Trapezoidal Block Split Using Orthogonal C2 Transforms for HEVC Video Coding

Itsik Dvir, Amiram Allouche, David Drezner,
Ady Ecker, Dror Irony, Natan Peterfreund,
Haitao Yang, Zhou Jiantong
Huawei Technologies Co., Ltd.

***Abstract*—We present an extension for HEVC intra-frame coding with trapezoidal splits and orthogonal transforms. A block can be split into two 180-degrees rotationally-symmetric (C2) trapezoidal parts, each coded separately using standard DCT implementation. We also introduce part-to-part prediction from a diagonal edge. The optimal trapezoidal split of a quad tree block is selected in a rate-distortion sense. We achieved 0.8% reduction in BD-rate over HEVC in standard test conditions for intra coding.**

I. Introduction

State-of-the-art video compression standards, such as HEVC [1], use square or rectangular blocks as coding tree units. This design allows for hierarchical quadtree partitioning structure that can adapt to the content of the image, and it combines well with 2D separable transforms such as 2D-DCT [2]. However, axis-aligned splits may be suboptimal for representing all natural images. For example, object occlusion may create image blocks where a diagonal edge separates two regions with very different content. In such cases, it could be more efficient to represent the two parts with two sets of transform coefficients, and it could be more efficient to perform intra-prediction along the diagonal edge.

Diagonal splits create trapezoid-like shapes for which the 2D-DCT cannot be applied. In [3], a transform in the spirit of Shape Adaptive DCT (SA-DCT [4]) was developed to represent trapezoidal blocks. The idea is to perform two passes of 1D-DCT, along diagonals and rows. However, the basis vectors in this transform, as in SA-DCT, are not orthogonal in 2D, hence it is less efficient compared to orthogonal transforms such as the 2D-DCT.

A new orthogonal transform for trapezoidal shapes was developed in [5] where the authors suggested to use a segmentation algorithm to decompose images to trapezoidal, rectangular and triangular blocks, according to the shapes of objects. That approach achieved better compression than JPEG.

In this paper, following the idea in [5], we suggest an alternative option to encode a rectangular block [6] [7] [8]. We select the best split, among a predefined set of splits, to encode a given rectangular block by two 180-degree rotationally-symmetric parts. Since this transform is based on rotational symmetry of order 2, from here on we will abbreviate this transform as ”C2”.

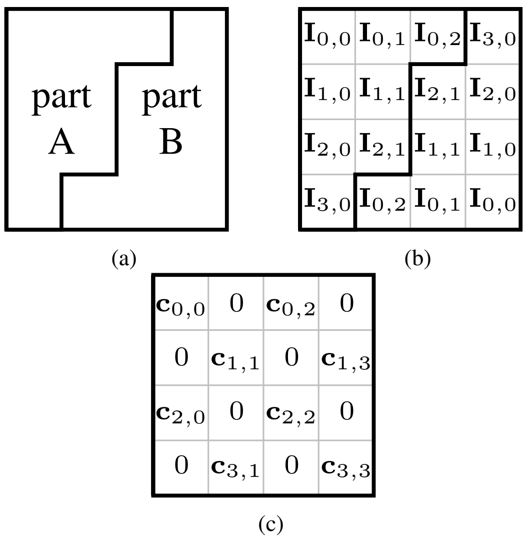
part A
part B
$\mathbf{I}_{0,0}$ $\mathbf{I}_{0,1}$ $\mathbf{I}_{0,2}$ $\mathbf{I}_{3,0}$
$\mathbf{I}_{1,0}$ $\mathbf{I}_{1,1}$ $\mathbf{I}_{2,1}$ $\mathbf{I}_{2,0}$
$\mathbf{I}_{2,0}$ $\mathbf{I}_{2,1}$ $\mathbf{I}_{1,1}$ $\mathbf{I}_{1,0}$
$\mathbf{I}_{3,0}$ $\mathbf{I}_{0,2}$ $\mathbf{I}_{0,1}$ $\mathbf{I}_{0,0}$
$\mathbf{c}_{0,0}$ 0 $\mathbf{c}_{0,2}$ 0
0 $\mathbf{c}_{1,1}$ 0 $\mathbf{c}_{1,3}$
$\mathbf{c}_{2,0}$ 0 $\mathbf{c}_{2,2}$ 0
0 $\mathbf{c}_{3,1}$ 0 $\mathbf{c}_{3,3}$
(a)
(b)
(c)

Fig. 1: The C2 transform. (a) A square is split to two rotationally-symmetric trapezoidal parts. (b) To perform the C2 transform of part A, 2D mirroring is preformed by rotating part A and overlaying it on part B. (c) The C2 transform is the DCT transform of the entire square (b). Note that half of the coefficients are always 0. The C2 transform of part B is computed similarly.

We present a complete integration of the C2 transform into the workflow of the HEVC standard. Our focus is on compressing intra-frames or still images. We integrated the C2 transfrom into the systematic search phase of HEVC. In the modified search, a block of a quad tree is tested also for an alternative coding as two C2 parts.

We also adapted the intra-prediction to allow part-to-part prediction from diagonal edges as opposed to the horizontal and vertical edges in standrad HEVC. RDOQ [9] and coefficients scanning mechanisms were also adapted to the C2 transform.

Our work is along a recent active research line on multiple transforms [10]. In contrast to previous work, our transform uses the standard DCT and can be implemented efficiently with existing DCT algorithms.

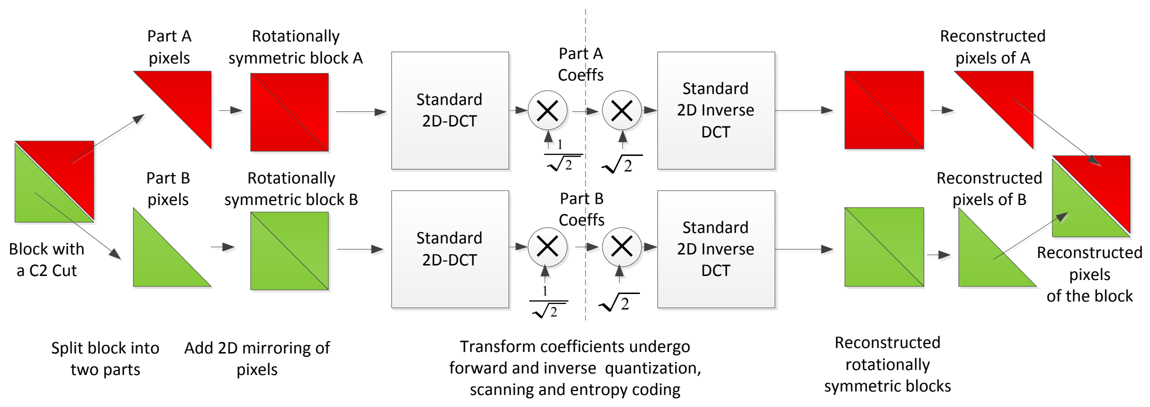
Part A pixels
Rotationally symmetric block A
Standard 2D-DCT
Part A Coeffs
Standard 2D Inverse DCT
Reconstructed pixels of A
Block with a C2 Cut
Part B pixels
Rotationally symmetric block B
Standard 2D-DCT
Part B Coeffs
Standard 2D Inverse DCT
Reconstructed pixels of B
Reconstructed pixels of the block
$\frac{1}{\sqrt{2}}$
$\sqrt{2}$
Split block into two parts
Add 2D mirroring of pixels
Transform coefficients undergo forward and inverse quantization, scanning and entropy coding
Reconstructed rotationally symmetric blocks

Fig. 2: A system to encode/decode rectangular blocks using C2 transform.

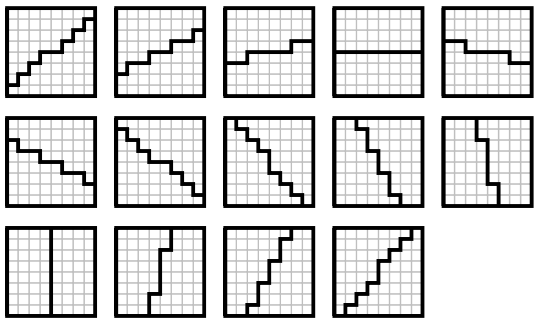

Fig. 3: The predefined subset of C2 cuts for 8×8 blocks. Each cut is obtained by shifting its endpoints clock-wise one pixel. For an $N{\times}N$ block, there are $2N{-}2$ such cuts.

II. The C2 Transform

The C2 transform is an orthogonal DCT expansion of trapezoidal blocks, as explained in Fig. 1. The transform is applied to trapezoidal shapes whose 180° rotation about the center point of the block equals their complement shape in the block (square or rectangle). A full encoding/decoding scheme for rectangular blocks using C2 transform is shown in Fig. 2.

In the rest of the paper, we will refer to the rotationally-symmetric staircase line between the parts as a "cut". Note that the number of such possible cuts can be quite large. Therefore, in this work, for an $N{\times}N$ block we consider only a subset of $2N{-}2$ cuts, as shown in Fig. 3 for an 8×8 block.

The C2 transform is obtained by rotating 180° each part about the center of the block and performing the 2D-DCT. It was proven in [5] that the DCT of a rotationally-symmetric image has zeros in positions (i, j) where $i + j$ is odd. Furthermore, the restriction of the DCT basis functions with even $i{+}j$ to the domain of part A forms an orthogonal basis for that part [5]. In order to get an orthonormal basis we multiply the basis functions by $\sqrt{2}$.

Since the C2 transform is derived from 2D-DCT, it has several appealing properties. First, it is an orthogonal transform. Second, it admits a computationally fast transform with existing algorithms. Third, it is a separable transform, a useful property to represent axis-aligned structures that are common in images.

In the appendix we derive some mathematical properties of the C2 transform that were not explained in [5].

III. Implementation

In this section we describe in more detail our integration of the C2 transform into the HM-13 test model [11]. We applied the C2 transform only to luma blocks. $N{\times}N$ leaf blocks in the quadtree structure, where $N{=}4, 8, 16$, may be coded either by the standard DCT or by the C2 transform. The choice is signaled in the bitstream with a new *C2Flag*. For 4×4 blocks we examine only 2 possible cuts, parallel and orthogonal to the intra-prediction direction, since for 4×4 blocks the overhead in signaling the cut index is significant. For 8×8 and 16×16 blocks, the search for C2 cuts is over a fixed set of 14 and 30 $(2N{-}2)$ possible cuts of the block, respectively. The cut index is coded with a single bit to signal whether the cut is closer to the intra-prediction direction or the orthogonal direction, and the difference from that reference direction is coded by truncated unary and exp-Golomb [12]. Our coding implementation uses additional contexts. For example, there are contexts for the coded block flags (*CBF*) for parts A and B. The contexts of the *C2Flag* and *CBF* are neighborhood-dependent. Different sets of contexts are used depending on the existence of C2 transforms in neighboring prediction units (PU). The best encoding in rate-distortion sense is selected.

In addition to the transform, we made several adaptations to other parts of the codec.

A. *Intra-Prediction Directions Selection*

In the standard flow of HEVC, a subset of k directions out of 35, where k depends on the block size, is chosen for evaluation in each Prediction Unit (PU). For each prediction direction out of the 35, the residual is formed by subtracting the prediction in that direction from the image. The cost of the block in HEVC is estimated as:

$$cost_{HEVC}(prediction\ direction) = \\ \lambda \cdot cost(prediction\ direction, MPM) + SATD\ , \tag{1}$$

where the first term is an estimation of coding cost of the prediction mode with respect to the Most Probable Mode (*MPM*), and *SATD* is the sum of absolute difference values of the Hadamard transform, used to estimate the cost of the DCT. The lowest k values are maintained for further examination.

Since the HEVC candidate prediction directions selected for DCT are sub-optimal for the C2 transform, we create another list, with k prediction directions, and merge it with the list of k directions selected by the HEVC standard. The idea here is to pick prediction directions s.t. at least one of the parts (A or B) of the residual is as smooth as possible. We measure smoothness simply by checking the maximal derivative of the residual in absolute value over all possible cuts.

$$cost_{C2}(prediction\ direction) = \\ \min_{\text{all C2 cuts}} \Bigg(\lambda \cdot cost(prediction\ direction, MPM) + \\ \min\Big(\max_{\text{part A}} \big(|dx|, |dy|\big), \max_{\text{part B}} \big(|dx|, |dy|\big) \Big) \Bigg). \tag{2}$$

The procedure above is applied unless the maximal absolute derivative in the residual block is smaller than some threshold. In such case the entire residual block is smooth, and no C2 cuts are examined to save computations. Our experience with this procedure is that it is more effective than simply enlarging the candidates list with the HEVC's procedure.

We turned off the `HHI_RQT_INTRA_SPEEDUP` flag in our experiments both in the reference and C2 test code. This turns off an optional speed optimization that does not pick the best prediction directions for C2, so we can demonstrate the potential of C2 without optimization shortcuts.

B. Intra-Prediction from Part to Part

Angular prediction [13] is a key contributor to the success of HEVC. The goal of angular prediction is to carry information from blocks already coded to the next block to be coded. Unlike [3], our implementation performs prediction between parts with a diagonal edge. After the first part is reconstructed, we back-project the pixels along the cut onto the block boundary, and re-use the standard angular prediction for the second part, as illustrated in Fig. 4. Both parts (A and B) share a single prediction direction. The prediction direction determines which part (A or B) is predicted first, and whether the projection is from the top or left boundary.

C. Coefficients Packing and Scanning

In HEVC the transform coefficients are scanned into a vector. In C2 transform, only coefficients $\mathbf{c}_{i,j}$ with even $i+j$ are nonzero. To avoid transmitting zeros, we pack the coefficients by moving $\mathbf{c}_{i,j}$ to position $\mathbf{c}_{i,\lfloor j/2 \rfloor}$. The coefficients scanning orders follow the standard, except that we skip the fixed zero coefficients (in $4{\times}4$ blocks) and fixed zero coding-groups (in $8{\times}8$ and $16{\times}16$ blocks).

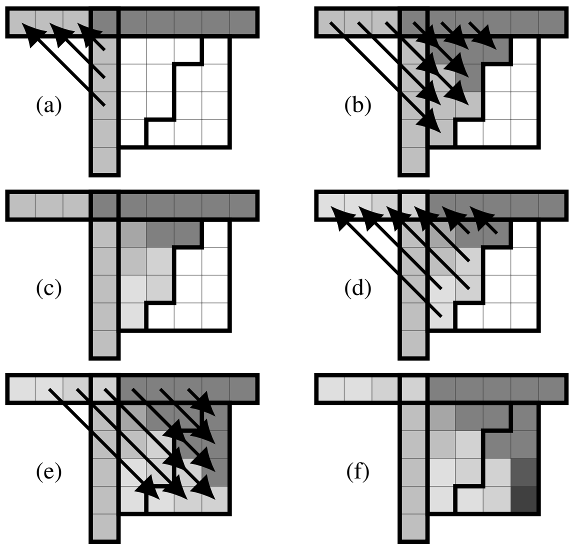

Fig. 4: Encoding of a C2 block with prediction from part to part. (a) The left boundary is projected onto the top boundary. (b) Angular prediction to part A. (c) Part A after C2 transform, quantization, inverse quantization and inverse C2 transform. (d) Cut boundary pixels are back-projected onto the top boundary. (e) Angular projection to part B. (f) Part B after C2 transform, quantization, inverse quantization and inverse C2 transform.

IV. Results

We evaluated our system on the entire standard Common Test Conditions (CTC) benchmark [14]. The results are shown in Table I. The average BD-rate [15] reduction over HEVC is 0.8%. Note especially the reduction of 4.3% in Class *F* that contains screen content images with sharp edges.

Fig. 5 shows a section of the Transform Unit (TU) tree for the first frame of the CTC sequence "BasketballDrill", produced by our system. The magenta lines show TUs with C2 transform and their C2 cuts. In this frame 12.7% of the pixels were coded by the C2 transform.

We did not optimize the running time beyond skipping the C2 transform for very smooth blocks. The running time can be reduced in several ways. For example, C2 transforms of consecutive cuts can be computed by updating, since the difference in the C2 transform of the next cut depends only on about N pixels that enter or leave part A. Also, the running time can be greatly reduced by not evaluating all C2 cuts, especially for the larger block sizes.

V. Conclusion

In this paper we investigated the integration of the C2 transform into HEVC. We presented several contributions. First, the method can encode a quad tree block as two 180-degree rotationally-symmetric (C2) trapezoids. Second,

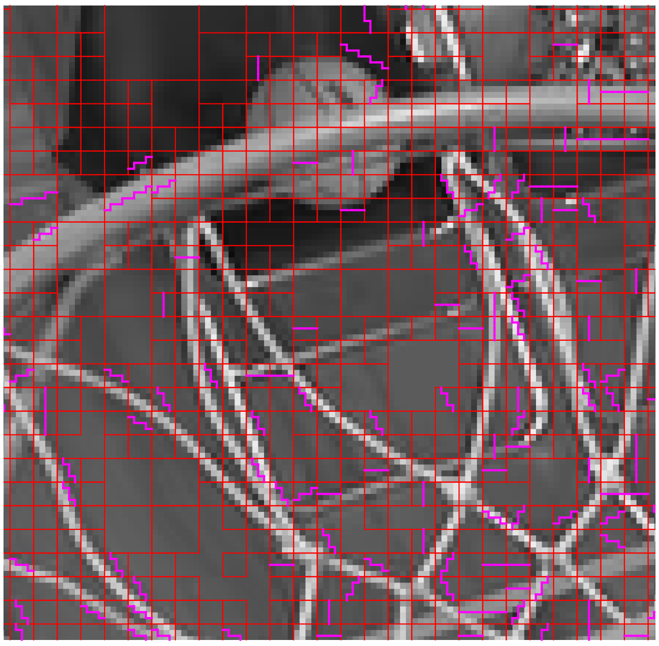

Fig. 5: A TU quadtree. In red: the coded quadtree structure. In magenta: C2 cuts for coded C2 blocks. Often C2 splits blocks into smooth and detailed parts.

All Intra Main	**Y**	**U**	**V**
Class *A*	-0.3%	0.1%	-0.1%
Class *B*	-0.5%	0.5%	0.4%
Class *C*	-1.3%	0.4%	0.4%
Class *D*	-1.4%	0.2%	0.2%
Class *E*	-0.5%	0.3%	0.3%
Overall	**-0.8%**	**0.3%**	**0.3%**
Class *F*	-4.3%	-2.0%	-2.0%
Enc Time[%]	**530%**		
Dec Time[%]	**97%**		

TABLE I: HEVC vs. HEVC+C2 on CTC benchmark.

it utilizes existing optimized DCT implementations (no need for additional transform hardware). Third, a block is encoded more efficiently when the rate-distortion optimization function selects the best C2 cut among a predefined set of block partitions. Fourth, we introduced part-to-part intra prediction from a diagonal edge (C2 cut) in addition to the intra prediction of HEVC from the vertical and horizontal edges. Fifth, in the appendix we presented a mathematical analysis describing the connections between the C2 coefficients of trapezoidal blocks and the DCT coefficients of their bounding square block. We also presented several implementation adaptations and performed the HEVC CTC evaluation.

While enlarging the set of the C2 cuts contribute to the coding gain, we found that the overhead of signaling the cut index can be significant. We address this issue by using neighborhood-adaptive contexts and restricting the number of cuts for $4{\times}4$ blocks. Overall we obtained 0.8% average reduction in BD-rate over standard HEVC.

VI. Appendix: The Connection Between DCT and C2 Coefficients

In this appendix we derive mathematical properties of the C2 transform that explain how C2 coefficients relate to DCT coefficients. We use the following notation: Let $\mathbf{P}$ be some $N{\times}N$ patch, $\mathbf{P}_A$ and $\mathbf{P}_B$ are the C2 parts of $\mathbf{P}$ defined by some cut. $\mathbf{p}, \mathbf{p}_A, \mathbf{p}_B$ are vectorizations of the pixels of $\mathbf{P}, \mathbf{P}_A, \mathbf{P}_B$ by a column-wise scanning order, $\mathbf{v}_{even}$, $\mathbf{v}_{odd}$ are $N{\times}N/2$ coefficient vectors of the DCT transform of $\mathbf{P}$ that correspond to DCT bases functions $\mathbf{b}_{i,j}$ with $i+j$ even or odd, where $i,j = 0,\dots,N{-}1$. $\mathbf{c}_A$, $\mathbf{c}_B$ are $N{\times}N/2$ vectors of the C2 coefficients of $\mathbf{P}_A$ and $\mathbf{P}_B$.

Observation 1.

$$\mathbf{v}_{even} = (\mathbf{c}_A + \mathbf{c}_B)/\sqrt{2} \tag{3}$$

Proof. Consider any single DCT basis vector $\mathbf{b}_{ij}$, where $i+j$ is even. Denote by $\mathbf{b}_{ijA}$ and $\mathbf{b}_{ijB}$ the components of $\mathbf{b}_{ij}$ on the domain of the pixels of $\mathbf{P}_A$ and $\mathbf{P}_B$. The corresponding DCT coefficient is $\mathbf{b}_{ij}^t \cdot \mathbf{p} = (\mathbf{b}_{ijA}^t \cdot \mathbf{p}_A + \mathbf{b}_{ijB}^t \cdot \mathbf{p}_B)/\sqrt{2}$. The $\sqrt{2}$ factor arises because the $\mathbf{b}_{ijA}^t$ and $\mathbf{b}_{ijB}^t$ needs to be scaled compared to $\mathbf{b}_{ij}^t$ to become unit norm basis functions for $\mathbf{P}_A$ and $\mathbf{P}_B$ [5]. However, $\mathbf{b}_{ijA}^t \cdot \mathbf{p}_A$ and $\mathbf{b}_{ijB}^t \cdot \mathbf{p}_B$ are exactly the C2 coefficient for $\mathbf{P}_A$ and $\mathbf{P}_B$. $\square$

Observation 1 relates the even DCT coefficients of an $N{\times}N$ patch to the C2 coefficients of parts A and B in case there is no prediction between these parts. In that case, it is possible to speed up the computation of all C2 cuts, by computing $\mathbf{v}_{even}$ once, $\mathbf{c}_A$ for all other cuts, and deriving $\mathbf{c}_B$ without a transform. However, in our implementation, we used prediction from part A to part B. Since this prediction is a non-linear operation (uses reconstructed samples of part A) we cannot use this observation to speed up performance.

Observation 2. *For any C2 cut s, there exists an orthogonal matrix $\mathbf{M}_s$ s.t.*

$$\begin{aligned} \mathbf{c}_A &= (\mathbf{v}_{even} + \mathbf{M}_s \cdot \mathbf{v}_{odd})/\sqrt{2} \\ \mathbf{c}_B &= (\mathbf{v}_{even} - \mathbf{M}_s \cdot \mathbf{v}_{odd})/\sqrt{2} \end{aligned} \tag{4}$$

Proof. Let $\mathbf{B}_{even}$ be a matrix whose columns are the DCT basis vectors $\mathbf{b}_{ij}$ with even $i+j$, and $\mathbf{B}_{odd}$ a similar matrix with odd $i+j$. A vectorized patch $\mathbf{p}$ can be represented as $\mathbf{p} = \mathbf{B}_{even} \cdot \mathbf{v}_{even} + \mathbf{B}_{odd} \cdot \mathbf{v}_{odd}$, where $\mathbf{v}_{even}$, $\mathbf{v}_{odd}$ are the corresponding DCT coefficients. A restriction to the domain of the pixels of $\mathbf{P}_A$ gives $\mathbf{p}_A = (\mathbf{B}_{evenA} \cdot \mathbf{v}_{even} + \mathbf{B}_{oddA} \cdot \mathbf{v}_{odd})/\sqrt{2}$, where $\mathbf{B}_{evenA}$ and $\mathbf{B}_{oddA}$ are obtained from $\mathbf{B}_{even}$ and $\mathbf{B}_{odd}$ by removing the rows that correspond to pixels of $\mathbf{P}_B$, and multiplying by $\sqrt{2}$ to get unit norm basis vectors. The C2 transform of $\mathbf{P}_A$ is obtained by applying the matrix $\mathbf{B}_{evenA}^t$

$$\begin{aligned} \mathbf{c}_A &= (\mathbf{B}_{evenA}^t \mathbf{B}_{evenA} \mathbf{v}_{even} + \mathbf{B}_{evenA}^t \mathbf{B}_{oddA} \mathbf{v}_{odd})/\sqrt{2} \\ &= (\mathbf{v}_{even} + (\mathbf{B}_{evenA}^t \mathbf{B}_{oddA}) \mathbf{v}_{odd})/\sqrt{2} \\ &= (\mathbf{v}_{even} + \mathbf{M}_s \cdot \mathbf{v}_{odd})/\sqrt{2}\ , \end{aligned}$$

$$\mathbf{M}_s = \mathbf{B}_{evenA}^t \mathbf{B}_{oddA}\ . \tag{5}$$

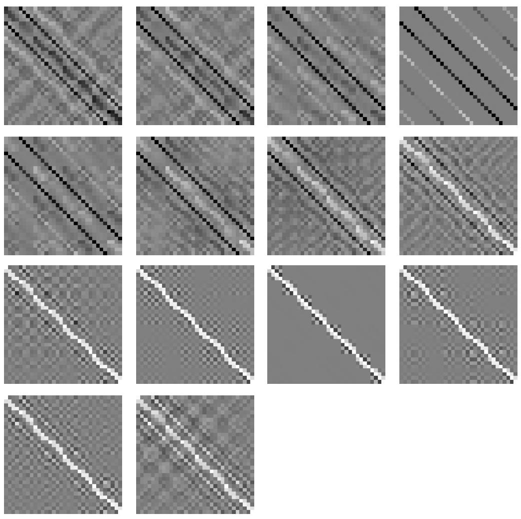
Fig. 6: The matrices $\mathbf{M}_s$ for the 14 possible cuts of $8{\times}8$ patches with same cuts order of Fig. 3. Bright pixels are strong positive values while dark pixels are strong negative values. In the first 7 cuts, the energy of an odd coefficient is distributed mainly into its upper an lower neighbors. In the last 7 cuts, the energy is distributed mainly to its left and right neighbors.

The columns of $\mathbf{M}_s$ in (5) are nothing but the C2 transform of the odd-index DCT basis functions. $\mathbf{M}_s$ is orthogonal since

$$\mathbf{M}_s^t\mathbf{M}_s{=}\mathbf{B}_{oddA}^t\mathbf{B}_{evenA}\mathbf{B}_{evenA}^t\mathbf{B}_{oddA}{=}\mathbf{B}_{oddA}^t\mathbf{B}_{oddA}=\mathbf{I}\ .$$

From observation 1 we get

$$\begin{aligned}\mathbf{c}_{_B}&{=}\sqrt{2}\mathbf{v}_{even}-\mathbf{c}_{_A}{=}\sqrt{2}\mathbf{v}_{even}-(\mathbf{v}_{even}+\mathbf{M}_s\cdot\mathbf{v}_{odd})/\sqrt{2}\\&{=}(\mathbf{v}_{even}-\mathbf{M}_s\cdot\mathbf{v}_{odd})/\sqrt{2}\ .\end{aligned}$$

□

Observation 2 implies that the C2 coefficients are generated by linearly transforming the odd-index DCT coefficients of the $N{\times}N$ patch, and adding to the even-index coefficients. The C2 transform essentially redistributes the energy of the odd coefficients over the even coefficients. The matrices $\mathbf{M}_s$ govern the way by which this redistribution occurs. Fig. 6 displays the matrices $\mathbf{M}_s$ for $N{=}8$. It can be observed that significant parts of the energy are spread along the main diagonal and two principle diagonals. These diagonals correspond to spreading the energy of an odd coefficient to nearby even coefficients on the same row or the same column of the two-dimensional ordering of the DCT bases.

With observations 1 and 2 at hand, we can now analyze an ideal C2 case. Assume a block as shown in Fig. 7 where after DC prediction the residual of part A is zero and of part B is arbitrary (not constant). The $N{\times}N$ DCT may have N^2 coefficients, whereas the C2 may have $N{\times}N/2$ coefficients. However, by observation 1, $\mathbf{c}_{_A}=0$, and hence $\mathbf{c}_{_B}=\sqrt{2}\mathbf{v}_{even}$. Therefore, each C2 coefficient in $\mathbf{c}_{_B}$ is $\sqrt{2}$ larger than the $\mathbf{v}_{even}$ coefficients. When this overhead is much

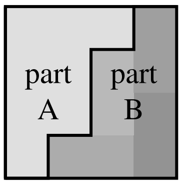

Fig. 7: An ideal case for C2. The C2 cut separates this block into a uniform part (A) and a non-uniform part (B).

smaller than the $N{\times}N/2$ coefficients of $\mathbf{v}_{odd}$, the C2 encoding is more efficient than DCT encoding of the $N{\times}N$ block.

References

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] G. Strang, "The discrete cosine transform," *SIAM review*, vol. 41, no. 1, pp. 135–147, 1999.

[3] S. Hu, R. A. Cohen, A. Vetro, and C.-C. J. Kuo, "Screen content coding for HEVC using edge modes," in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, May 2013, pp. 1714–1718.

[4] T. Sikora and B. Makai, "Shape-adaptive DCT for generic coding of video," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 5, no. 1, pp. 59–62, 1995.

[5] J.-J. Ding, Y.-W. Huang, P.-Y. Lin, S.-C. Pei, H.-H. Chen, and Y.-H. Wang, "Two-dimensional orthogonal DCT expansion in trapezoid and triangular blocks and modified JPEG image compression," *IEEE transactions on image processing*, vol. 22, no. 9, pp. 3664–3675, 2013.

[6] I. Dvir, N. Peterfreund, D. Irony, and D. Derezner, "Systems and methods for processing a digital image," International Patent Application WO2 016 074 744, Nov. 14, 2014.

[7] ——, "Systems and methods for processing a block of a digital image," International Patent Application WO2 016 074 745, Nov. 14, 2014.

[8] ——, "Systems and methods for mask based processing of a block of a digital image," International Patent Application WO2 016 074 746, Nov. 14, 2014.

[9] J. Stankowski, C. Korzeniewski, M. Domański, and T. Grajek, "Rate-distortion optimized quantization in HEVC: Performance limitations," in *Picture Coding Symposium (PCS), 2015*. IEEE, 2015, pp. 85–89.

[10] A. Arrufat Batalla, "Multiple transforms for video coding," Theses, INSA de Rennes, Dec. 2015. [Online]. Available: https://tel.archives-ouvertes.fr/tel-01303759

[11] I.-K. Kim, K. McCann, K. Sugimoto, B. Bross, W.-J. Han, and G. Sullivan, "JCTVC-O1002 High Efficiency Video Coding (HEVC) test model 13 (HM 13) encoder description," *JCT-VC, Jan*, 2014.

[12] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard," *IEEE Transactions on circuits and systems for video technology*, vol. 13, no. 7, pp. 620–636, 2003.

[13] J. Lainema, F. Bossen, W.-J. Han, J. Min, and K. Ugur, "Intra coding of the HEVC standard," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1792–1801, 2012.

[14] F. Bossen, "JCTVC-L1100 Common Test Conditions and softwere reference configurations," *JCT-VC, Jan*, 2013.

[15] G. Bjontegaard, "Calcuation of average PSNR differences between RD-curves," *Doc. VCEG-M33 ITU-T Q6/16, Austin, TX, USA, 2-4 April*, 2001.