTC systems.

Chapter 8

Proposed MDTC configurations

Contents

8.1 Motivation

Numerous alternative MDTC systems have been designed and presented throughout this thesis. As a result, this Chapter presents several interesting alternatives that offer different levels of trade-offs between the provided BD-rate and the added complexity.

In Chapters 6 and 7, various systems have been designed with complete independence from the HEVC test set proposed in the CTC [11]. In order to provide results in well specified test conditions with a known set of test sequences, this Chapter evaluates the most interesting systems in terms of trade-off between complexity (ROM and encoding time) and bitrate savings on the HEVC test set.

8.2 Retained systems

Some interesting systems in terms of bitrate savings, encoding time, decoding time and storage requirements have been selected to cover different complexity and performance trade-offs. The retained systems have been designed using a non-homogeneous number of transforms per IPM (§6.3) and exploiting existing residual symmetries (§6.4). Depending on the desired performance and complexity trade-off, the proposed systems are:

— 4 kB DTT-based MDTC

— 16 kB RDOT-based MDTC

— 32 kB RDOT-based MDTC

— 64 kB RDOT-based MDTC

— 128 kB RDOT-based MDTC

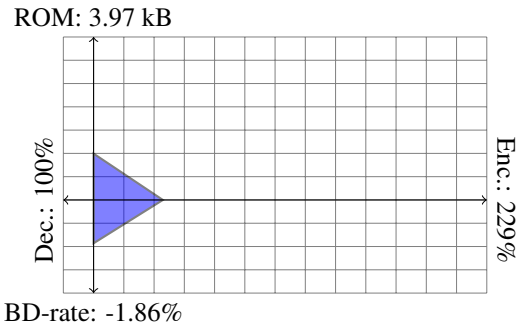
Table 8.2.1 contains a detailed summary of the bitrate savings obtained with each system in both AI and RA coding configurations, as well as the encoding complexity. The actual results per sequence of the HEVC test set for AI and RA are provided in tables 8.2.2 and 8.2.3, respectively. Moreover, a graphical representation of the AI coding configuration bitrate savings is shown in figure 8.2.2 and in figure 8.2.3 for RA.

	AI					RA				
	DTT	RDOT				DTT	RDOT			
	4 kB	16 kB	32 kB	64 kB	128 kB	4 kB	16 kB	32 kB	64 kB	128 kB
Class A	-1.55	-2.07	-2.45	-2.73	-3.11	-0.67	-0.98	-1.19	-1.33	-1.54
Class B	-1.74	-2.27	-2.74	-3.05	-3.28	-1.01	-1.37	-1.63	-1.80	-1.96
Class C	-2.02	-3.00	-3.36	-3.73	-3.85	-1.23	-1.94	-2.09	-2.31	-2.45
Class D	-2.05	-2.96	-3.26	-3.62	-3.87	-1.09	-1.60	-1.76	-1.97	-2.09
Class E	-1.84	-2.66	-3.19	-3.65	-3.90	-2.41	-3.46	-3.95	-4.46	-4.86
Class F	-1.97	-4.34	-4.40	-4.66	-4.91	-1.78	-3.73	-3.73	-3.93	-4.22
Worst	-0.57	-0.58	-0.64	-0.64	-0.61	0.21	0.20	0.27	0.26	0.23
Best	-2.92	-5.16	-5.29	-5.60	-5.92	-2.68	-5.11	-5.14	-5.43	-5.74
Median	-1.98	-3.03	-3.40	-3.75	-3.95	-1.28	-2.09	-2.24	-2.44	-2.62
Complexity	229	372	481	761	1297	106	114	119	133	163
Mean	-1.86	-2.87	-3.21	-3.55	-3.79	-1.31	-2.09	-2.29	-2.53	-2.73

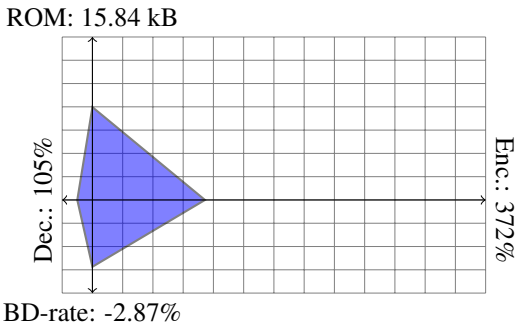
Table 8.2.1 – Summary of the retained MDTC systems referred to HEVC (%). Y BD-rates are presented per Class, as well as their mean, median, best and worst values. The encoding complexity of each system is also provided.

		DTT	RDOT			
	Sequence	4 kB	16 kB	32 kB	64 kB	128 kB
Class A (2560×1600)	NebutaFestival	-0.77	-0.77	-0.95	-1.16	-1.40
	PeopleOnStreet	-2.59	-3.36	-3.95	-4.46	-5.23
	SteamLocomotiveTrain	-0.57	-0.58	-0.64	-0.64	-0.61
	Traffic	-2.27	-3.59	-4.25	-4.68	-5.20
	Average	-1.55	-2.07	-2.45	-2.73	-3.11
Class B (1920×1080)	BasketballDrive	-1.33	-1.68	-2.17	-2.54	-2.63
	BQTerrace	-1.47	-2.22	-2.68	-3.04	-3.24
	Cactus	-2.33	-3.02	-3.59	-4.02	-4.19
	Kimono1	-0.63	-0.85	-1.00	-1.07	-1.30
	ParkScene	-2.92	-3.57	-4.26	-4.59	-5.04
	Average	-1.74	-2.27	-2.74	-3.05	-3.28
Class C (832×480)	BasketballDrill	-1.39	-2.47	-2.54	-2.83	-2.99
	BQMall	-2.30	-3.12	-3.68	-4.09	-4.36
	PartyScene	-2.43	-3.50	-3.85	-4.22	-4.40
	RaceHorses	-1.96	-2.92	-3.37	-3.77	-3.65
	Average	-2.02	-3.00	-3.36	-3.73	-3.85
Class D (416×240)	BasketballPass	-1.77	-2.69	-3.14	-3.47	-3.73
	BlowingBubbles	-2.25	-3.04	-3.35	-3.70	-3.95
	BQSquare	-2.10	-3.50	-3.62	-3.94	-4.24
	RaceHorses	-2.10	-2.63	-2.93	-3.36	-3.54
	Average	-2.05	-2.96	-3.26	-3.62	-3.87
Class E (1280×720)	FourPeople	-2.31	-3.26	-3.88	-4.42	-4.74
	Johnny	-1.47	-2.17	-2.66	-3.08	-3.24
	KristenAndSara	-1.74	-2.53	-3.03	-3.45	-3.72
	Average	-1.84	-2.66	-3.19	-3.65	-3.90
Class F (various resolutions)	BasketDrillText	-1.82	-3.42	-3.43	-3.73	-3.96
	ChinaSpeed	-1.58	-3.84	-3.98	-4.24	-4.45
	SlideEditing	-2.01	-4.93	-4.90	-5.06	-5.31
	SlideShow	-2.46	-5.16	-5.29	-5.60	-5.92
	Average	-1.97	-4.34	-4.40	-4.66	-4.91
All sequences	Overall	-1.86	-2.87	-3.21	-3.55	-3.79

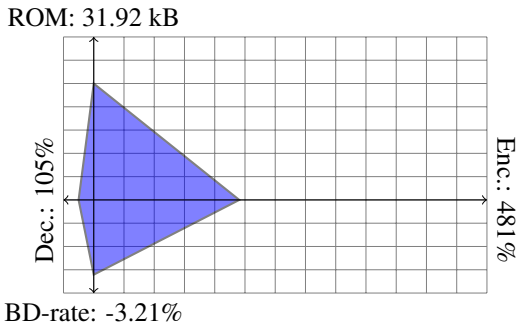
Table 8.2.2 – Y BD-rate (%) for proposed MDTC systems in AI



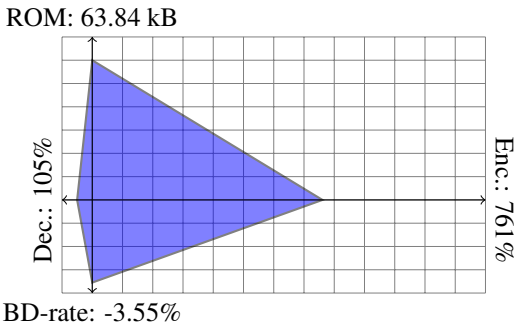
(a) 4 kB DTT-based MDTC



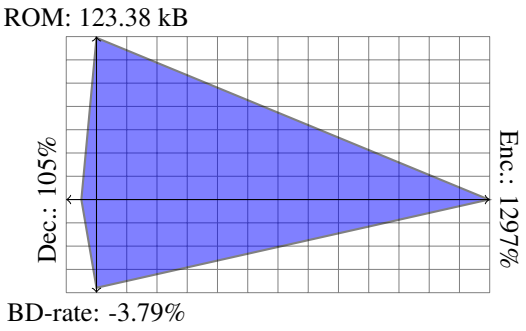
(b) 16 kB RDOT-based MDTC



(c) 32 kB RDOT-based MDTC



(d) 64 kB RDOT-based MDTC



(e) 128 kB RDOT-based MDTC

Figure 8.2.1 – Graphical comparison of the five retained MDTC systems

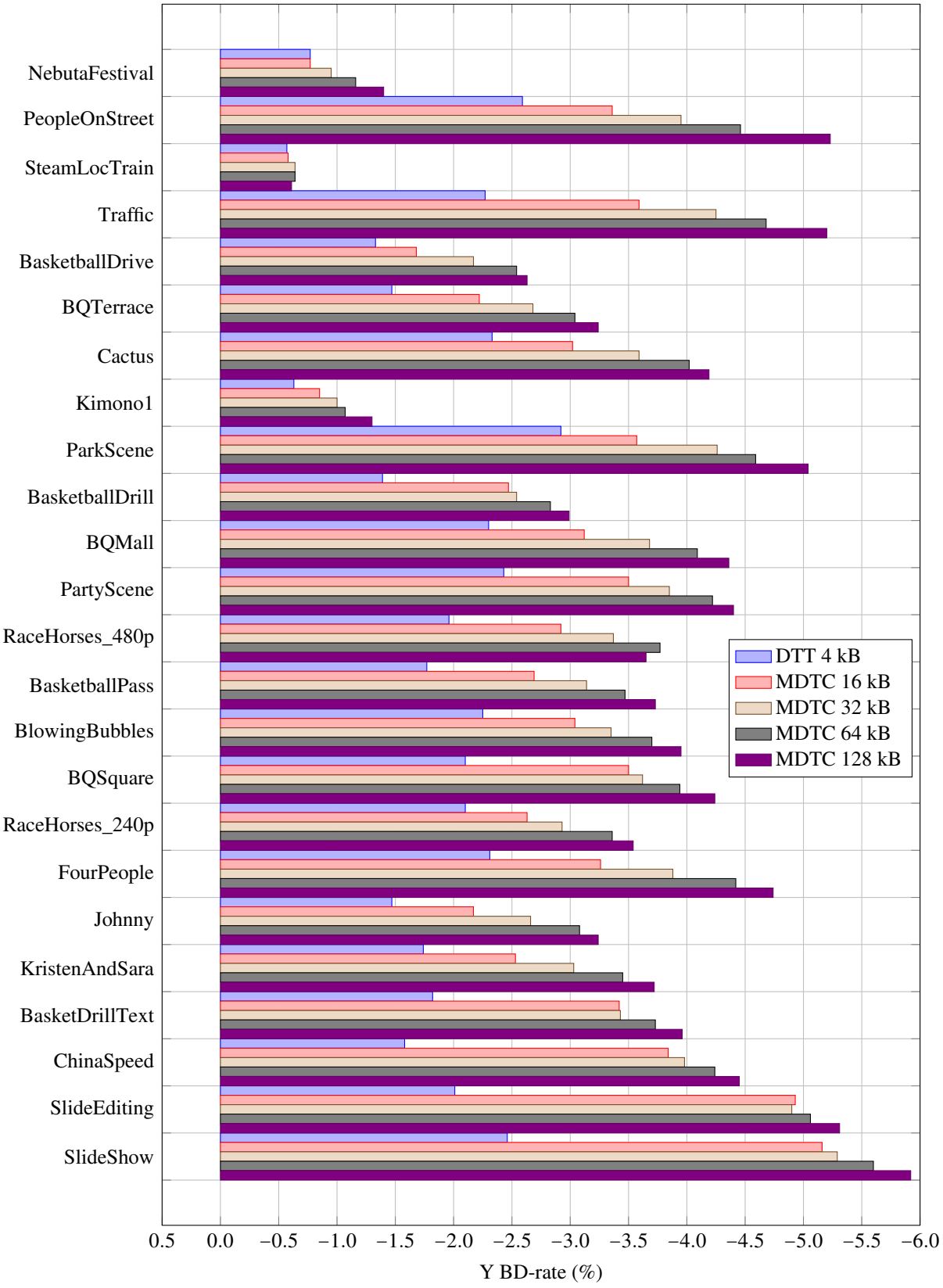


Figure 8.2.2 – Bar chart graphically summarising the retained MDTC systems in AI

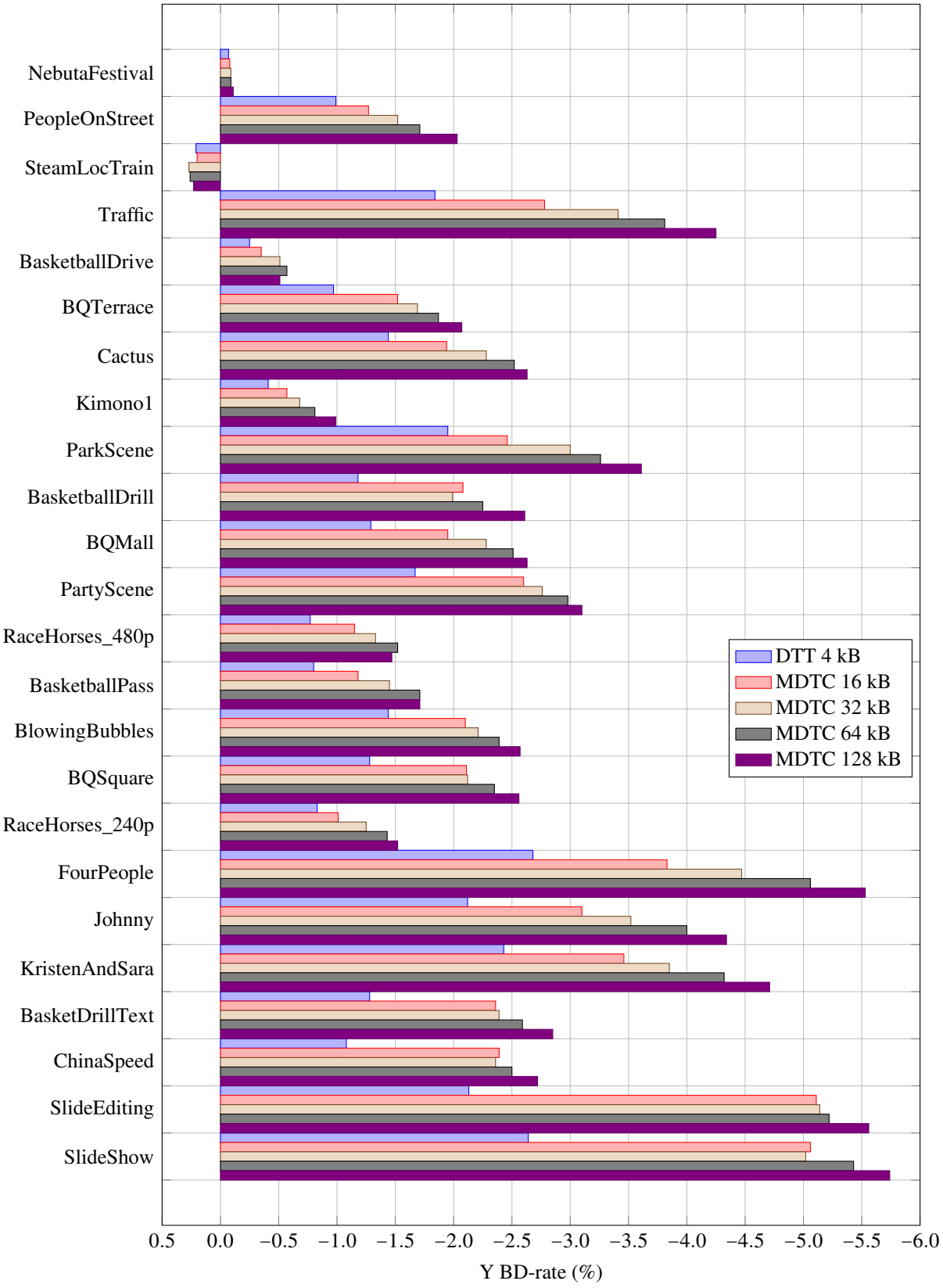


Figure 8.2.3 – Bar chart graphically summarising the retained MDTC systems in RA

		DTT	RDOT			
	Sequence	4 kB	16 kB	32 kB	64 kB	128 kB
Class A (2560×1600)	NebutaFestival	-0.07	-0.08	-0.09	-0.09	-0.11
	PeopleOnStreet	-0.99	-1.27	-1.52	-1.71	-2.03
	SteamLocomotiveTrain	0.21	0.20	0.27	0.26	0.23
	Traffic	-1.84	-2.78	-3.41	-3.81	-4.25
	Average	-0.67	-0.98	-1.19	-1.33	-1.54
Class B (1920×1080)	BasketballDrive	-0.25	-0.35	-0.51	-0.57	-0.51
	BQTerrace	-0.97	-1.52	-1.69	-1.87	-2.07
	Cactus	-1.44	-1.94	-2.28	-2.52	-2.63
	Kimono1	-0.41	-0.57	-0.68	-0.81	-0.99
	ParkScene	-1.95	-2.46	-3.00	-3.26	-3.61
	Average	-1.01	-1.37	-1.63	-1.80	-1.96
Class C (832×480)	BasketballDrill	-1.18	-2.08	-1.99	-2.25	-2.61
	BQMall	-1.29	-1.95	-2.28	-2.51	-2.63
	PartyScene	-1.67	-2.60	-2.76	-2.98	-3.10
	RaceHorses	-0.77	-1.15	-1.33	-1.52	-1.47
	Average	-1.23	-1.94	-2.09	-2.31	-2.45
Class D (416×240)	BasketballPass	-0.80	-1.18	-1.45	-1.71	-1.71
	BlowingBubbles	-1.44	-2.10	-2.21	-2.39	-2.57
	BQSquare	-1.28	-2.11	-2.12	-2.35	-2.56
	RaceHorses	-0.83	-1.01	-1.25	-1.43	-1.52
	Average	-1.09	-1.60	-1.76	-1.97	-2.09
Class E (1280×720)	FourPeople	-2.68	-3.83	-4.47	-5.06	-5.53
	Johnny	-2.12	-3.10	-3.52	-4.00	-4.34
	KristenAndSara	-2.43	-3.46	-3.85	-4.32	-4.71
	Average	-2.41	-3.46	-3.95	-4.46	-4.86
Class F (various resolutions)	BasketDrillText	-1.28	-2.36	-2.39	-2.59	-2.85
	ChinaSpeed	-1.08	-2.39	-2.36	-2.50	-2.72
	SlideEditing	-2.13	-5.11	-5.14	-5.22	-5.56
	SlideShow	-2.64	-5.06	-5.02	-5.43	-5.74
	Average	-1.78	-3.73	-3.73	-3.93	-4.22
All sequences	Overall	-1.31	-2.09	-2.29	-2.53	-2.73

Table 8.2.3 – Y BD-rate (%) for proposed MDTC systems in RA

8.3 Conclusions

This Chapter summarises and groups the most representative separable MDTC systems designed in this thesis and tests them against the HEVC test set in order to provide results for a well-known sequence set.

The retained systems prove the interest of using multiple transforms for video coding. Different levels of performances are provided depending on the allowed complexity on the encoder side.

The iterative design described in Chapter 6 has allowed creating a continuum of systems, satisfying different ROM requirements. Although the system design has been carried out for different ROM constraint, a system could be conceived for any point. This means that systems using more than 128 kB are possible and can provide higher bitrate savings.

Chapter 9

Conclusions and future work

Thesis objectives

The objective of this thesis is to provide a set of tools and systematic methods of designing multiple transforms to be used in video coders. In order to design performing learning algorithms, the RDOT metric, based on the ℓ_0 norm was selected. The appropriateness of this design method is justified over the traditional KLT in Chapter 2.

The methodology for learning adapted transforms based on the RDOT metric and presented in Chapters 2, 3 and 4 consists of the following steps:

1. Select a set of learning sequences.
2. Encode the sequences with HEVC using an AI configuration.
3. Extract the residuals (difference between original and predicted blocks) for the desired TU sizes, grouped by IPM.
4. Learn adapted set of transforms for each TU size and IPM using the RDOT metric from (2.3.2):
   (a) MDDT (Chapter 3).
   (b) MDTC (Chapter 4).
   (c) Incomplete transforms (Chapter 5).
   (d) DTTs (Chapter 7).
5. Simplify the system storage requirements with methods proposed in Chapter 6.

By having a look at the steps described above, one can see that is fairly straightforward to extend the usage of multiple transforms for video coding the following ways:

— Learning transforms for TU sizes other than 4×4 and 8×8, such as 16×16 and 32×32.

— Learning transforms for inter prediction residuals by encoding in RA instead of AI.

Conclusions

This thesis has proved that state-of-the-art video coding standards, such as HEVC, can be improved by designing multiple transforms adapted to video coding.

The design of the transforms is carried out off-line through a learning. Despite the fact that off-line learning techniques might have known issues such as over learning, if the learning set is chosen wisely (using several resolutions, frame rates and sources), the resulting transforms can be used outside the sequences of the learning set with good performances. This way, hardware implementations of the transforms are possible, since transforms coefficients are known.

Moreover, off-line learning has been possible thanks to the RDOT design introduced in [47]. The RDOT metric, based on the ℓ_0 norm, has served as a good approximation of the one implemented in RDO loop from HEVC.

The following points should be kept in mind:

— Every 10 years, a new video codec is standardised halving the bitrate required for an equivalent perceived quality.

— HEVC provided 23% of bitrate savings with regards H.264/MPEG-4 AVC in AI coding configuration with about twice the complexity [31].

— A video standard is the result of hard-work and negotiations between many engineers and companies during years.

With those points stated, the results in this thesis have proved the interest in using multiple transforms for video coding.

The first attempt is to compare the performances of the KLT against the RDOT through the MDDT technique. Improvements of over 1 BD-rate point are obtained when using the RDOT design, which takes into account the sparsity of the signal in the transform domain.

In order to put the use of multiple transforms further, the MDTC is born as an evolution of the MDDT where more than one transform is available in each IPM. Depending on the desired complexity and bitrate savings trade-off, this technique alone is able to obtain bitrate savings of 7%, for non-separable transforms, and around 4% for separable transforms.

However, to obtain those levels of performance, first obtained in Chapter 4, the encoding complexity and the storage requirements are too high for a commercial system. For this reason, Chapters 6 and 7 provide different systematic ways to reduce the storage requirements for the MDTC systems. As a side effect, the encoding complexity is also reduced.

Chapter 7 presents the DTT-based MDTC systems as a simplification of the MDTC systems used throughout this thesis. Despite the early stages of the work, DTTs have proved their interest by being a very competitive approach in terms of bitrate savings, complexity and storage requirements. However, the learning algorithm for these systems is known to be sub-optimal when using numerous DTTs. For this reason, improvements in the learning algorithm are essential to continue improving the DTT approach.

The last Chapter groups some of the most representative MDTC systems presented in this thesis. Thanks to the learning methods and simplifications presented in Chapter 6 and 7, a continuum of systems can be designed which are able to satisfy different complexity and bitrate savings trade-offs.

Perspectives for future work

Although encouraging results have been proved when using multiple transforms for video coding, there is still room for improvement in many areas of the techniques proposed that have been carried out in this thesis.

Learning set

In Chapters 3, 4 and 5, the test set that was used included in the learning set to be in the best possible conditions and validate the approach that the use of multiple transforms in video coding provide bitrate savings. Due to the promising results obtained in those Chapters, in order to make sure there was no over-learning, the learning set is replaced by a completely independent set of sequences in Chapter 6. The new learning set provides many more residuals from which to learn the transforms, but is less heterogeneous, since all frames come from the same sequence, which implies same frame rate, resolution, filters, etc.

Chapter 6 shows that there can be a small impact due to transform over-learning to the learning set. However, the over-learning effect impact could be lowered by choosing a more diversified learning set in terms of resolution, frame rate and contents. The choice of *Tears of Steel* short film was made to make a point: even with a learning set that is nothing like the test set, good results can be achieved. Moreover,

due to the fact that the short film is publicly available in raw format, makes the results reproducible by any party.

Therefore, using a learning set made up from sequences coming from different sources, at different resolutions, frame rates and contents can lead to better performing transforms.

Signalling

The current way of indicating the chosen transform to the decoder is via a very simplistic approach: during the RDO loop, the encoder tests all transforms in each TU and selects the one that provides the best trade-off in terms of rate-distortion. Then, a flag is used to indicate that a transform different from those of HEVC is used, and signalled with a fixed length codeword. This approach has the advantage of being very flexible, allowing a PU that has been split into four TUs to have different transforms.

However, if transforms were signalled one level upwards, that is, at a PU level instead of per TU, transform signalling would be reduced by a factor of 4, although all transforms in a PU would be identical. It seems reasonable to think that more bitrate savings could be achieved by using this new signalling scheme.

Another improvement in signalling might come by not using fixed length codewords, but favouring some transforms instead with shorter codewords. This approach will also allow using a number of transforms different from a power of 2. By signalling some transforms using fewer bits, they would be used more often than those with longer code words, leading to possible bitrate savings. Nevertheless, for this approach to work properly, the signalling cost of the transform should be taken into account during the learning step in order to minimise the mismatch between the learning and the testing phases.

Transform coefficients quantisation

All the transforms implemented in this thesis are integer versions of orthogonal transforms, whose coefficients are floating point numbers. As previously explained, each transform coefficient has been scaled and rounded to the nearest integer fitting into 1 byte range. This method is far from being optimal, and thus a thorough study on how transform coefficients should be quantised could lead to some improvements in terms of rate-distortion.

Coding complexity

The presented solutions to reduce the storage requirements for MDTC systems in Chapter 6 have reduced the encoding complexity as a side effect: smartly reducing the number of transforms to test on the encoder side leads to modest decreases in the encoding time, since less coding alternatives are explored.

However, no specific attention or effort has been put to reduce the encoding time of the proposed solutions. This is due to, in part, the fact that the encoder is not standardised, only the decoder, and focusing on the decoder time seemed more sensible to prove the viability of a new technique, such as the use of multiple transforms.

Currently, the proposed MDTC systems perform an exhaustive search to find the best combination of block size (PU and TU), IPM and transform. A possible way of simplifying the approach would be not to test the complete set of transforms by implementing a fast decision mode. Another way would be to remove the transform selection from the Rate-Distortion Optimised Quantisation (RDOQ) (see §8.1.7 from [56]) and select it by computing directly the RDOT metric value for all available transforms.

Further study of DTT-based MDTC systems

The DCT-based MDTC systems introduced in Chapter 7 have come late in the work presented in this thesis. Accordingly, these systems have not been studied with the same level of detail as the RDOT-based MDTC systems.

Despite the low level of maturity of the approach using DTTs, these systems have presented very promising trade-offs in the ROM — Y BD-rate plane. As a consequence, a thorough study on the DTT family, their properties and fast algorithm implementations might lead to improvements in the encoding and decoding complexities. Besides, the algorithm used to design the MDTC systems is sub-optimal, since the computational complexity of choosing the best set of N transforms amongst the 256 available unique 2D transforms is high, and grows with N. Improvements to the learning algorithm would lead to better choices regarding the transforms used in each IPM.

Appendix A

Publications

Conference papers

Non-separable mode-dependent transforms for intra coding in HEVC

IEEE VCIP 2014

Authors: Adrià Arrufat, Pierrick Philippe and Olivier Déforges

Abstract

Transform coding plays a crucial role in video coders. Recently, additional transforms based on the DST and the DCT have been included in the latest video coding standard, HEVC. Those transforms were introduced after a thoroughly analysis of the video signal properties. In this paper, we design additional transforms by using an alternative learning approach. The appropriateness of the design over the classical KLT learning is also shown. Subsequently, the additional designed transforms are applied to the latest HEVC scheme. Results show that coding performance is improved compared to the standard. Additional results show that the coding performance can be significantly further improved by using non-separable transforms. Bitrate reductions in the range of 2% over HEVC are achieved with those proposed transforms.

Note This publication received the *Best Paper Award* at IEEE VCIP 2014, which has held in Valletta, Malta in December 2014.

Rate-distortion optimised transform competition for intra coding in HEVC

Authors: Adrià Arrufat, Pierrick Philippe and Olivier Déforges

IEEE VCIP 2014

Abstract

State of the art video coders are based on prediction and transform coding. The transform decorrelates the signal to achieve high compression levels. In this paper we propose improving the performances of the latest video coding standard, HEVC, by adding a set of RDOTs. The transform design is based upon a cost function that incorporates a bit rate constraint. These new RDOTs compete against classical HEVC transforms in the RDO loop in the same way as prediction modes and block sizes, providing additional coding possibilities. Reductions in BD-rate of around 2% are demonstrated when making these transforms available in HEVC.

Mode-dependent transform competition in HEVC

IEEE ICIP 2015

Authors: Adrià Arrufat, Pierrick Philippe and Olivier Déforges

Abstract

Transform coding plays a key role in state-of-the-art video coders, such as HEVC. However, transforms used in current solutions do not cover the varieties of video coding signals. This work presents an adaptive transform design method that enables the use of multiple transforms in HEVC. A different transform set is learnt for each Intra Prediction Mode, allowing the video encoder to perform better decisions regarding block sizes, prediction modes and transforms. Different systems are proposed to accommodate trade-offs between complexity and performance. Bitrate reductions in the range of 2% to 7% are reported, depending on complexity.

Image coding with incomplete transforms for HEVC

IEEE ICIP 2015

Authors: Adrià Arrufat, Pierrick Philippe and Anne-Flore Perrin

Abstract

Overcomplete transforms have received considerable attention over the past years. However, they often suffer from a complexity burden. In this paper, a low complexity approach is provided, where an orthonormal basis is complemented with a set of incomplete transforms: those incomplete transforms include a reduced number of basis vectors that allow a reduction on the coding complexity and ensure a certain level of sparsity. The solution has been implemented in the HEVC standard and coding gains of around 1% on average are reported while reducing the decoder complexity in about 5%.

Low complexity transform competition for HEVC

IEEE ICASSP 2016

Authors: Adrià Arrufat, Pierrick Philippe, Kevin Reuzé and Olivier Déforges

Abstract

The use of multiple transforms in video coding can lead to substantial bitrate savings. However, these savings come at the expense of increased coding complexity and storage requirements, which challenge the usability of this approach. In this paper, a systematic procedure is proposed to design low complexity systems making use of transform competition. Multiple trade-offs accommodating the complexity are unveiled and it is demonstrated that they can keep a certain level of performance. Compared to the HEVC standard, some of them provide bitrate savings around 2% with a 50% increase in the encoding time, using less than 4 kB of extra ROM and no added decoding complexity.

Patent applications

1. Procédé de codage et de décodage d'images, dispositif de codage et de décodage d'images et programmes d'ordinateur correspondants, patent application INPI 1457768, August 2014
2. Procédé de codage et de décodage d'images, dispositif de codage et de décodage d'images et programmes d'ordinateur correspondants, patent application INPI 155394, April 2015.
3. Procédé de codage et de décodage d'images, dispositif de codage et de décodage d'images et programmes d'ordinateur correspondants, patent application INPI 1558066 August 2015
4. Procédé de codage et de décodage d'images, dispositif de codage et de décodage d'images et programmes d'ordinateur correspondants, patent pending, October 2015
5. Procédé de codage et de décodage d'images, dispositif de codage et de décodage d'images et programmes d'ordinateur correspondants, patent pending November 2015.

Appendix B

Transform usage in MDTC systems

This appendix contains tables describing used MDTC systems from Chapters 6 and 7. The *System* top row represents the system label and upper ROM boundary in kB.

Non-symmetrical MDTC systems

IPM \ System	1	2	4	8	12	16	24	32	48	64	96	128
0	2	2	8	16	16	16	32	32	32	32	32	32
1	2	2	2	8	8	8	8	8	8	8	8	8
2	0	0	1	1	1	1	1	2	2	1	4	4
3	0	0	0	1	1	1	2	2	4	2	8	8
4	0	0	0	0	0	0	1	1	4	1	4	8
5	0	0	1	1	1	1	1	4	4	1	4	4
6	0	0	0	1	1	2	2	4	4	2	8	8
7	0	0	0	2	2	2	2	2	4	2	8	8
8	0	0	2	2	2	2	2	2	2	2	2	4
9	1	1	1	1	1	1	1	1	1	1	4	4
10	1	1	2	4	4	4	8	8	8	8	8	16
11	0	0	1	1	2	2	2	2	2	2	4	4
12	1	1	1	1	1	1	1	1	1	1	2	2
13	1	1	1	1	1	1	1	1	2	1	4	4
14	0	0	1	1	1	1	2	2	8	2	8	8
15	0	0	0	0	0	0	1	1	4	1	8	8
16	0	0	0	0	0	0	0	1	2	0	8	8
17	0	0	1	2	2	2	2	2	2	2	4	4
18	0	0	0	0	1	1	2	2	2	2	4	4
19	0	0	0	0	0	0	0	1	2	0	2	4
20	0	0	1	1	1	1	1	2	2	2	4	4
21	0	0	0	1	2	2	2	8	16	4	32	32
22	0	0	0	0	0	0	2	2	2	2	4	4
23	0	0	0	0	0	0	2	2	2	2	4	4
24	0	0	1	1	2	2	2	2	2	2	2	2
25	0	0	0	1	1	1	1	2	2	2	4	4
26	1	1	1	4	8	8	16	16	16	16	16	32
27	0	1	2	2	2	2	4	8	8	4	8	8
28	0	0	0	1	2	2	2	4	4	2	8	8
29	0	0	0	1	2	2	2	2	2	2	2	2
30	0	0	0	0	0	0	0	0	0	0	4	4
31	0	0	1	1	1	1	1	1	1	1	8	8
32	0	0	1	2	2	2	4	4	4	4	4	8
33	0	0	0	0	0	0	0	0	4	0	4	4
34	0	0	0	0	0	0	1	2	4	1	4	4

Table B.1 – Transform repartition for non-symmetrical MDTC systems on 4×4 TUs

IPM \ System	1	2	4	8	12	16	24	32	48	64	96	128
0	1	4	4	4	4	8	32	32	64	128	128	128
1	0	0	1	4	8	16	16	32	64	64	128	128
2	0	0	0	0	0	0	0	0	1	1	1	4
3	0	0	0	0	0	0	0	0	1	1	1	2
4	0	0	0	1	1	1	1	1	1	1	2	8
5	0	0	0	0	0	1	1	1	1	2	4	16
6	0	0	0	1	1	1	2	2	2	2	2	8
7	0	0	0	0	0	0	0	1	1	1	4	4
8	0	0	1	1	1	1	1	2	4	8	8	8
9	0	0	1	1	2	2	2	2	4	8	8	16
10	1	2	2	4	4	4	8	16	16	32	32	64
11	0	0	1	1	2	2	2	4	8	8	8	16
12	0	0	0	0	2	2	2	2	2	2	2	4
13	0	0	0	1	1	2	2	2	2	2	2	4
14	0	0	0	1	1	1	1	1	1	2	2	4
15	0	0	0	0	0	1	1	1	1	1	4	16
16	0	0	0	0	0	0	0	1	1	1	1	2
17	0	0	0	0	0	0	1	1	1	2	2	4
18	0	0	0	0	1	1	2	2	2	2	2	4
19	0	0	0	0	0	0	0	0	0	0	0	2
20	0	0	0	0	0	1	1	1	1	1	2	4
21	0	0	0	1	1	2	2	2	2	2	2	2
22	0	0	0	0	1	1	1	1	1	1	2	8
23	0	0	0	0	1	1	2	2	2	2	2	2
24	0	0	0	1	2	2	2	2	2	2	2	8
25	0	0	1	1	2	2	2	2	2	2	4	4
26	1	2	2	4	8	8	8	8	8	8	64	64
27	0	0	1	1	2	4	4	8	8	16	16	32
28	0	0	0	1	1	1	1	1	1	2	4	4
29	0	0	0	0	1	1	1	1	1	1	1	4
30	0	0	0	0	0	0	1	2	2	2	2	2
31	0	0	0	0	0	0	1	1	1	1	1	2
32	0	0	0	0	0	0	0	1	2	2	2	8
33	0	0	0	0	0	0	0	1	1	2	2	8
34	0	0	0	0	0	0	0	0	0	0	2	16

Table B.2 – Transform repartition for non-symmetrical MDTC systems on 8×8 TUs

Symmetrical MDTC systems

IPM \ System	1	2	4	8	12	16	24	32	48	64	96	128
0	0	2	2	2	2	16	16	16	32	16	32	32
1	0	2	4	8	8	8	8	8	32	32	32	32
2	0	0	0	4	4	4	4	4	8	8	8	8
3	0	1	1	1	1	8	8	8	8	8	16	16
4	0	1	1	2	2	8	8	8	32	16	32	32
5	1	2	2	2	2	4	4	4	16	4	16	16
6	0	0	0	0	0	4	4	4	4	4	4	4
7	0	1	1	2	2	2	2	2	8	4	8	8
8	1	1	1	1	1	1	1	1	4	4	4	4
9	1	1	1	2	2	2	2	2	16	2	16	16
10	2	2	4	16	16	32	32	32	32	32	32	32

Table B.3 – Transform repartition for symmetrical MDTC systems on 4×4 TUs

IPM \ System	1	2	4	8	12	16	24	32	48	64	96	128
0	1	2	4	4	16	16	32	32	32	32	128	128
1	0	0	1	4	8	8	16	32	32	64	64	128
2	0	0	0	0	2	2	2	2	8	16	16	32
3	0	0	0	2	4	4	4	4	16	16	16	32
4	0	0	1	2	2	2	2	4	4	4	16	16
5	0	0	0	2	2	2	2	4	8	16	16	32
6	0	0	1	2	4	4	4	4	8	8	8	32
7	0	0	2	2	2	2	2	2	4	8	16	64
8	0	1	2	2	2	2	8	16	16	16	16	16
9	2	2	4	4	4	4	16	16	16	64	64	64
10	1	2	2	8	8	16	16	32	64	64	64	64

Table B.4 – Transform repartition for symmetrical MDTC systems on 8×8 TUs

Symmetrical DTT-based MDTC systems

	4×4			8×8		
IPM \ System	1	2	4	1	2	4
0	8	8	8	2	4	16
1	2	4	4	2	4	16
2	0	0	0	0	2	4
3	4	4	4	1	1	2
4	2	2	2	0	1	4
5	2	2	2	1	2	4
6	0	1	1	1	1	1
7	2	2	2	0	0	0
8	2	2	2	0	0	0
9	1	1	1	0	0	0
10	4	8	8	2	8	8

Table B.5 – Transform repartition for symmetrical DTT-based MDTC systems on 4×4 and 8×8 TUs

Bibliography

[1] M. Aharon, M. Elad, and A. Bruckstein. K-SVD: An algorithm for designing overcomplete dictionaries for sparse representation. *Signal Processing, IEEE Transactions on*, 54(11):4311–4322, Nov 2006.

[2] A. Akansu and M. Torun. Toeplitz approximation to empirical correlation matrix of asset returns: A signal processing perspective. *Selected Topics in Signal Processing, IEEE Journal of*, 6(4):319–326, Aug 2012.

[3] 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*, pages 61–64, Dec 2014.

[4] A. Arrufat, P. Philippe, and O. Déforges. Rate-distortion optimised transform competition for intra coding in HEVC. In *Visual Communications and Image Processing Conference, 2014 IEEE*, pages 73–76, Dec 2014.

[5] A. Arrufat, P. Philippe, and O. Déforges. Mode-dependent transform competition in HEVC. In *Image Processing, 2015. ICIP 2015. 22th IEEE International Conference on*, pages 149–152, 2015.

[6] A. Arrufat, P. Philippe, and A. F. Perrin. Image coding with incomplete transform competition in HEVC. In *Image Processing, 2015. ICIP 2015. 22th IEEE International Conference on*, pages 159–162, 2015.

[7] A. Arrufat, P. Philippe, K. Reuzé, and O. Déforges. Low complexity transform competition for HEVC. In *Acoustics Speech and Signal Processing (ICASSP), 2016 IEEE International Conference on*, 2016.

[8] G. Bjøntegaard. Calculation of average PSNR differences between RD-curves. Technical Report VCEG-M33, ITU-T SG16/Q6 VCEG, Austin, Texas, April 2001.

[9] G. Bjøntegaard. Improvements of the BD-PSNR model. Technical Report VCEG-AI11, ITU-T SG16/Q6 VCEG, Berlin, Germany, July 2008.

[10] Tears of Steel, 2014. `http://www.tearsofsteel.org`.

[11] F. Bossen. Common test conditions and software reference configurations. Technical Report JCT-VC I1100, ITU-T, Geneva, Switzerland, May 2012.

[12] V. Britanak, P. Yip, and K. R. Rao. *Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations*. Academic Press, 2004.

[13] Cisco-Systems. VNI forecast highlights. `http://www.cisco.com/web/solutions/sp/vni/vni_forecast_highlights`, 2014. [Accessed 3rd March 2015].

[14] C. Cutler. Differential quantization of communication signals, July 29 1952. US Patent 2,605,361.

[15] M. Effros, H. Feng, and K. Zeger. Suboptimality of the Karhunen-Loève Transform for transform coding. *Information Theory, IEEE Transactions on*, 50(8):1605–1619, Aug 2004.

[16] F. Fernandes. Low Complexity Rotational Transform. Technical Report JCT-VC C096, ITU-T, Guangzhou, China, October 2010.

[17] A. Gersho and R. Gray. *Vector Quantization and Signal Compression*. Kluwer international series in engineering and computer science: Communications and information theory. Springer US, 1992.

[18] V. Goyal. High-rate transform coding: how high is high, and does it matter? In *Information Theory, 2000. Proceedings. IEEE International Symposium on*, pages 207–, 2000.

[19] L. Guo, M. Karczewicz, and J. Chen. Transform for inter-layer prediction residues in scalable video coding. In *Multimedia and Expo Workshops (ICMEW), 2013 IEEE International Conference on*, pages 1–6, July 2013.

[20] J. Han, A. Saxena, and K. Rose. Towards jointly optimal spatial prediction and adaptive transform in video/image coding. In *Acoustics Speech and Signal Processing (ICASSP), 2010 IEEE International Conference on*, pages 726–729, 2010.

[21] K. Huang and S. Aviyente. Sparse representation for signal classification. In *In Adv. NIPS*, 2006.

[22] ITU-T. Recommendation H.264: Advanced video coding, May 2003.

[23] ITU-T. Recommendation H.265: High efficiency video coding, January 2013.

[24] A. Jain. Image Coding Via a Nearest Neighbors Image Model. *Communications, IEEE Transactions on*, 23(3):318–331, Mar 1975.

[25] A. Jain. A Fast Karhunen-Loève Transform for a Class of Random Processes. *Communications, IEEE Transactions on*, 24(9):1023–1029, Sep 1976.

[26] N. S. Jayant and P. Noll. *Digital coding of waveforms*. Prentice Hall, 1984.

[27] JCT-VC. Test Model under Consideration. Technical Report JCT-VC A204, ITU-T, Dresden, Germany, April 2010.

[28] Joint Photographic Experts Group. JPEG image coding system, iso/iec 10918, itu-t t.81, September 1992.

[29] Joint Photographic Experts Group. JPEG 2000 image coding system, iso/iec 15444-1:2000, December 2000.

[30] E. Lam and J. Goodman. A mathematical analysis of the DCT coefficient distributions for images. *Image Processing, IEEE Transactions on*, 9(10):1661–1666, Oct 2000.

[31] B. Li, G. Sullivan, and J. Xu. Comparison of Compression Performance of HEVC Draft 10 with AVC High Profile. Technical Report JCT-VC M0329, ITU-T, Incheon, South Korea, April 2014.

[32] J. Li, M. Gabbouj, J. Takala, and H. Chen. Laplacian modeling of dct coefficients for real-time encoding. In *Multimedia and Expo, 2008 IEEE International Conference on*, pages 797–800, June 2008.

[33] 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*, 7(6):990–1000, Dec 2013.

[34] M. Mrak, A. Gabriellini, S. N., and D. Flynn. Transform skip mode. Technical Report JCT-VC F077, ITU-T, Torino, Italy, July 2011.

[35] M. Naccari, A. Gabriellini, and M. M. Quantization for transform skipping. Technical Report JCT-VC H0208, ITU-T, San José, CA, USA, February 2012.

[36] C. A. Poynton. A guided tour of color space. In *SMPTE Advanced Television and Electronic Imaging Conference*, pages 167–180, Feb 2005.

[37] M. Puschel and J. Moura. Algebraic Signal Processing Theory: Cooley–Tukey Type Algorithms for DCTs and DSTs. *Signal Processing, IEEE Transactions on*, 56(4):1502–1521, April 2008.

[38] K. R. Rao and P. Yip. *The Transform and Data Compression Handbook*. Boca Raton, 2001.

[39] Y. Reznik. Relationship between dct-ii, dct-vi, and dst-vii transforms. In *Acoustics, Speech and Signal Processing (ICASSP), 2013 IEEE International Conference on*, pages 5642–5646, May 2013.

[40] A. Saxena and F. Fernandes. Jointly optimal intra prediction and adapted primary transform. Technical Report JCT-VC C108, ITU-T, Guangzhou, China, October 2010.

[41] A. Saxena and F. Fernandes. Mode-dependent DCT/DST for intra prediction in video coding. Technical Report JCT-VC D033, ITU-T, Daegu, Korea, January 2011.

[42] A. Saxena and F. Fernandes. Mode-dependent DCT/DST without 4*4 full matrix multiplication for intra prediction. Technical Report JCT-VC E125, ITU-T, Geneva, Switzerland, March 2011.

[43] A. Saxena and F. Fernandes. On fast implementation of 4-point DST Type-7 with 5 multiplications. Technical Report JCT-VC F283, ITU-T, Torino, Italy, July 2011.

[44] A. Saxena, F. Fernandes, and Y. Reznik. Fast transforms for intra-prediction-based image and video coding. In *Data Compression Conference (DCC), 2013*, pages 13–22, March 2013.

[45] A. Saxena, Y. Shibahara, F. Fernandes, and T.Nishi. On secondary transforms for intra prediction residual. Technical Report JCT-VC G108, ITU-T, Geneva, Switzerland, November 2011.

[46] O. Sezer. *Data-driven transform optimization for next generation multimedia applications*. PhD thesis, Georgia Institute of Technology, 2011.

[47] O. Sezer, O. Harmanci, and O. Guleryuz. Sparse orthonormal transforms for image compression. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 149–152, 2008.

[48] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas. Transform coefficient coding in HEVC. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1765–1777, 2012.

[49] 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*, pages 9–12, Nov 2009.

[50] G. Sullivan, J. Ohm, W. J. Han, and T. Wiegand. Overview of the High Efficiency Video Coding standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1649–1668, 2012.

[51] G. Sullivan and T. Wiegand. Rate-distortion optimization for video compression. *Signal Processing Magazine, IEEE*, 15(6):74–90, Nov 1998.

[52] K. Ugur and A. Saxena. Summary report of Core Experiment on intra transform mode dependency simplifications. Technical Report JCT-VC J0021, ITU-T, Stockholm, Sweden, July 2012.

[53] Z. Wang, A. Bovik, H. Sheikh, and E. Simoncelli. Image quality assessment: from error visibility to structural similarity. *Image Processing, IEEE Transactions on*, 13(4):600–612, April 2004.

[54] T. Wedi and T. Tan. AGH report - Coding Efficiency Improvements. Technical Report VCEG-AI11, ITU-T SG16/Q6 VCEG, Nice, France, October 2005.

[55] C. Wei-Jun, J. Sole, and M. Karczewicz. Last position coding for CABAC. Technical Report JCT-VC G704, ITU-T, Geneva, Switzerland, November 2011.

[56] M. Wien. *High Efficiency Video Coding – Coding Tools and Specification*. Springer, 2015.

[57] J. Xu, F. Wu, and W. Zhang. Intra-predictive transforms for block-based image coding. *Signal Processing, IEEE Transactions on*, 57(8):3030–3040, Aug 2009.

[58] M. Yang. Matrix decomposition. Technical report, Northwestern University, Evanston, Illinois, 2004.

[59] Y. Ye and M. Karczewicz. Improved intra coding. Technical Report VCEG-AG11, ITU-T SG16/Q6 VCEG, Shenzhen, China, October 2007.

[60] 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*, pages 2116–2119, 2008.

[61] C. Yeo, Y. H. Tan, and Z. Li. Low-complexity 4-point Integer Discrete Sine Transform. Technical Report JCT-VC D048, ITU-T, Daegu, Korea, January 2011.

[62] C. Yeo, Y. H. Tan, Z. Li, and S. Rahardja. Mode-Dependent Fast Separable KLT for Block-based Intra Coding. Technical Report JCT-VC B024, ITU-T, Geneva, Switzerland, July 2010.

[63] G. Yovanof and S. Liu. Statistical analysis of the dct coefficients and their quantization error. In *Signals, Systems and Computers, 1996. Conference Record of the Thirtieth Asilomar Conference on*, volume 1, pages 601–605 vol.1, Nov 1996.

[64] W.-C. Yueh. Eigenvalues of several tridiagonal matrices. *Applied Mathematics E-Notes*, 5(66-74):210–230, 2005.

[65] F. Zou, O. Au, C. Pang, J. Dai, X. Zhang, and L. Fang. Rate-distortion optimized transforms based on the Lloyd-type algorithm for intra block coding. *Selected Topics in Signal Processing, IEEE Journal of*, 7(6):1072–1083, 2013.

Index

INSA de RENNES
Service des Formations

AVIS DU JURY SUR LA REPRODUCTION DE LA THESE SOUTENUE

Titre de la thèse:
Transformées multiples pour la compression vidéo

Nom Prénom de l'auteur : ARRUFAT BATALLLA ADRIA

Membres du jury :
- Madame GUILLEMOT Christine
- Monsieur PEREIRA Fernando
- Monsieur SALEMBIER Philippe
- Monsieur WIEN Mathias
- Monsieur DEFORGES Olivier
- Monsieur PHILIPPE Pierrick
- Madame PESQUET-POPESCU Béatrice

Président du jury : C. Guillemot

Date de la soutenance : 11 Décembre 2015

Reproduction de la these soutenue

Thèse pouvant être reproduite en l'état
~~Thèse pouvant être reproduite après corrections suggérées~~

Fait à Rennes, le 11 Décembre 2015

Signature du président de jury

Le Directeur,

M'hamed DRISSI

Résumé

Les codeurs vidéo état de l'art utilisent des transformées pour assurer une représentation compacte du signal. L'étape de transformation constitue le domaine dans lequel s'effectue la compression, pourtant peu de variabilité dans les types de transformations est constatée dans les systèmes de codage vidéo normalisés : souvent, une seule transformée est considérée, habituellement la transformée en cosinus discrète (DCT).

Récemment, d'autres transformées ont commencé à être considérées en complément de la DCT. Par exemple, dans le dernier standard de compression vidéo, nommé HEVC (High Efficiency Video Coding), les blocs de taille 4x4 peuvent utiliser la transformée en sinus discrète (DST), de plus, il est également possible de ne pas les transformer. Ceci révèle un intérêt croissant pour considérer une pluralité de transformées afin d'augmenter les taux de compression.

Cette thèse se concentre sur l'extension de HEVC au travers de l'utilisation de multiples transformées. Après une introduction générale au codage vidéo et au codage par transformée, une étude détaillée de deux méthodes de construction de transformations est menée : la transformée de Karhunen Loève (KLT) et une transformée optimisée en débit et distorsion sont considérées. Ces deux méthodes sont comparées entre-elles en substituant les transformées utilisées par HEVC. Une expérimentation valide la pertinence des approches.

Un schéma de codage qui incorpore et augmente l'utilisation de multiples transformées est alors introduit : plusieurs transformées sont mises à disposition de l'encodeur, qui sélectionne celle qui apporte le meilleur compromis dans le plan débit distorsion. Pour ce faire, une méthode de construction qui permet de concevoir des systèmes comportant de multiples transformations est décrite. Avec ce schéma de codage, le débit est significativement réduit par rapport à HEVC, tout particulièrement lorsque les transformées sont nombreuses et complexes à mettre en œuvre. Néanmoins, ces améliorations viennent au prix d'une complexité accrue en termes d'encodage, de décodage et de contrainte de stockage. En conséquence, des simplifications sont considérées dans la suite du document, qui ont vocation à limiter l'impact en réduction de débit.

Une première approche est introduite dans laquelle des transformées incomplètes sont motivées. Les transformations de ce type utilisent un seul vecteur de base, et sont conçues pour travailler de concert avec les transformations de HEVC. Cette technique est évaluée et apporte une réduction de complexité significative par rapport au précédent système, bien que la réduction de débit soit modeste.
Une méthode systématique, qui détermine les meilleurs compromis entre le nombre de transformées et l'économie de débit est alors définie. Cette méthode utilise deux types différents de transformée : basés sur des transformées orthogonales séparables et des transformées trigonométriques discrètes (DTT) en particulier. Plusieurs points d'opération sont présentés qui illustrent plusieurs compromis complexité / gain en débit. Ces systèmes révèlent l'intérêt de l'utilisation de transformations multiples pour le codage vidéo.

Abstract

State of the art video codecs use transforms to ensure a compact signal representation. The transform stage is where compression takes place, however, little variety is observed in the type of transforms used for standardised video coding schemes: often, a single transform is considered, usually a Discrete Cosine Transform (DCT).

Recently, other transforms have started being considered in addition to the DCT. For instance, in the latest video coding standard, High Efficiency Video Coding (HEVC), the 4x4 sized blocks can make use of the Discrete Sine Transform (DST) and, in addition, it also possible not to transform them. This fact reveals an increasing interest to consider a plurality of transforms to achieve higher compression rates.

This thesis focuses on extending HEVC through the use of multiple transforms. After a general introduction to video compression and transform coding, two transform designs are studied in detail: the Karhunen Loève Transform (KLT) and a Rate-Distortion Optimised Transform are considered. These two methods are compared against each other by replacing the transforms in HEVC. This experiment validates the appropriateness of the design.

A coding scheme that incorporates and boosts the use of multiple transforms is introduced: several transforms are made available to the encoder, which chooses the one that provides the best rate-distortion trade-off. Consequently, a design method for building systems using multiple transforms is also described. With this coding scheme, significant amounts of bit-rate savings are achieved over HEVC, especially when using many complex transforms. However, these improvements come at the expense of increased complexity in terms of coding, decoding and storage requirements. As a result, simplifications are considered while limiting the impact on bit-rate savings.

A first approach is introduced, in which incomplete transforms are used. This kind of transforms use one single base vector and are conceived to work as companions of the HEVC transforms. This technique is evaluated and provides significant complexity reductions over the previous system, although the bit-rate savings are modest.

A systematic method, which specifically determines the best trade-offs between the number of transforms and bit-rate savings, is designed. This method uses two different types of transforms based separable orthogonal transforms and Discrete Trigonometric Transforms (DTTs) in particular.
Several designs are presented, allowing for different complexity and bit-rate savings trade-offs. These systems reveal the interest of using multiple transforms for video coding.

N° d'ordre : **15ISAR 25 / D15 - 25**

Institut National des Sciences Appliquées de Rennes
20, Avenue des Buttes de Coëmes • CS 70839 • F-35708 Rennes Cedex
Tel : 02 23 23 82 00 - Fax : 02 23 23 83 96