Adaptive Transform Skipping for Improved Coding of Motion Compensated Residuals

Andrea Gabriellini, Matteo Naccari, Marta Mrak, David Flynn and Glenn Van Wallendael

BRITISH BROADCASTING CORPORATION
Adaptive Transform Skipping for Improved Coding of Motion Compensated Residuals

Andrea Gabriellini, Matteo Naccari, Marta Mrak, David Flynn and Glenn Van Wallendael

Abstract

The content of this White Paper has originally been published in Signal Processing: Image Communication journal.

New generations of video compression algorithms, such as those included in the under development High Efficiency Video Coding (HEVC) standard, provide substantially higher compression compared to their ancestors. The gain is achieved by improved prediction of pixels, both within a frame and between frames. Novel coding tools that contribute to the gain provide highly uncorrelated prediction residuals for which classical frequency decomposition methods, such as the discrete cosine transform, may not be able to supply a compact representation with few significant coefficients. To further increase the compression gains, this paper proposes transform skip modes which allow skipping one or both 1D constituent transforms (i.e. vertical and horizontal), which is more suitable for sparse residuals. The proposed transform skip mode is tested in the HEVC codec and is able to provide bitrate reductions of up to 10% at the same objective quality when compared with the application of 2D block transforms only. Moreover, the proposed transform skip mode outperforms the full transform skip currently investigated for possible adoption in the HEVC standard.

Additional Key Words: high efficiency video coding, discrete cosine transform, transform skip, coefficient scaling
ADAPTIVE TRANSFORM SKIPPING FOR IMPROVED CODING OF MOTION COMPENSATED RESIDUALS

Andrea Gabriellini¹, Matteo Naccari¹, Marta Mrak¹, David Flynn¹, and Glenn Van Wallendael²

¹British Broadcasting Corporation, Research and Development Department, London, United Kingdom
²Ghent University – IBBT ELIS Department - Multimedia Lab, Ledeberg-Ghent, Belgium

Abstract

New generations of video compression algorithms, such as those included in the under development High Efficiency Video Coding (HEVC) standard, provide substantially higher compression compared to their ancestors. The gain is achieved by improved prediction of pixels, both within a frame and between frames. Novel coding tools that contribute to the gain provide highly uncorrelated prediction residuals for which classical frequency decomposition methods, such as the discrete cosine transform, may not be able to supply a compact representation with few significant coefficients. To further increase the compression gains, this paper proposes transform skip modes which allow skipping one or both 1-D constituent transforms (i.e. vertical and horizontal), which is more suitable for sparse residuals. The proposed transform skip mode is tested in the HEVC codec and is able to provide bitrate reductions of up to 10% at the same objective quality when compared with the application of 2-D block transforms only. Moreover, the proposed transform skip mode outperforms the full transform skip currently investigated for possible adoption in the HEVC standard.

Keywords: high efficiency video coding, discrete cosine transform, transform skip, coefficient scaling.

1 Introduction

The recent advances in video capturing and display technologies will increase the presence of high and ultra-high definition video contents in mass market multimedia applications. This higher definition video content will require more storage space and transmission bandwidth. Therefore, in order not to exceed the physical limitations of storage and transmission devices, the compression efficiency for such content must be improved, notably beyond the efficiency provided by the state-of-the-art H.264/AVC video coding standard [1]. This ambition has gained further strength with the standardisation activities carried out by both ITU and ISO which joined efforts in the so-called Joint Collaborative Team on Video Coding (JCT-VC). The JCT-VC has the mandate to investigate and define the future High Efficiency Video Coding (HEVC) standard. HEVC follows a successful Call for Proposals (CfP) from January 2010 and targeting coding technology providing the same H.264/AVC objective quality at approximately half the bitrate [2]. JCT-VC has now achieved this goal by providing up to 51% bitrate reduction on average, compared to H.264/AVC, for certain classes of sequences and coding configuration, [3]. Gains are larger for sequences of higher resolution, which
is a desired outcome since such content is more demanding in terms of requirements for storage and transmission.

The HEVC codec still relies on the conventional block based hybrid motion compensated predictive architecture although several novel coding tools have been introduced at each coding stage. In particular, novel prediction directions are introduced for intra coding and block sizes larger than the hitherto common $16 \times 16$ are considered in intra prediction. Moreover, motion compensation is further improved with the adoption of longer interpolation filters and the lossy coding distortion is reduced by including two in-loop filters. These two new filters are the well known deblocking filter [4] extended towards the larger block sizes considered in HEVC and the Sample Adaptive Offset (SAO). Regarding the frequency transformation, a new integer variant of the Discrete Cosine Transform (DCT) is used for blocks of sizes $4 \times 4$ to $32 \times 32$, [5]. The DCT is widely used both in image and video coding given its good approximation of the optimal Karhunen-Loève Transform (KLT) in terms of energy compaction [6]. However, it is well known that for motion compensated predicted coded blocks the DCT may not provide a good energy compaction as the residuals are not as correlated as their image block counterparts [7], [8]. This phenomenon is particularly true for the new HEVC motion compensated predictions which lead to highly uncorrelated prediction residuals. Therefore, to efficiently encode the aforementioned signals, a more adaptive frequency decomposition strategy must be devised.

For coding of intra predicted residuals, studies related to HEVC already addressed a more adaptive approach that uses transforms dependent on the prediction mode. Basically, a choice of a horizontal and a vertical transform selected between the integer implementations of DCT and the Discrete Sine Transform (DST) can improve compression. Such transform selection improves representation of a given signal in the transform domain since the properties of the residual depend on the prediction direction [9], [10].

In that context, this paper addresses improved transformation of motion compensated residuals by introduction of a new Transform Skip Mode (TSM) which allows skipping one or both the 1-D constituent transforms (i.e. horizontal and vertical). The paper also addresses the adjustments required by TSM in order be aligned with the design of other coding tools, like scaling introduced in core transform design and quantisation. Moreover, a solution for interactions of TSM and perceptual quantisation matrices is proposed, for both square and non-square blocks which have been recently adopted in the HEVC draft standard. The increased flexibility granted by TSM improves exploitation of the statistics of the signal being transformed. Moreover, the proposed TSM does not require any additional transform basis and, given that some transformation operations are skipped, it can reduce the decoder complexity. Finally, it is demonstrated how TSM integrated in the HEVC reference software Model (HM) can improve performance compared to application of 2-D transforms only under common conditions.

The remainder of this paper is organised as follows. Section 2 provides a brief overview of alternative transforms and residual coding modes proposed in the literature. Section 3 proposes the TSM and its interaction with other coding tools, while Section 4 presents the experimental results obtained. Finally Section 5 concludes the paper and indicates future research directions.

2 Transforms in video coding

State-of-the-art video encoders, such as MPEG and H.26x standards, utilise four main processes to achieve a high level of visual data compression while achieving a desired level of video quality: prediction, transformation, quantisation and entropy coding. While in some cases unchanged source pixels are transformed, to greatly reduce the capacity required to represent the data the prediction process can be used to exploit the spatial and, in video, temporal redundancies. The prediction error, i.e. the residual, together with the additional prediction information, such as the prediction mode and motion vectors, is sent to the decoder to reconstruct the original signal. The transformation process aims to reduce the correlation present in the residual signals, and if the transform is a good match for a given signal, the energy of the signal will be concentrated into a few coefficients. Subsequent coding steps, quantisation and entropy coding, are then designed to compress such coefficients.
Residuals differ significantly due to their statistical properties resulting from the quality of the prediction and whether the prediction exploits spatial or temporal redundancy. It is well known that the DCT is suboptimal for some types of residuals [7]-[11]. For instance, a KLT approximation called Mode Dependent Directional Transform (MDDT) can be used for improved coding of intra prediction residuals [12]. In MDDT the transform basis functions are dependent on the intra prediction direction and have been derived by extensive training over video content spanning a wide range of spatial and temporal activities. The MDDT has been firstly proposed for the Key Technical Area (KTA) software platform used by the Video Coding Expert Group (VCEG) to research new coding tools improving the H.264/AVC compression efficiency. When integrated in the KTA software, the MDDT can achieve bitrate reductions of up to 9.7 % when compared to the H.264/AVC in the High profile [12].

Further work [9], [10] on MDDT concluded that, assuming a first autoregressive pixel correlation model, the basis function would be related to certain types of either DCT or DST. It has been shown that DST is a better choice than DCT for transformation of rows, when the directional prediction is from a direction that is closer to horizontal than vertical, and, similarly, is a better choice for transformation of columns when the directional prediction is closer to vertical. In the remaining directions (e.g. on rows, when DST is applied on columns), DCT is used. Given these characteristics, the authors in [13] demonstrated that a practical DCT/DST mode dependent transform enhances the performance of the HEVC codec [14], [15], saving up to 1.3 % bitrate compared to use of DCT only.

A further alternative to the MDDT and mode dependent DCT/DST is the Rotational Transform (ROT) proposed in [11] for intra predicted residuals. The rationale behind the ROT is that after the main transformation (e.g. DCT) the coefficients still present some directionality which can be reduced by applying some rotations. Such rotations arrange the coefficients along directions which are then more suitable for the following entropy encoding process. Only 4×4 and 8×8 ROT sizes are designed in [11] and applied over all square transform unit sizes specified by the HEVC standard, i.e. from 4×4 to 32×32. In particular, for 4×4 and 8×8 block sizes the 4×4 and 8×8 ROT are applied, respectively, while for higher block sizes, the 8×8 ROT is applied only to 8×8 top left subset of coefficients. When evaluated in HEVC, the ROT provides bitrate reductions of up to 3.5% for the same objective quality of the HEVC codec.

In addition to algorithms which are specific to intra predicted blocks, coding of residuals with strong directional components (both intra predicted or motion compensated) can benefit from application of other alternative transforms. The work in [16] proposes to apply the KLT over the motion compensated predicted residuals. Since the KLT basis functions are signal dependent [6], the usage of the KLT imposes the transmission of such functions to the decoder increasing the coding rate. In [16] this problem is addressed by estimating the basis functions at both encoder and decoder by applying a set of shifts and rotations to the prediction residuals. This approach has been tested in H.264/AVC with gains of up to 0.9 dB at the price of significant computational complexity increment at the decoder due to the basis function estimation. In [17], the use of 1-D directional transforms applied to prediction residuals is introduced. The intuition is that prediction residuals are often sparse signals with non-zero coefficients concentrated along objects boundaries and edges, forming mono-dimensional structures along them. This type of signal, presenting 1-D structures at various orientations, does not benefit from the common 2-D DCT; better energy compaction can be then achieved by selecting appropriate 1-D directional transforms. Since a set of transforms are now available for selection it becomes necessary to transmit the chosen transform to the decoder. With a specific set of 1-D directional block transforms for block sizes 4 × 4 and 8 × 8 in H.264/AVC, reported bitrate reduction is up to 12 % when compared with the standard codec. Another solution that applies directional transform is the Direction-Adaptive Residuals-Transform (DART) for intra and inter prediction residuals [18]. The DART consists in two steps: first 1-D DCT applied over 8×8 residuals blocks and along pre-defined directions and second 1-D DCT applied only to the DC coefficients generated in the first step. The proposed DART has been integrated in the KTA software and tested over a wide range of spatial and temporal resolution videos. Compared to the KTA software, the DART provides average bitrate reductions of up to 6.7% for the same objective...
quality. Moreover, when compared with the KTA software equipped with the MDDT, the DART provides up to 6.56% bitrate reduction more than the KTA with MDDT as proposed in [12].

An additional increase of the transform efficiency can be achieved by spatial partitioning of coding units. The Residual Quad Tree (RQT) [19] approach used in HEVC adaptively selects the size of a transform block by recursively partitioning a $B \times B$ block of residuals in a quad tree fashion. At the encoder, the splitting configuration for each coded block is decided upon a Lagrangian Rate Distortion (RD) cost function minimisation computed over each Transform Unit (TU). Further extensions of RQT include Non-Square Quad Tree (NSQT) [20], which enable non-square TU leading to the need for non-square transforms. This approach supports asymmetric motion compensated predictor partitioning, which improves compression of motion compensated residuals.

Finally, an alternative to the transformation process may consist in the residuals quantisation directly in the spatial domain without applying any transform. The work in [8] proposes a spatial domain quantisation to be used in any block based motion compensated predictive video coding architecture. The residuals are quantised with a uniform dead-zone quantiser, scanned and then entropy coded. Differently from transformed domain residuals, in spatial domain the scanning is in the direction of decreasing absolute gradient of the residuals values. When applied in the KTA, such scheme provides up to 8% bitrate reduction for the same objective quality. The idea of residual spatial domain quantisation has been also exploited in the context of the HEVC for intra coded blocks. The Transform Skip (TS) coding tool [21] is a spatial domain residuals quantisation applied only to 4×4 intra coded blocks. When integrated in the HM codec and tested for intra-only coding, it provides gains of up to 7.8% BD-rate for screen content (i.e. content that consists of both computer generated/synthesised and camera captured elements).

Video coding schemes that are not block based also benefit from adaptive transforms. For example, in wavelet video coding the directionlets [22] are capable of characterising oriented patterns in images or motion compensated frames by adaptation of the transform directions to the locally dominant direction across the spatial domain. Adaptive lifting-based one dimensional wavelet transforms have also been proven to deliver gains in compression of residuals [23]. While these examples illustrate similar requirements for transform adaptation where frame-level transforms are used, the reminder of this paper is focused on block-based codecs utilising DCT and DST transforms.

From a practical implementation point of view, new transforms require additional matrices or additional implementations of fast algorithms as well as additional memory and logic at both encoder and decoder sides. If a matrix multiplication realisation of a transform is considered and even if the memory to store transform matrices is small, matrix multiplication realisation design may be too complex for software implementations. On the other hand, for hardware implementation, different hardware logic would be required for each transform if fast algorithms are designed for each core base, making such implementations impractical. For example, if a different butterfly or other fast algorithm is considered for additional transforms, it would require totally different functions to be optimised for additional transforms, making it impractical for hardware because of numerous different transform modules.

Therefore, in order to make transforms more useful in today’s video codec, in the next section an alternative way of adapting the transform is proposed which does not suffer any of the aforementioned limitations.

# 3 The proposed transform skip mode

It is possible, and indeed it is also common under some circumstances, for the two-dimensional signal in input to the transform to display different statistics along the two vertical and horizontal axes. In this case it would be desirable to choose the best performing transform on each axis. On the other hand, the introduction of a new transform basis is usually impractical for real-world implementations. Moreover, instead of applying a transform along a given axis, a better choice may be to skip it. This section presents the transform skip mode proposed in this paper for inter coded
A. Transform Skip Mode

The transform skip mode proposed in this paper defines a skip in one or both dimensions on which the DCT is computed (i.e., horizontal and vertical). Figure 1 depicts the four modes in which the proposed TSM can operate. The arrows (blue and green) in the figure refer to the direction where the transform is applied on a block. Figure 1(a) refers to the usual DCT in which neither the horizontal nor the vertical direction is skipped. Conversely, Figure 1(b)-(c) depict the skip in the vertical and the horizontal dimensions, respectively. It should be noted that these skip modes allow the transform to decorrelate those residuals which present a correlation only on either the horizontal or the vertical dimension, respectively. Finally Figure 1(d) depicts the skip in both dimensions which corresponds to not applying any transformation, thus quantising the residuals directly in the pixel domain. This latter skip mode is selected for those residuals that do not present any correlation in either dimension. A schematic representation of the whole TSM process is also depicted in Figure 2. It should be noted that the skip mode in Figure 1(d) is conceptually the same as the one proposed in [8]. Therefore, the spatial domain quantisation proposed in [8] can be seen as a special case of the TSM proposed in this paper. The four possible modes in which TSM can operate can be represented by a single word, denoted with $TSM_{C}$, as shown Table 1.

![Figure 1](image1.png)

**Figure 1:** Allowed transform skip modes for a coded block: (a) 2D transform, (b) horizontal transform (vertical skip), (c) vertical transform (horizontal skip) and (d) no transform (full skip).

![Figure 2](image2.png)

**Figure 2:** Schematic representation of the implementation of the transform skip modes.

<table>
<thead>
<tr>
<th>$TSM_{C}$ value</th>
<th>Horizontal Direction</th>
<th>Vertical direction</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Transformed</td>
<td>Transformed</td>
<td>2-D Transform</td>
</tr>
<tr>
<td>1</td>
<td>Transformed</td>
<td>Skipped</td>
<td>1-D Transform</td>
</tr>
<tr>
<td>2</td>
<td>Skipped</td>
<td>Transformed</td>
<td>1-D Transform</td>
</tr>
<tr>
<td>3</td>
<td>Skipped</td>
<td>Skipped</td>
<td>No transform</td>
</tr>
</tbody>
</table>

**Table 1:** $TSM_{C}$ values and associated skipped transform.
HEVC standard, the HM codec will be considered as case study. When TSM is integrated in the HM codec, some implementation issues should be addressed. More precisely, TSM interacts with coefficient scaling, entropy coding of $TSM_C$ value and coefficients, the usage of quantisation matrices and finally the Rate Distortion Optimisation (RDO). The following sub-sections will provide a description of how TSM interacts with these coding stages and the steps taken to integrate it into HM.

### B. Coefficients scaling

The challenge associated with efficient application of TSM is related to ensuring that coefficients are at the same level irrespectively of TSM mode. With signal levels of each TSM mode carefully adjusted, coefficient quantisation and RD optimisations can follow well-established 2D transform routines. The problem of different signal levels for each TSM mode is related to the properties of the underlying codec. Focusing on HEVC, during transforms the levels of the signal are changed according to:

- The norms of the transforms, and
- Scaling factors to preserve 16-bit precision after each transform stage.

Therefore these two parameters have to be considered for coefficient level scaling when transforms are skipped.

In HEVC the one-dimensional DCT is defined for signals of 4, 8, 16 and 32 samples. For each of these transforms, the norms of its basis vectors are sufficiently close to each other to allow the use of the same scaling values during quantisation or dequantisation of transform coefficients. The values of actual norms are very close to approximated values shown in Table 2.

<table>
<thead>
<tr>
<th>Transform size</th>
<th>scale</th>
<th>$Iscale$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Integer approximation of the transform norm</td>
<td>$\log_2$ approximation of the transform norm</td>
</tr>
<tr>
<td>4</td>
<td>128</td>
<td>7</td>
</tr>
<tr>
<td>8</td>
<td>181</td>
<td>7.5</td>
</tr>
<tr>
<td>16</td>
<td>256</td>
<td>8</td>
</tr>
<tr>
<td>32</td>
<td>362</td>
<td>8.5</td>
</tr>
</tbody>
</table>

For implementation purposes, in video coding it is common to limit the precision of transform coefficients after each transform stage (horizontal or vertical). The core transforms in HEVC use 16–bit intermediate data representation and after each transformation stage the coefficients values are limited to 16 bits via scaling operations (binary shifts).

A straightforward approach for adjusting levels of blocks with any TSM mode is to apply scaling directly on the coefficients values. Each coefficient from a block on which transforms are skipped or applied, can be scaled (multiplied) by a factor $W(TSM_C, N, M) / 2^{\delta(TSM_C)}$, where $W$ and $\delta$ depend on TSM choice and block height ($M$) and width ($N$), since the scaling depends on the skipped transform size. Variable $W$ reflects the norms of skipped transforms and is defined as:

$$W(TSM_C, N, M) = \begin{cases} 
1 & TSM_C = 0 \\
\text{scale}(N) & TSM_C = 1 \\
\text{scale}(M) & TSM_C = 2 \\
\text{scale}(N) \cdot \text{scale}(M) & TSM_C = 3 
\end{cases}$$

where $\text{scale}(\cdot)$ is dependent on the transforms used in the targeted codec (Table 2). Finally, the
variable $\delta$ depends on support for precision limiting due to intermediate data range restrictions and on the internal bit depth increment:

$$\delta(TSM_c) = \begin{cases} 
0 & TSM_c = 0 \\
sc & TSM_c = 1 \\
sr & TSM_c = 2 \\
sc + sr & TSM_c = 3 
\end{cases}$$

(2)

where and $s_C$ and $s_R$ are shifts that are necessary for preservation of the intermediate data range, for the transform on columns and rows, respectively. For example, in this scenario HEVC decoder uses $s_C = 7$ and $s_R = 12 - \Delta$, where $\Delta$ denotes the internal bit depth increment, relative to the common processing bit depth (8 bits).

In practical implementation the aim is to reduce the number of operations required to achieve the desired performance. In that context, another example of how the level of the signal can be adjusted when a transform is skipped is presented in the following. In order to avoid the previously introduced weighting of each coefficient, the required level adjustment can be achieved in an alternative way, depending on the TSM choice:

- $TSM_c = 0$ (2-D transform): Level adjustment is not required.
- $TSM_c = 1$ and $TSM_c = 2$ (Transform performed in one direction only): Part of required scaling can be performed using bit-shifting. That part is related to the given $\delta$ and the integer part of $lscale(N)$ (see Table 2). Depending on skipped transform size, adjustment by an additional fractional component (0.5) of $lscale(N)$ may be required. This final adjustment can in HEVC be realised by adding an offset of 3 to the quantisation, which correspond to scaling by a factor of $\sqrt{2}$.
- $TSM_c = 3$ (Full skip): Each coefficient level is adjusted by shifting by $s_C + s_R$ bits, as in (2).

In this way the scaling for 1-D transform skip can be realised without additional burden on the underlying codec, while full skip requires only cheap bit-shifting operations.

**C. Entropy coding of TSM values and coefficients**

The TSM modes, represented by $TSM_c$ values, are entropy encoded using Context Adaptive Binary Arithmetic Coding (CABAC). The $TSM_c$ value for each block is encoded before the coefficients for the luminance block. Then, the coefficients are entropy encoded in the order defined by the scanning patterns. While diagonal and zig-zag scans are usually the best choices for the scanning of 2-D transformed coefficients, when the transform is skipped in one direction, the coefficients are clustered differently from the 2-D DCT. This is demonstrated in the example of Figure 3 where the TSM code is 1 and a more suitable scanning, vertical scanning, is suggested. Therefore, in this paper a TSM dependent scanning mode is defined and the selected scanning patterns are listed in Table 3. Note that introduction of new scans for different block sizes and different modes may increase entropy coding complexity (the number of context and logic needed) so it may be impractical for a codec implementation. Table 3 also summarises the choice of scanning modes which does not introduce an additional burden to the design of the HEVC entropy coding module. Finally, if no coefficients are encoded for the luminance component, then no $TSM_c$ information is sent and the value is assumed to be $TSM_0$, or standard 2-D DCT.
D. Use of quantisation matrices when transform is skipped

HEVC has adopted a flat quantiser, i.e., the same quantisation step is used on all transformed coefficients. However, it is well known that the Human Visual System (HVS) shows a different sensitivity to the noise introduced at different spatial frequencies [25], [26]. When considering the properties of the HVS, a different quantization step size may be applied to each coefficient. These different quantization steps are in this case defined by quantization matrices. The literature is rich in proposals that performed psycho-visual experiments to measure the HVS distortion sensitivity to different spatial frequency (see [25], [27], [28] and references therein). From the collected experimental data, some models for the HVS distortion sensitivity have been proposed [27], [28], [29]. The quantisation matrices used in the HM codec have been designed using the findings reported in [30]. This work proposes a nonlinear relation between the spatial radial frequency $f_r$ and the HVS sensitivity $H(f_r)$. The HVS sensitivity threshold corresponds to the maximum amount of distortion that can be added to a given spatial frequency value without being noticeable to human observers. The value for $H(f_r)$ is given by [30] as:

$$H(f_r) = a \cdot (b + c \cdot f_r) \cdot e^{-c(f_r)^d}, \quad (3)$$

where the parameters $a$, $b$, $c$, and $d$ are equal to 2.2, 0.192, 0.114 and 1.1, respectively. The spatial radial frequency $f_r$ is related to the horizontal and vertical spatial frequencies as follows:

\[ f_r = f_h + f_v, \]

Table 3: TSM dependent scanning patterns for 4 × 4 and 8 × 8 blocks.

<table>
<thead>
<tr>
<th>TSMc value</th>
<th>Scanning pattern</th>
<th>HEVC-specific patterns</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>Diagonal</td>
<td>Default(^1)</td>
</tr>
<tr>
<td>1</td>
<td>Vertical</td>
<td>Vertical for 4 × 4 and 8 × 8 blocks(^2)</td>
</tr>
<tr>
<td>2</td>
<td>Horizontal</td>
<td>Horizontal for 4 × 4 and 8 × 8 blocks(^2)</td>
</tr>
<tr>
<td>3</td>
<td>Diagonal</td>
<td>Default(^1)</td>
</tr>
</tbody>
</table>

\(^1\) Refers to the HEVC default scan for 2-D transform [15].
\(^2\) For other block sizes the default scan is used.
\[ f_r = \sqrt{f_h(i)^2 + f_v(j)^2}. \]  

Each spatial frequency \( f_h(i), f_v(j) \) is related to the corresponding DCT (DST) bands \((i, j)\) as:

\[ f_h(i) = \frac{i}{dp \times 2M} \quad f_v(j) = \frac{j}{dp \times 2N}, \]  

where \( M \) and \( N \) denote the horizontal and vertical block sizes, respectively. In earlier video coding standards blocks are squares \((M = N)\), while the model in (3) also supports rectangular blocks (as required for non-square blocks in HEVC). The quantity \( dp \) denotes the display dot pitch, which corresponds to the distance between two dots (i.e., two pixels of the same colour) in the display screen. The dot pitch is usually measured in millimetres and according to [31] it has been set to 0.25 mm. Finally, the values for the Quantisation Matrix (QM) for the DCT band \((i, j)\), \( QM(i, j) \) is given by:

\[ QM(i, j) = \left\lfloor \frac{QP}{H(f_r)} + 0.5 \right\rfloor, \]  

where \( QP \) denotes the average quantisation parameter, which is set to 12 for HEVC. When the transform is skipped in one direction (horizontal or vertical), the use of the quantisation matrices should be carefully analysed, namely in the following aspects:

- Whether the values for the current quantisation matrices still apply when a transform is skipped.
- How the 2-D quantisation matrices can be reused for the 1-D transform case, in order to limit the overall number of quantisation matrices used. In fact, the HEVC standard allows video content producers to use different quantisation matrices specifically designed for a given video content. This flexibility requires the matrix values to be transmitted to the decoder, thus increasing the coded rate.

To answer the first point, it should be noted that when a transform is performed in only one direction (and skipped in the other) it is equivalent to having either \( f_h(i) \) or \( f_v(j) \) equal to zero. In this case, \( QM \) is computed as in (6) with either \( f_h(i) \) or \( f_v(j) \) equal to zero, depending on which dimension is skipped. More precisely, by looking at (5), if the transform is skipped in the vertical direction, the radial spatial frequency varies only along the horizontal dimension and therefore is dependent only on the \( j \) variable. Therefore, the quantisation matrix when the vertical transform is skipped \( QM(0, j) \) for \( j = 0, \ldots, N - 1 \) The same analysis holds also for the quantisation matrix when the transform is skipped along the horizontal direction. For that case the quantisation matrix is \( QM(i, 0) \) for \( i = 0, \ldots, M - 1 \). Finally, when both the transforms are skipped (i.e., no transform) the quantisation takes place in the pixel domain and therefore it is not desirable to use a frequency weighted quantisation matrix. From the analysis given for the first point, the second point can now be addressed as well. In particular, for an \( M \times N \) block of residuals, it is proposed to derive the TSM quantisation matrix \( (QM_{TSM}) \) as follows:

\[ QM_{TSM}(i, j) = \begin{cases} 
QM(i, j) & \text{if } TSM_c = 0 \\
QM(0, j) & \text{if } TSM_c = 1 \\
QM(i, 0) & \text{if } TSM_c = 2 \\
1 & \text{if } TSM_c = 3 
\end{cases}, \]  

with \( i = 0, \ldots, M - 1 \) and \( j = 0, \ldots, N - 1 \). A graphical representation of the \( QM_{TSM} \) selection is also demonstrated in Figure 4 assuming that the transform is skipped in the vertical direction over a 4×4 block of residuals.
E. TSM integration in the rate distortion optimisation loop

For each block the choice of TSM mode is decided by minimising a Lagrangian RD cost function. More precisely, in HEVC, during the Rate Distortion Optimisation (RDO) loop performed over a given RQT, the four transform skip modes are tested and the one leading to the best RD cost is selected. Since one TSM value is coded and transmitted per transform unit, all image components, luminance and two chrominances, are processed with the same transform skip mode. Finally, the HEVC encoder performs Rate Distortion Optimised Quantisation (RDOQ, [24]) to improve the compression efficiency. The RDOQ is not applied for TSMc = 3 (full skip) because this skip mode retains the block coefficients in their spatial (pixel) positions, while RDOQ in HM assumes frequency domain ordering of coefficients.

4 Experimental results and discussion

This section presents the experiments performed to assess the RD performance of the proposed TSM. First the coding conditions, test material, benchmarks and performance indicators are introduced, and then the results are presented and discussed.

A. Coding conditions and test material

The JCT-VC test video sequences have been used, including the screen content (Class F) videos [32]. Four configurations have been evaluated in the tests, Random Access – Main (RA-Main), Random Access – High Efficiency 10 (RA-HE10), Low Delay – Main (LD-Main), and Low Delay – High Efficiency 10 (LD-HE10) as specified in [32]. In addition to the basic tools used in the Main setting, HE10 includes adaptive loop filter, chroma from luma intra prediction mode, non-
square transforms, 10-bit internal processing bit depth and full transform skip for 4 × 4 intra coded blocks. Finally, four quantisation parameters (QP) were used, 22, 27, 32 and 37 for intra coded frames while the QPs for the inter ones are automatically derived by the encoder according to the frame position in the Group Of Picture (GOP) and the GOP length [32].

JCT-VC test material is grouped into 6 classes. First 5 classes (A to E) contain camera captured content. Class A (2560 × 1600, 30 and 60 fps) contains 4 sequences that represent material of the highest resolution, thus this content is tested only in random access setting. Classes B (5 sequences, 1920 × 1080, 24 to 60 fps), C (4 sequences, 832 × 480, 30 to 60 fps) and D (4 sequences, 416 × 240, 50 and 60 fps) are common to both random access and low-delay configurations. Class E is video conferencing material, therefore considered for testing of low-delay configurations only. It contains 3 sequences (1280 × 720, 60 fps). Sequences in Class F are screen content sequences, i.e. they contain both camera captured and graphical elements (captured desktops including text editing, pictures, computer game, camera captured content with graphical overlays). Resolution of 4 sequences in Class F ranges from 832 × 480 to 1280 × 720, with frame rates of 20 to 50 fps. All sequences are in 4:2:0 chrominance sampling format.

B. Benchmarks and performance indicators

To fully assess the rate distortion performance of the HM codec equipped with proposed transform skip mode (hereafter denoted as HM-TSM), two benchmarks are considered for comparison. The first benchmark is represented by the HM codec (version 7.0) which uses only 2D transform; this codec will be hereafter denoted as HM-7.0. Conversely, the second benchmark is the HM-codec which uses either the 2D transform or the fully skip mode for inter coded blocks. This codec will be hereafter denoted as HM Transform Skip HM-TS and corresponds to the inter Transform Skip (TS) as proposed in [33], [34] and currently under investigation in the JCT-VC. It should be noted that the HM-TS codec can be seen as the integration of the method in [8] in HM. The main differences with respect to the method proposed in [8] consist in the block size restricted to 4 × 4 and in the scanning of the coefficients which is not performed according to the gradient computed over the residuals.

As performance indicator, the Bjøntegaard Difference (BD, [35]), with respect to the HM-7.0 codec is used assuming the Peak-Signal-to-Signal-Ratio (PSNR) as distortion measure. More precisely, for each video content the BD consists in an integral difference between the RD curve of the HM-7.0 and the RD curve of the codec used for comparison (i.e. either HM-TSM or HM-TS). The BD may be computed to quantify the bitrate reduction for the same objective quality (commonly known as BD-rate) or to quantify the PSNR difference for the same bitrate (commonly known as BD-PSNR). In this paper the BD-rate are used and negative numbers indicate bitrate savings with respect to the HM-7.0 codec.

C1 Experimental results

A summary of BD-rates of the proposed TSM versus HM-7.0 codec are listed in Table 4. In this table some test points (RA Class E and LD Class A) are not measured according to the JCT-VC common test conditions [32]. As may be noted, TSM provides bitrate reductions for the luminance component for almost all test points and the reduction is up to 5.3 % (Class F, LD-Main). The highest BD-rate reductions are obtained for the Class F sequences. These sequences contain a high quantity of text details and sharp edges which lead to prediction residuals correlated only along the horizontal or vertical or neither dimension. For these residual statistics TSM performs better than applying a 2-D transform. Moreover, Table 5 reports the BD-rates for the HM-TS codec, as may be noted the HM-TS codec provides lower BD-rate values than the proposed HM-TSM. This result suggests that limiting the block size to 4 × 4 penalises the overall codec performance. Most importantly, for camera captured content (Classes A to E) using only full skip is not as beneficial as the proposed method that can skip transform only in one direction. The reason for this performance difference lies in the nature of motion compensated residuals that often have strong horizontal and vertical patterns.
Table 4: Average BD-rates percentages for the HM-TSM codec.

<table>
<thead>
<tr>
<th>Class</th>
<th>RA-Main</th>
<th>RA-HE10</th>
<th>LD-Main</th>
<th>LD-HE10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>-0.1</td>
<td>0.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>-0.5</td>
<td>-0.2</td>
<td>-0.8</td>
<td>-0.2</td>
</tr>
<tr>
<td>C</td>
<td>-0.6</td>
<td>-0.4</td>
<td>-1.1</td>
<td>-0.6</td>
</tr>
<tr>
<td>D</td>
<td>-0.8</td>
<td>-0.7</td>
<td>-1.6</td>
<td>-1.2</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td></td>
<td>-0.9</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>-2.4</td>
<td>-1.9</td>
<td>-5.3</td>
<td>-3.5</td>
</tr>
<tr>
<td>Average</td>
<td>-0.88</td>
<td>-0.62</td>
<td>-1.94</td>
<td>-1.10</td>
</tr>
</tbody>
</table>

Table 5: Average BD-rates percentages for the HM-TS codec.

<table>
<thead>
<tr>
<th>Class</th>
<th>RA-Main</th>
<th>RA-HE10</th>
<th>LD-Main</th>
<th>LD-HE10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.0</td>
<td>0.1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>C</td>
<td>-0.2</td>
<td>-0.1</td>
<td>-0.2</td>
<td>-0.3</td>
</tr>
<tr>
<td>D</td>
<td>-0.3</td>
<td>-0.3</td>
<td>-0.5</td>
<td>-0.5</td>
</tr>
<tr>
<td>E</td>
<td>-</td>
<td></td>
<td>0.0</td>
<td>0.0</td>
</tr>
<tr>
<td>F</td>
<td>-2.2</td>
<td>-1.8</td>
<td>-4.3</td>
<td>-3.1</td>
</tr>
<tr>
<td>Average</td>
<td>-0.54</td>
<td>-0.42</td>
<td>-1.00</td>
<td>-0.78</td>
</tr>
</tbody>
</table>

Figure 5 provides further insight into the results obtained with TSM. In this figure, for four sequences, namely “Basketball Drive” (Class B), “BQ Square” (Class D), “China Speed” (Class F) and “Slide Editing” (Class F). In these figures, the bitrate savings provided by TSM are computed for all tested configurations and QPs. More precisely, the saving in bitrate is computed at the same PSNR level provided by the HM-7.0 codec for each tested bitrate. It can be observed that, for some test points the savings are about 8-10% (Slide Editing, QP = 27, LD-Main and China Speed, QP = 27, LD-Main, respectively). Overall, the gain is more significant for low-delay applications because of the nature of that configuration. More precisely, low-delay by definition has only one intra coded frame in each tested sequence and therefore the ratio of motion compensated blocks, on which TSM is applied, is higher.
Figure 5: Bitrate savings introduced by TSM in bit-rate points of the HM-7.0 anchors.

To demonstrate the PSNR ranges obtained in this test, Figure 6 shows some examples of rate-distortion curves. The curves offer a comparison between the HM-7.0 and the HM-TSM codecs. Once again it is possible to note the sizeable coding efficiency improvement yielded by TSM for selected material. For other test points used in the summary presented in Table 4 PSNR values are also in the range 30 to 45 dB, i.e. the test covers typical compression ranges.
Figure 6: Examples of RD curves for some tested contents.

Figure 7 provides an example of the TSM mode selection for a frame from the Slide Editing sequence belonging to Class F. More precisely, the 253rd frame is represented in Figure 7(a) whilst Figure 7(b) shows the same frame coded with LD-Main and QP = 22. In Figure 7(b), black lines denote the TU partitioning, red blocks denote TUs where the transform is skipped vertically ($TSM_c = 1$), yellow blocks denote TUs where the transform is skipped horizontally ($TSM_c = 2$), green blocks denote TUs where the transform is fully skipped and finally blocks without a shade denote TUs where the HEVC 2-D transform is applied. As may be noted the Slide Editing sequence contains several sharp details relative to the text and the icon buttons. The main motion component in this sequence is related to the slide scrolling and text editing. The residuals for the areas involved by this kind of motion present strong correlation either along the horizontal or vertical direction. In these areas it is therefore preferable to skip the transform in the vertical or horizontal direction (red and yellow blocks respectively). In areas interested by text with different characters and thus several orientations, the transform is fully skipped (green blocks).
Finally, it is worthwhile to analyse the complexity of the proposed TSM. At the encoder side the complexity is increased since during the RDO the transform skip modes are checked in addition to the usual 2-D transform. In common test conditions, where the computational complexity is quantified in terms of the encoder run time, a 30% increment in the encoder run time has been reported. It should be noted that this complexity can be reduced using fast algorithms which detect which transform will be used on the basis of some heuristics. At the decoder the computational complexity can be also reduced given that some transform operations are skipped. As an example, Table 6 reports the percentage of usage of the TSM for all block sizes and all the JCT-VC contents.
coded in the Low Delay configuration (both Main and HE10). Assuming that each 2-D transform counts as one operation, the proposed TSM allows to reduce by 22% the number of full 2-D transform operations required.

Table 6: Percentage of usage of different TSM modes for all contents in the Low Delay configurations.

<table>
<thead>
<tr>
<th>TSMc value</th>
<th>Percentage of selection over all transform block sizes</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>63</td>
</tr>
<tr>
<td>1</td>
<td>16</td>
</tr>
<tr>
<td>2</td>
<td>14</td>
</tr>
<tr>
<td>3</td>
<td>7</td>
</tr>
</tbody>
</table>

5 Conclusions and future work

This paper has proposed the transform skip mode for inter coded blocks. TSM allows a transform to be skipped in one or both dimensions to better match the statistics of each block of prediction residuals, providing significant bitrate reductions. To harmonise simple transform skip with other components of a video codec, new methods for coefficient scaling have been devised assuming the codec currently considered in the HEVC standardisation project as a case study. The experimental results presented show that TSM improves RD performance for a wide range of inter coding configurations and sequences, with the largest gains for video content with sharp details, such as text and computer generated graphics. Such content is typical in today’s television broadcast programmes (e.g. news, sport events, etc.). Further work should address evaluation of TSM for different types of transforms and content, such as 3D content and scalability.

References


[34] A. Gabriellini, M. Mrak, D. Flynn and M. Naccari, “Transform skipping for inter predicted coding units”, JCTVC-J0077, 10th meeting, Stockholm, SE, July 2012.