ADVANCES ON TRANSFORMS FOR HIGH EFFICIENCY VIDEO CODING

Miguel Capelo
Instituto Superior Técnico, Av. Rovisco Pais, 1049-001 Lisboa, Portugal
miguel.capelo@ist.utl.pt

ABSTRACT

The increasing presence of high and ultra high definition video contents in several multimedia applications has led to the necessity of developing a new coding standard that can achieve further compression efficiency regarding the H.264/AVC state-of-the-art. As an answer to this need, ITU-T VCEG and ISO/IEC MPEG started a new standardization project called High Efficiency Video Coding (HEVC). In this context, a transform coding technique which can adaptively select the Discrete Cosine Transform (DCT) or a modified Karhunen-Loève Transform (KLT) is presented in this work. This modified KLT which is implemented in both the encoder and the decoder uses a similar technique, avoiding coding them as required in the usual KLT. To evaluate the compression efficiency of this adaptive transform, it is combined with the HEVC codec. These experiments show that the adopted transform coding solution can achieve bitrate savings of up to 16% over the usual DCT.

Index Terms— Transform coding, DCT, KLT, adaptive transform, HEVC standard.

1. INTRODUCTION

Digital video has been a regular presence in our lives for many years now. Whether used for digital television, in personal computers, hand held devices or other multimedia applications, its use has grown tremendously in the last years and it seems that this growth is not slowing down. With the currently available transmission and storage supports, this growth is only possible with the use of powerful compression tools allowing the reduction of the number of bits needed to represent the video content by exploiting the data correlation and the limitations of the Human Visual System (HVS) to remove the redundant and irrelevant data, respectively. These compression tools have been included in several video coding standards defined by the International Telecommunication Union – Telecommunication Standardization Sector (ITU-T) and the Moving Picture Experts Group (MPEG) over the last two decades. Currently, the H.264/AVC [1] coding standard, developed by the Joint Video Team (JVT) formed by the ITU-T VCEG and ISO/IEC MPEG bodies, is considered the state-of-the-art in terms of video coding. However, with the recent advances in video capturing and display technologies, the presence of High Definition (HD) and Ultra High Definition (UHD) video contents in various multimedia applications is quickly increasing. Clearly, this type of video resolutions requires higher bandwidth for its transmission and larger storage capacities. In this way, the compression ratios achieved by the current state-of-the-art video coding standard for HD and UHD content do not seem enough taking in account the available transmission and storage supports. With this in mind, the ITU-T VCEG and ISO/IEC MPEG bodies created the Joint Collaborative Team on Video Coding (JCT-VC) which is currently developing a new video coding standard, the High Efficiency Video Coding (HEVC) [2] standard, with the objective of increasing the highest available compression ratios, particularly for very high resolution video contents. To do this, new coding techniques have to be developed that can guarantee better compression over the current ones even if at the price of some additional complexity. In this context, this work focuses on the design, implementation and assessment of a novel coding technique for a particular data compression module: the transform coding. Transform coding is used since the first image and video coding standards and it is still present in the current state-of-the-art video coding standard. The main objective of transform coding is to remove the spatial redundancy present in a particular image or video frame by transforming it from the spatial to the frequency domains. Since the HVS is less sensible to the higher frequencies than to the lower frequencies, this may also be an effective way to discard irrelevant data contained in the higher frequency bands. Currently, all the available video coding standards make use of the Discrete Cosine Transform (DCT) [3], but in this work the adaptive transform (AT) introduced by Biswas et al. in 2010 [4] is studied in the context of the high resolution video content.

The remainder of this paper is organized as follows: Section 2 briefly reviews the basic principles and concepts on transform coding, focusing on the two transforms used in the adopted coding solution (the DCT and the KLT). Section 3 introduces the two main background technologies of the adopted coding solution. Thus, the details of the adaptive transform proposed by Biswas et al. in [4] are first presented, followed by the introduction of the main technical novelties of the currently under development HEVC standard. After this, the adopted solution architecture is presented in Section 4, followed by a functional description of its coding processes. In Section 5 the experimental results obtained with the adopted coding solution are presented and analyzed. Finally, Section 6 concludes the paper and discusses future research directions.

2. BASICS ON TRANSFORM CODING

Transform coding is one of the basic tools used in digital compression, notably image, video and also audio data. In image and video compression, the transforms are mainly used to reduce the spatial redundancy by representing the pixels in a frequency domain prior to data reduction through compaction and quantization. To achieve data compression, the original signal is decorrelated by using an appropriate transform, redistributing its energy to a typically small number of transform coefficients, usually located in the low frequency region. These coefficients can then be quantized with the aim of discarding perceptually irrelevant information, without significantly affecting the subjective quality of the reconstructed/decoded image and video. Although the transform process does not theoretically involve data losses, the closely associated quantization process is lossy, since the original values cannot be
recovered due to the associated quantization error. From the compression point of view, an 'ideal' transform should have the following characteristics:

- **Reversibility** – A transform is reversible if the input signal can be recovered in its original domain after applying the transform and its associated inverse transform without error (if no numerical constraints exist). In image and video compression, this is an essential feature since the original data has to be recovered in the spatial domain to be visualized.

- **Energy compaction** – Energy compaction regards the capability to reduce the number of energy elements without any loss of information by removing existing redundancy. This means that the ideal transform must concentrate the original signal energy in the smallest number of coefficients possible.

- **Decorrelation** – Decorrelated coefficients are coefficients that do not transmit the same information; this assures that each coefficient carries additional information with no or small repetition and, thus, it always adds value by itself.

- **Data-independent** – A data-independent transform is a transform that is independent of the input signal; ideally, the transform should achieve good compression efficiency for most image types. While it is natural that the optimal transform depends on the input signal properties, the computational complexity to find this optimal transform and the overhead required to transmit it to the decoder is not typically practicable and desirable.

- **Low complexity** – The complexity of a transform is related with the computational resources required to perform it, e.g., the number of operations required; it is naturally desirable that a transform be performed with the lowest possible computational complexity and this may require the development of fast transform implementations.

The Kahunen-Loève Transform (KLT) [5] is considered to be the best available transform in terms of energy compaction and decorrelation performance. But it is not data-independent, requiring the computation of its basis functions. To compute its basis functions, first the covariance matrix of the input signal has to be determined to obtain its eigenvectors. Then, the KLT basis functions can be obtained by simply transposing the matrix of eigenvectors. This comes as a major drawback, since, besides their computation (which by itself increases significantly this transform computation complexity), the KLT basis functions also have to be coded in order to be available on the decoder side. In this way, all the available video coding standards make use of the DCT to decorrelate the input video signals. This transform cannot achieve the energy compaction performance of the KLT, but for highly correlated signals comes very close. Additionally, the DCT is data-independent, avoiding the computation of the basis functions, along with its storage and transmission.

### 3.1. An Adaptive Transform for Improved H.264/AVC-Based Video Coding

In [4], Biswas et al. propose a coding solution using an adaptive transform which allows a dynamic selection between the DCT and a modified KLT (MKLT), depending on the block content. This solution does not require coding and transmitting the MKLT basis functions. Instead, they are estimated in both the encoder and decoder using the same technique, thus assuring equivalent transform basis at both ends of the coding chain. In this way, it is possible to exploit the optimal behavior of the KLT, particularly for blocks which are hard to code using the DCT (e.g., blocks with diagonal edges). As the proposed KLT-based technique is only applicable to prediction error blocks, this adaptive transform solution can only bring compression improvements for inter-coded blocks.

To compute the MKLT basis functions, the prediction error is estimated and then the basis functions are computed using a similar method to the one used in the KLT. To estimate the prediction error, Biswas et al. [4] assume that the actual prediction error is caused by errors in the motion estimation process, particularly:

- **Interpolation errors** – In the motion compensation process, some errors can occur when interpolating the reference frame pixels for quarter-pixel accuracy.

- **Imprecise edges prediction** – In blocks with strong diagonal edges, the motion vectors may not be predicted with full accuracy, thus causing small shifts in the location of the edges between the original and the Motion Compensated Prediction (MCP) block.

With this in mind, Biswas et al. [4] propose the simulation of these conditions to estimate the prediction error. This is done by subtracting shifted and rotated versions of the MCP block from the MCP block itself which plays here the role of the ‘original’ data. The use of the MCP block for this purpose is natural as it is the only piece of information that is simultaneously available at both the encoder and decoder.

By shifting the MCP block horizontally and vertically by -0.5, -0.25, 0.0, 0.25 and 0.5 pixels and rotating it by -0.5°, 0.0° and 0.5° results into a set of 75 shifted and rotated MCP blocks. Then, the difference between the actual MCP block and the set of shifted and rotated MCP blocks is computed in order to obtain a set of 75 estimated prediction error blocks. With these set of estimated prediction error blocks, is then possible to compute its covariance matrix. In this way, the covariance between a pixel in position (u,v) and a pixel in position (r,s) for a set of N×N blocks is given by

\[
\sigma^2_{jk} = \frac{1}{N^2} \sum_{i=1}^{N} (E_i(u,v) - \bar{E})(E_i(r,s) - \bar{E})
\]

where \(u, v, r, s = 0,...,(N-1)\), \(j = u+n,v\), \(k = r+n,s\), \(E_i(u,v)\) is the estimated prediction error in position (u,v) of the i-th block and N is the number of blocks in the set. With the covariance matrix for a particular set of estimated prediction error blocks available, it is then possible to determine the associated eigenvectors and eigenvalues and then the MKLT basis functions can be obtained by transposing the eigenvectors matrix, as done in the usual KLT. After the determination of the MKLT basis functions, it is then possible to actually compute the MKLT both at the encoder and decoder.

The integration of this adaptive transform in the H.264/AVC codec this video coding solution can achieve a significant improvement in terms of compression performance when compared to the actual state-of-the-art video coding standard, the H.264/AVC codec, with
PSNR improvements and bitrate savings of up to 0.9 dB and 20%, respectively [4].

3.2. Introduction to the High Efficiency Video Coding Standard

As referred before, the JCT-VC is currently developing a new video coding standard, called High Efficiency Video Coding (HEVC) standard, with the purpose of replacing H.264/AVC as the state-of-the-art video coding standard. Moreover, this emerging standard is designed taking into account that some main emerging applications will use both high and ultra high definition video contents. Thus, the HEVC standard introduces a new image partitioning scheme based on a novel coding unit definition and not anymore the usual macroblocks. The previous macroblock concept is replaced by a more flexible structure comprised by Coding Tree Blocks (CTB). With this structure, each CTB can have various sizes (from 8x8 to 128x128, using always powers of 2) and can be recursively split according to a quad-tree partitioning. The maximum size of a CTB and the maximum depth of the quad-tree partitioning are defined at the sequence level. When the splitting process is finalized, the leaf nodes of the CTB hierarchical tree become Prediction Units ( PU) and are used for motion-compensated predictive coding. Besides the CTBs and PUs, the HEVC standard also introduces the Transform Units (TU). These units are defined for transform and quantization purposes and can be as large as the size of the corresponding CTB leaf, i.e., the corresponding PU. The partitioning of TUs is also represented by quad-trees, with their maximum size and hierarchical depth being signaled in the bitstream. The transform block sizes are constrained to the maximum and minimum transform sizes, 4x4 and 64x64, respectively.

In addition to this novel partitioning scheme, the HEVC introduces new coding tools related to the intra-coding prediction supporting up to 33 spatial prediction directions for 8x8 to 64x64 blocks. With HEVC, it is also possible to use a 12-tap DCT-based interpolation filter to provide the same quarter-pixel accuracy interpolation already present in H.264/AVC. In this way, only one filtering procedure is needed, allowing a simplification of the implementation and a complexity reduction of the filtering process. In terms of the deblocking filter used in H.264/AVC, a symmetric Wiener filter has been added to allow a reduction of the quantization distortion in the reconstructed blocks and its adaptation is made to support the new larger block sizes. Regarding the entropy coding process, the HEVC standard offers two kinds of entropy coding methods: one for low-complexity, which uses 10 pre-determined VLC tables designed for different probability distributions; and one for high-complexity, using a variation of the CABAC solution defined in H.264/AVC. The bases of this codec are similar but the parallelization of the entropy encoding and decoding is introduced. Related to transform coding, the new partitioning scheme leads to new Integer DCT (IDCT) sizes, notably 16x16, 32x32 and 64x64, in addition to the usual H.264/AVC 4x4 and 8x8 block sizes. Besides the DCT, two types of directional transforms are adopted in the HEVC standard. These transforms are used when the DCT basis functions do not offer a good transform performance, e.g. uncorrelated signals or blocks with strong diagonal edges. The first directional transform is a Rotational Transform (ROT), which is applied as a second transform after the DCT for blocks of 16x16 and higher sizes. The basic principle behind this directional transform is the rotation of the transform basis coordination system, instead of the rotation of the input data. The second type of directional transform is the Mode Dependent Direction Transform (MDDT) which is used to encode 4x4 and 8x8 intra prediction residuals and is paired with the selected intra prediction mode. The 33 intra prediction modes for the 8x8 block size are grouped into nine separate directions; the MDDT is designed with nine separate basis functions, one for each direction. These basis functions are estimated from the statistics of the intra prediction residuals for each mode, using a separable transform based on the KLT, the Singular Value Decomposition (SVD). This transform is used to better exploit the spatial redundancy (versus the DCT) without excessively increasing the transform complexity (versus the KLT). In this way, the SVD is used first in the vertical and then in the horizontal directions.

4. ADOPTEO CODING SOLUTION

As already noted, the adopted video coding solution is based on the tool proposed, in 2010, by Biswas et al. [4]. Thus, it also uses a similar adaptive transform technique to code the prediction error associated to the inter-coded macroblocks. In this solution, the adaptive transform can switch between the standard H.264/AVC DCT and a modified KLT (very similar to the MKLT presented in Section 3.1) to obtain a better compression performance, depending on the particular details of the image area being coded. It was also referred above that this coding solution is based on the new HEVC codec as a replacement for the H.264/AVC codec used in [4]. However, it has to be noted that the proposed adaptive transform is not integrated in the codec’s reference software that is usually made available by the standardization groups, in this case the JCT-VC team. The full integration was not made because it would not only require detailed knowledge of the software structure and organization, which in this case would involve major extra time since this is a new software, still under development, but it would also require major software development and testing which is not the main objective of this work. As a reasonable compromise, HEVC encoded and decoded data is obtained/extracted (using the HEVC reference software) and used externally to simulate a large portion of the actual coding framework; for example, the HEVC entropy coding tool is not used. In this way, the developed coding solution is only applicable at the frame level, since the reference frames used for the inter-coded frames are always extracted from the HEVC codec and are not decoded from previous codings using the developed coding solution.

The general architecture of adopted coding solution designed and implemented in this work is presented in Figure 1. This solution is only used to code the prediction error block; thus, it uses only the inter-coded frames as input. Additionally, the bitstream generated by its encoding process and the reconstruction made by the decoding process only contain information about the prediction error.
Figure 1: Architecture of the adopted coding solution.

The architecture presented in Figure 1 includes three main processes which are described next:

- **HEVC framework** – This process is used to extract data from the HEVC to be used in the encoding and decoding processes of the adopted coding solution. In order to do this, the original frame is inter-coded with the HEVC codec.

- **AT encoder** – This process is used to encode each TU prediction error block using the proposed adaptive transform. Additionally, the coefficients generated by this transform are also quantized and entropy encoded. With the adaptive transform bitstream sent to the decoder side.

- **AT decoder** – This process is used to decode the adaptive transform bitstream sent by the decoder. To do this, each inter-coded TU bitstream is entropy decoded, inverse quantized and inverse transformed and the resulting reconstructed prediction error blocks are then rearranged to form the reconstructed prediction error frame.

The MKLT used in the developed coding solution is similar to the one proposed in [4], with the only difference being related to the used shift and rotation parameters. In this way, besides the already used shift and rotation parameters (referred in Section 3.1), ±0.75 and ±1.0 pixel shifts are also considered along with ±0.1° rotations.

### 5. PERFORMANCE EVALUATION

To evaluate the performance of the adopted coding solution, three CIF resolution video sequences were used: Container, Foreman and Mobile. The frame rate used for this type of sequences was 30 fps and all their 300 frames were coded. Additionally, a HD resolution video sequence was also coded, the Kimono sequence. The frame rate used to code this sequence was 25 fps and only 50 frames were coded, because of the large computation time required to code this type of resolutions.

To encode the selected test video sequences using the HEVC codec (TMuC software, version 0.9 [6]), the “Random access, high-efficiency setting” defined by the ICT-VC team has been used (described in Section 4.3 of [7]). This configuration was used since the objective of this work is to study the coding efficiency of the developed coding solution and not its complexity and this is the appropriate ICT-VC defined configuration for this purpose. The maximum and minimum CTB sizes are defined between 4×4 and 32×32, as well as the maximum and minimum TU sizes. The group-of-pictures structure uses an I-frame followed by P-frames with an intra-frame refresh period of 24. To compare the adopted coding solution performance with the usual DCT, three coding modes for the proposed adaptive transform have been defined with the following parameters:

- **Adaptive transform with half range shift and rotation parameters** – This mode of the adaptive transform uses a Half Range shift and rotation parameters Set (HRS) to compute the MKLT basis functions. This means that the maximum shift parameter is 0.5 pixels and the maximum rotation parameter is 0.5°; this AT mode is basically the same as used in [4].

- **Adaptive transform with full range shift and rotation parameters** – In this mode, the used MKLT basis functions are computed with a Full Range shift and rotation parameters Set...
(FRS). This means that the maximum shift parameter is 1.0 pixels and the maximum rotation parameter is 1.0°; this AT mode uses more shifts and rotations to estimate the prediction error than those used in [4].

- **Adaptive transform with HRS and FRS** – With this adaptive transform coding mode, the MKLT is basically divided into two MKLTs: one using the HRS mode and another using the FRS mode. In this way, at the decision module, the selection is made between 3 transforms: the DCT, the MKLT with HRS and the MKLT with FRS. The performance evaluation of the tested codecs is made by obtaining their Rate-Distortion (RD) curves. These curves are obtained by plotting the objective quality metric value for the reconstructed prediction error as a function of the amount of bits per second needed to code it. In the following, the adopted objective quality metric is the PSNR as it is commonly done in the literature, defined as follows:

\[
\text{PSNR} = 10 \log_{10} \left( \frac{\text{MAX}^2}{\text{MSE}} \right)
\]

where **MAX** is the maximum value of the input signal (255 for 8 bits samples) and **MSE** is the mean squared-error between the actual prediction error and the reconstructed prediction error. In Figure 2, Figure 3 and Figure 4 the RD curves for the CIF sequences are shown. The obtained results have shown that the proposed adaptive transform using a combination of the HRS and FRS modes (AT HFRS) can achieve a 0.61 dB gain of objective prediction error and 7% bitrate savings for the CIF sequences, always on average and over the DCT. For the other two adaptive transforms (AT HRS and AT FRS), the average results are very similar, with a prediction error PSNR improvement of 0.44 dB and bitrate savings of 5% over the DCT. This results show that the use of an additional FRS mode does not bring any compression improvements when used alone, but can bring approximately 0.2 dB of average PSNR improvement and 2% of average bitrate savings when used in combination with the HRS mode.

For the HD resolution video sequence (see Figure 5), the obtained results revealed reasonably higher coding gains that those obtained for the CIF sequences, although these gains are only verified for the low bitrate values. In this way, the adaptive transform using both the HRS and FRS modes (AT HFRS) was able to achieve 1.89 dB better prediction error objective quality and 16.0% bitrate saving in relation to the DCT, always on average. In this case, the adaptive transform only using the HRS mode (AT HRS) to compute the MKLT basis functions clearly outperformed the adaptive transform using the FRS mode (AT FRS). In this way, the adopted coding solution with an adaptive transform as proposed in [4] could achieve a prediction error PSNR improvement of 1.67 dB and bitrate savings of 14.7% over the DCT, always on average. Since the use of a FRS mode introduces a significant complexity increase in the video codec (as it uses 5.4 times more estimated prediction error blocks), the similarity of the results with only the HRS mode and with both the HRS and FRS modes indicate that the use of FRS may not be useful for HD resolution video sequences.

![Figure 2: Container sequence RD performance for the DCT, AT HRS, AT FRS and AT HFRS.](image-url)
Figure 3: *Foreman* sequence RD performance for the DCT, AT HRS, AT FRS and AT HFRS.

Figure 4: *Mobile* sequence RD performance for the DCT, AT HRS, AT FRS and AT HFRS.
An observation has to be made in relation to the achieved bitrate values. It is clear that these values are extremely higher than the ones achieved with the actual state-of-the-art video coding standard. However, these values are never used in their absolute form to take any type of conclusions about the adopted coding solution performance. Instead, these values are always analyzed in a relative way. The reason behind these high bitrate values is principally related to the entropy coder used in this solution. It is a very simple entropy coder which is not the object of study of this work. Additionally, it is also true that the HEVC codings are always performed using only P-frames and a single reference frame, which do not allow the exploitation of all the motion prediction tools.

6. CONCLUSIONS AND FUTURE WORK

A recent advance related to transform coding was studied, implemented and assessed in the context of the HEVC standard. Although the integration of the studied adaptive transform in the HEVC standard was not fully accomplished (for the reasons explained in Section 4), it was possible to extract the necessary data to simulate as much as possible a full integration scenario. With this, a performance evaluation of a video coding solution including the adopted adaptive transform was successfully made, showing positive results when compared to the currently used transform coding tools.

Clearly, the first improvement that can be made to the coding solution developed in this work is related to the full integration of the used adaptive transform in the HEVC codec. Future releases of new software versions of this codec tend to become more legible and organized from a programmer point of view. In this way, the proposed adaptive transform should be fully integrated in HEVC to allow a more complete and accurate evaluation of the performance changes introduced by the adaptive transform.

In the case of obtaining positive RD performance gains with the fully integrated coding solution proposed before, then the next step should target the study of the computational complexity associated to this solution, which was not considered in this work. This would be important to evaluate the trade-off between the additional complexity and the coding gains associated to the adaptive transform and to possibly develop new algorithms allowing a faster computation of the encoding and decoding process.

7. REFERENCES