A NOVEL FRAMEWORK FOR DESIGNING DIRECTIONAL LINEAR TRANSFORMS WITH APPLICATION TO VIDEO COMPRESSION

Maurizio Masera, Giulia Fracastoro, Maurizio Martina, Enrico Magli

Department of Electronics and Telecommunications
Politecnico di Torino - Italy

ABSTRACT

Transforms incorporating directional information are appealing in a wide range of applications. In this paper, we introduce a new framework that allows to define a directional transform starting from any two-dimensional separable transform. The proposed method is highly general and it can be of interest in many areas of signal processing. We also show an example of possible application. We define a directional integer DCT and DST and we show their application in video compression by integrating them in the HEVC video coding standard.

***Index Terms*—** Directional transform, video coding

1. INTRODUCTION

Directional transforms are of great interest in signal processing thanks to their numerous applications in signal compression and analysis. The adaptation of geometric parameters can optimally match the transform to the signal of interest, resulting in a more precise signal representation that can be convenient in many application fields.

In the past, many attempts have been made to incorporate directional information into the transform operator. One of the most famous directional transforms is the directional discrete cosine transform [1], which consists in a separable transform in which the first 1D-DCT may follow a direction other than the vertical or horizontal one. Another method to introduce directionality in the DCT has been presented in [2], where directional primary operations have been introduced for the lifting-based DCT. Moreover, [3] presents another directional transform called rotational transform, which consists in a secondary transform applied after the DCT that induces small modifications of the DCT coefficients.

Other directional transforms employ more sophisticated nonseparable geometries. Curvelets [4] are one of such examples, which provide an efficient representation of discontinuities along smooth curves. Another example is presented in [5], where the authors introduce a new class of bases called bandlets, which are adapted to the local directions in which the image has regular variations. Moreover, in [6] the discrete contourlet transform is presented, where the authors apply a directional image expansion using contour segments.

Recently, the theory of graph signal processing [7] has been exploited in order to design a steerable discrete cosine transform [8, 9, 10] and a steerable discrete Fourier transform [11]. In the first case, starting from the graph transform of a grid graph, a new directional transform can be designed by rotating the 2D-DCT basis in any chosen direction. Instead, in the second case this concept is extended and a new generalization of the DFT is presented. The proposed steerable DFT can be defined in one or two dimensions. In the first case, it can be interpreted as a rotation on the complex plane, instead in the second case it represents a rotation in the 2D Euclidean space.

All these directional methods, however, are designed for a specific transform. Instead, in this paper we propose a new technique for designing directional transforms that can be applied to a wide range of transforms. In particular, we extend the framework presented in [9, 11] in order to define a new method that allows us to define a directional transform for *any* two-dimensional separable transform. This new directional transform can be computed in an efficient way and it can be oriented in any chosen direction. We also show an example of possible applications of the proposed technique. We define a directional integer DCT and DST and we present their application to video compression by implementing them in the HEVC video coding standard. We show that, using transform basis with different orientations, we can obtain a sparser representation, and therefore an improved coding efficiency.

2. PROPOSED METHOD

Given a two-dimensional signal represented by a matrix $X \in \mathbb{C}^{n\times n}$, an arbitrary two-dimensional invertible transform may be represented as $\hat{\mathbf{x}} = T\mathbf{x}$, where $\mathbf{x} \in \mathbb{C}^{n^2}$ is the vectorization of X obtained by stacking the columns of X and $T \in \mathbb{C}^{n^2\times n^2}$ is the transform matrix. A separable two-dimensional transform can be defined as $Y = AXB^H$, where $A \in \mathbb{C}^{n\times n}$ and $B \in \mathbb{C}^{n\times n}$ are one-dimensional transforms that operate separately on the signal columns and rows, respectively. The equivalent two-dimensional transform matrix is defined as $T = A \otimes B$, where $\otimes$ is the Kronecker product. If we consider the case when $A = B$, we have that

$$T = A \otimes A. \tag{1}$$

Then, let $t^{(l,k)} \in \mathbb{C}^{n^2}$ be the vector corresponding to the i-th row of T, where $i = l \cdot n + k$ and $0 \leq l, k \leq n-1$, we can define the matrix $T^{(l,k)} \in \mathbb{C}^{n \times n}$ in the following way

$$T^{(l,k)} = \begin{bmatrix} t_1^{(l,k)} & t_{n+1}^{(l,k)} & \cdots & t_{(n-1)n+1}^{(l,k)} \\ \vdots & \vdots & & \vdots \\ t_n^{(l,k)} & t_{2n}^{(l,k)} & \cdots & t_{n^2}^{(l,k)} \end{bmatrix}.$$

From (1), we obtain that $T^{(l,k)} = T^{(k,l)^T}$. This shows that, if $l \neq k$, $T^{(l,k)}$ and $T^{(k,l)}$ represent the same frequency component, but one in the horizontal direction and the other one in the vertical direction. Therefore, by properly rotating these vectors we can define a general method for designing, starting from T, a new directional transform. The proposed technique represents a generalization of the method presented in [9, 11] to any separable transform defined as in (1).

Given a pair of vectors $t^{(l,k)} \in \mathbb{C}^{n^2}$ and $t^{(k,l)} \in \mathbb{C}^{n^2}$ corresponding respectively to the i-th and j-th rows of T, where $i = l \cdot n + k$ and $j = k \cdot n + l$ with $0 \leq l, k \leq n-1$ and $l \neq k$, we can rotate them in the following way

$$\begin{bmatrix} t^{(l,k)'} \\ t^{(k,l)'} \end{bmatrix} = \begin{bmatrix} \cos(\theta_{k,l}) & \sin(\theta_{k,l}) \\ -\sin(\theta_{k,l}) & \cos(\theta_{k,l}) \end{bmatrix} \begin{bmatrix} t^{(l,k)} \\ t^{(k,l)} \end{bmatrix}, \qquad (2)$$

where $\theta_{k,l}$ is an angle in $[0, 2\pi]$. In order to define the new directional transform $T(\boldsymbol{\theta}) \in \mathbb{C}^{n^2 \times n^2}$, all the pairs $t^{(l,k)}$ and $t^{(k,l)}$, with $0 \leq l, k \leq n-1$ and $l \neq k$, are replaced by the rotated ones $t^{(l,k)'}$ and $t^{(k,l)'}$. In this way, the i-th row of $T(\boldsymbol{\theta})$, with $i = l \cdot n + k$, corresponds to the vector $t^{(l,k)'}$ if $l \neq k$, instead if $l = k$ the i-th row of $T(\boldsymbol{\theta})$ is equal to the i-th row of T. It is interesting to point out that we can use more than one rotation angle per block. The number of rotated vector pairs is $p = \frac{n(n-1)}{2}$ and each vector pair can be rotated by a differend angle. The vector $\boldsymbol{\theta} \in \mathbb{R}^p$ contains all the angles used.

The new directional transform $T(\boldsymbol{\theta}) \in \mathbb{C}^{n^2 \times n^2}$ can be defined as

$$T(\boldsymbol{\theta}) = R(\boldsymbol{\theta})T, \qquad (3)$$

where $R(\boldsymbol{\theta}) \in \mathbb{R}^{n^2 \times n^2}$ is the rotation matrix. The structure of $R(\boldsymbol{\theta})$ is defined so that, for each pair of vectors $t^{(l,k)}$ and $t^{(k,l)}$ where $k \neq l$, it performs the rotation described in (2). It is easy to see that $R(\boldsymbol{\theta})$ can be decomposed in two matrices as $R(\boldsymbol{\theta}) = \Delta + \tilde{R}(\boldsymbol{\theta})$, where $\Delta \in \mathbb{R}^{n^2 \times n^2}$ is a constant matrix representing the vectors that do not rotate (i.e. $t^{(l,l)}$), and $\tilde{R}(\boldsymbol{\theta}) \in \mathbb{R}^{n^2 \times n^2}$ represents the rotating vectors. Δ is a diagonal matrix, with $\Delta_{ii} = 1$ for any $i = k \cdot n + k$, otherwise $\Delta_{ii} = 0$. Given $0 \leq k, l \leq n-1$ and $k \neq l$, if $i = k \cdot n + l$ and $j = l \cdot n + k$, then $\tilde{R}(\boldsymbol{\theta})_{ii} = R(\boldsymbol{\theta})_{jj} = \cos(\theta_{k,l})$, $\tilde{R}(\boldsymbol{\theta})_{ij} = \sin(\theta_{k,l})$ and $\tilde{R}(\boldsymbol{\theta})_{ji} = -\sin(\theta_{k,l})$.

We highlight that the rotation described in (2) preserves the total energy of the coefficients

$$|\tilde{\mathbf{x}}_{l,k}|^2 + |\tilde{\mathbf{x}}_{k,l}|^2 = |\hat{\mathbf{x}}_{l,k}|^2 + |\hat{\mathbf{x}}_{k,l}|^2, \qquad (4)$$

where $\tilde{\mathbf{x}}_{l,k}$ and $\tilde{\mathbf{x}}_{k,l}$ are the transform coefficients of $T(\boldsymbol{\theta})$ corresponding respectively to the transform vectors $t^{(l,k)'}$ and $t^{(k,l)'}$ and $\hat{\mathbf{x}}_{l,k}$ and $\hat{\mathbf{x}}_{k,l}$ are the transform coefficients of T corresponding respectively to the transform vectors $t^{(l,k)}$ and $t^{(k,l)}$.

Algorithm 1 Efficient computation of the directional transform

1: **Input**: T: separable transform satisfying property (1), θ: chosen rotation angle;
2: Compute the transform coefficients in a separable way: $\hat{\mathbf{x}} = AxA^H$;
3: Compute the rotation matrix $R(\boldsymbol{\theta})$
4: Apply the rotation: $\tilde{\mathbf{x}} = R(\boldsymbol{\theta})$;

2.1. Efficient implementation

It is important to point out that, even if T is a separable transform, the directional transform $T(\boldsymbol{\theta})$ is non-separable. Since the complexity of a non-separable transform is much higher than the one of a separable transform, this can be an important drawback for the proposed directional transform. However, we note that we can define an efficient way for computing the transform coefficients of $T(\boldsymbol{\theta})$ by leveraging the separability of T in the following way

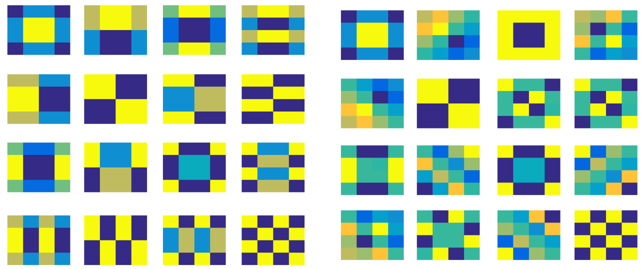
$$\tilde{\mathbf{x}} = T(\boldsymbol{\theta})\mathbf{x} = R(\boldsymbol{\theta})T\mathbf{x} = R(\boldsymbol{\theta})\hat{\mathbf{x}}, \qquad (5)$$

where $\tilde{\mathbf{x}}$ and $\hat{\mathbf{x}}$ are the transform coefficients of the signal $\mathbf{x}$ obtained using respectively $T(\boldsymbol{\theta})$ and T. Equation (5) shows that we can drastically reduce the complexity by first computing in a separable way the coefficients $\hat{\mathbf{x}}$ corresponding to the separable transform T, and then we can obtain the coefficients $\tilde{\mathbf{x}}$ of $T(\boldsymbol{\theta})$ just multiplying $\hat{\mathbf{x}}$ by the sparse matrix $R(\boldsymbol{\theta})$. Algorithm 1 summarises the proposed procedure for efficiently computing the directional transform.

2.2. Example of directional transform

In the previous part of this section, we have defined a general framework for efficiently designing a directional transform by rotating the basis vectors of a given transform in any chosen direction. It is important to underline that the proposed method could be applied to any two-dimensional separable transform that satisfies the property described in (1). The number of transforms that could be interested by the proposed framework is very large. Apart from the classical 2D-DCT, a few examples of possible transforms that could be interested by the proposed technique are 2D-DST, wavelets, 2D discrete Hartley transform [12, 13] and integer transforms like the integer DCT and the integer DST that are typically employed in many image and video compression standards [14].

As an example, here we consider the 2D-DST and we show how to design a directional DST. One of the major applications of the DST is in video compression, since it is used



(a) DST (b) Directional DST with $\theta = \frac{\pi}{4}$

Fig. 1: Transform basis vectors.

in the High Efficiency Video Coding (HEVC) standard for coding 4×4 blocks [14]. The 1D-DST is defined as follows

$$\hat{x}_k = \sum_{i=0}^{n-1} \alpha_i x_i \sin\left(\frac{\pi}{2n}(2i+1)(k+1)\right),$$

where $\alpha_i = \sqrt{\frac{1}{n}}$ if $i = n-1$, otherwise $\alpha_i = \sqrt{\frac{2}{n}}$. Then, since the 2D-DST is a separable transform, we can define the 2D-DST transform matrix T as $T = A \otimes A$, where A is the 1D-DST transform matrix. In this way, we can apply the proposed method as defined in (3) for designing a directional 2D-DST. In Fig. 1, we show an example of a directional 2D-DST transform matrix, where we rotate all the basis vectors by $\frac{\pi}{4}$. As can be seen, all the off-diagonal elements $t^{(l,k)}$, where $k \neq l$, are clearly rotated by the chosen angle.

In the next section, we present an examples of possible application of the proposed framework by focusing on integer transforms. In particular, we define a directional integer DCT and DST for HEVC applications by substituting matrix A in (1) with the integer approximations of the DCT and DST specified in [15, 16].

3. VIDEO COMPRESSION

The most recent video compression standard is HEVC [16], which nearly doubles the rate-distortion performance with respect to the previous H.264/AVC standard [17]. However, the increasing popularity of full-HD videos and emerging Ultra High Definition (UHD) formats leads to a big issue in storing and transmitting video contents. For this reason, there is a continuous effort in enhancing and developing new techniques to improve the coding efficiency. For example, the Joint Video Exploration Team (JVET) of ITU-T VCEG and ISO/IEC MPEG is now developing the future video coding technology, which will improve significantly the compression efficiency with respect to the current HEVC standard [18].

In this Section, we show how to exploit both the directional integer DCT and DST to achieve a coding efficiency improvement. The benefits are due to the fact that using transform basis vectors with different orientations may lead to a sparser signal representation. Therefore, the number of bits needed for encoding the transform coefficients is reduced and the rate-distortion performance is improved.

3.1. Implementation details

Stemming from the proposed framework, we implement the directional integer DCT and DST on top of HEVC, by applying them on all the integer DCT sizes from 4×4 up to 32×32 and on the integer DST of size 4×4, which are specified in the standard. Only the luminance signal has been considered in this work.

Despite of improvements achievable when using different rotation angles for different couples of transform vectors [9], we choose to use one single angle for all the rotation vectors in an image block. The angle is selected in a finite set of q possible angles, uniformly distributed in the interval $[0, \frac{\pi}{2}]$. Thanks to the symmetries of the transform vectors, this interval is enough to represent all the relevant angles. The usage of only one rotation angle for all the transform vectors reduces the encoder complexity because only q integer DCT or DST must be computed per each block. Also, this limits the number of overhead bits used to signal the rotation vector in the bit-stream. Indeed, the signaling is implemented by specifying two additional syntax elements. The first one is a flag at the Coding Unit (CU) level which indicates whether the directional integer DCT and DST are applied or not. When the flag is set to 0, all the Transform Units (TUs) inside the CU are coded without rotation. On the other hand, the second syntax element is used to specify the rotation angle for each TU belonging to the CUs which have the first flag set to 1. It requires $\log_2 q$ bits for each TU which is coded using the directional approach.

Concerning the encoding process, the directional integer DCT and DST of a TU are calculated for each of the q rotation angles using (5). The encoding algorithm chooses the best rotation for each TU by minimizing the rate-distortion cost, which is computed using the Sum of Squared Differences between the original and the reconstructed blocks and the number of bits used in the bit-stream to represent the block. Moreover, the cost associated to the directional transforms is taken into account in the overall rate-distortion process, which chooses the frame partitioning at the CU and TU levels. The resulting formula of the rate-distortion cost is

$$J = D + \lambda \cdot (R_{coeff} + R_{angles}),$$

where D is the distortion introduced by the encoding process, λ is the Lagrange multiplier and R_{coeff} and R_{angles} represent the number of bits required to code the transform coefficient and the rotation angle respectively. Thus, the modified HEVC encoder still produces a valid bit-stream optimized in the rate-distortion sense, which can be correctly decoded by the corresponding modified HEVC decoder.

At the decoder side, the bit-stream is parsed and the additional syntax elements related to the proposed framework are extracted and used to instruct the decoder to perform the inverse directional integer DCT and DST with the correct rotation angle.

3.2. Experimental results

To evaluate the improvements in terms of coding efficiency, we integrate the directional integer DCT and DST in the HEVC encoder software model version HM 16.6 [19] and we run coding experiments according to the HEVC common test conditions (CTC) [20]. The experiments are performed on all the video sequences specified in the CTC, which are classified by decreasing resolution from class A down to E, while the last class F contains screen content of different resolutions. To address the recent trend in resolution increase, also classes A1 and A2 from the latest JVET common test conditions [21] are used in our experiments, since they provide UHD 4K video sequences which were not included in the HEVC CTC. The encoder has been configured according to the All Intra, Low Delay and Random Access main settings using 10 bits as internal bit-depth.

The coding efficiency of the proposed encoder with directional integer DCT and DST is evaluated using the Bjøntegaard Delta Bit-Rate (BDBR) metric [22], using as reference method the original HEVC encoder HM-16.6. Therefore, negative values mean bit-rate savings, hence improved coding efficiency with respect to the HEVC standard, whereas positive values denote rate-distortion losses. Each rate-distortion curve is generated by performing coding experiments by using four different quantization parameters, namely 22, 27, 32 and 37, as specified in the CTC. The combined PSNR_{YUV} metric provided by the encoder is used as the measure of the quality.

Table 1 reports the BDBR for the video sequences of the HEVC and JVET CTC classes encoded with the three configurations. As can be observed from Table 1a, the coding efficiency improves when increasing the number of rotation angles, as expected. This is due to the fact that the encoder can choose the best transform in a larger set of candidates, so it can code the residual signal using a sparser representation. However, using more than 16 angles the rate-distortion does not improve further, because the performance gain is outweighed by the increased bit-rate used to signal the rotation angles in the bit-stream. Moreover, it is worth noting that the benefits of employing directional transforms are more significant for high resolution video sequences, which show large uniform regions. These areas of the frames are usually coded using larger TUs, for which the directional approach is very efficient. Indeed, since the bit-rate overhead of $\log_2 N$ bits to signal the rotation angle is constant for all the transform sizes, more benefits come when larger TUs are employed.

The best coding efficiency improvements with respect to HEVC are achieved when the All Intra configuration is employed (see Table 1a). As reported in Table 1b and 1c, smaller rate-distortion benefits are also observed for high resolution video sequences when encoded in the Low Delay and Random Access configurations, whereas this technique does not provide significant coding efficiency improvements for the other classes. This large difference in terms of coding efficiency is due to the fact that the Low Delay and Random Access configurations employ motion estimation and compensation to further enhance the prediction capability of the encoder. This results in a prediction signal which is more likely to be zero, thus lowering the impact of good transform coding techniques on the overall encoding process. On the other hand, the All Intra configuration employs only intra prediction techniques. In this way, the encoder produces a residual signal that still shows higher spatial redundancy, which is exploited by our proposed directional transforms.

Table 1: BDBR [%] Comparison of the proposed directional Transform versus HEVC for Video Compression Applications.

(a) All Intra

Class	$q = 2$	$q = 4$	$q = 8$	$q = 16$
A1	-0.433	-0.572	-0.628	-0.645
A2	-0.514	-0.618	-0.634	-0.633
A	-0.378	-0.577	-0.682	-0.716
B	-0.305	-0.433	-0.472	-0.464
C	-0.277	-0.325	-0.322	-0.302
D	-0.288	-0.379	-0.364	-0.339
E	-0.080	-0.106	-0.101	-0.089
F	-0.176	-0.208	-0.205	-0.220

(b) Low Delay

Class	$q = 2$	$q = 4$	$q = 8$	$q = 16$
A1	-0.040	-0.032	0.003	-0.022
A2	-0.084	-0.097	-0.055	-0.090
A	-0.213	-0.291	-0.243	-0.309

(c) Random Access

Class	$q = 2$	$q = 4$	$q = 8$	$q = 16$
A1	-0.111	-0.040	-0.014	-0.035
A2	-0.218	-0.256	-0.178	-0.236
A	-0.116	-0.243	-0.124	-0.173

4. CONCLUSIONS

In this paper, we have proposed a new method for designing a directional transform starting from any two-dimensional separable transform. The proposed technique is highly general and can be applied to a wide range of transforms. As an example of possible application, we have defined a directional integer DCT and DST and we have implemented them in the HEVC video coding standard. The experimental results show that the proposed directional framework can improve the coding efficiency of the HEVC standard.

5. REFERENCES

[1] B. Zeng and J. Fu, "Directional discrete cosine transformsa new framework for image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 18, no. 3, pp. 305–313, 2008.

[2] H. Xu, J. Xu, and F. Wu, "Lifting-based directional dct-like transform for image coding," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 10, pp. 1325–1335, 2007.

[3] E. Alshina, A. Alshin, and F. C. Fernandes, "Rotational transform for image and video compression," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2011, pp. 3689–3692.

[4] E.J. Candes and D.L. Donoho, "Curvelets: A surprisingly effective nonadaptive representation for objects with edges," Tech. Rep., DTIC Document, 2000.

[5] E. Le Pennec and S. Mallat, "Sparse geometric image representations with bandelets," *IEEE Transactions on Image Processing*, vol. 14, no. 4, pp. 423–438, 2005.

[6] M. N. Do and M. Vetterli, "The contourlet transform: an efficient directional multiresolution image representation," *IEEE Transactions on Image Processing*, vol. 14, no. 12, pp. 2091–2106, 2005.

[7] D.I. Shuman, S.K. Narang, P. Frossard, A. Ortega, and P. Vandergheynst, "The emerging field of signal processing on graphs: extending high-dimensional data analysis to networks and other irregular domains," *IEEE Signal Processing Magazine*, vol. 30, no. 3, pp. 83–98, 2013.

[8] G. Fracastoro and E. Magli, "Steerable discrete cosine transform," in *Proc. IEEE International Workshop on Multimedia Signal Processing (MMSP)*, 2015, pp. 1–6.

[9] G. Fracastoro, S. M. Fosson, and E. Magli, "Steerable discrete cosine transform," *IEEE Transactions on Image Processing*, vol. 26, no. 1, pp. 303–314, 2017.

[10] G. Fracastoro and E. Magli, "Subspace-sparsifying steerable discrete cosine transform from graph Fourier transform," in *Proc. IEEE International Conference on Image Processing (ICIP)*, 2016, pp. 1534–1538.

[11] G. Fracastoro and E. Magli, "Steerable discrete Fourier transform," *IEEE Signal Processing Letters*, vol. 24, no. 3, pp. 319–323, 2017.

[12] R. N. Bracewell, "Discrete Hartley transform," *Journal of the Optical Society of America*, vol. 73, no. 12, pp. 1832–1835, 1983.

[13] A. B. Watson and A. Poirson, "Separable two-dimensional discrete hartley transform," *Journal of the Optical Society of America*, vol. 3, no. 12, pp. 2001–2004, 1986.

[14] M. Wien, *High Efficiency Video Coding*, Springer, 2015.

[15] M. Budagavi, A. Fuldseth, G. Bjntegaard, V. Sze, and M. Sadafale, "Core Transform Design in the High Efficiency Video Coding (HEVC) Standard," *IEEE Journal of Selected Topics in Signal Processing*, vol. 7, no. 6, pp. 1029–1041, Dec 2013.

[16] 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, Dec 2012.

[17] J. R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the Coding Efficiency of Video Coding Standards - Including High Efficiency Video Coding (HEVC)," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 22, no. 12, pp. 1669–1684, Dec 2012.

[18] J. Chen, Y. Chen, M. Karczewicz, X. Li, H. Liu, L. Zhang, and X. Zhao, "Coding tools investigation for next generation video coding based on HEVC," in *Proc. SPIE 9599, Applications of Digital Image Processing XXXVIII*, Sept 2015, number 95991B.

[19] Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11, *HM 16.6 Reference Software*.

[20] F. Bossen, "Common test conditions and software reference configurations," Doc. JCTVC-L1100, Jan 2013.

[21] K. Suehring and X. Li, "JVET common test conditions and software reference configurations," Doc. JVET-B1010, Feb 2016.

[22] G. Bjøntegaard, "Calculation of Average PSNR Differences Between RD Curves," Doc. VCEG-M33, Apr 2001.