Adaptive Boundary Dependent Transform Optimization for HEVC

Juanting Fan[1, 2], Jicheng An[3], Shanshe Wang[1, 2], Nan Zhang[4], Ruiqin Xiong[1, 2], Siwei Ma[1, 2], Shawmin Lei[3]
[1]Institute of Digital Media, Peking University, Beijing, China
[2]Cooperative Medianet Innovation Center, Shanghai, China
[3]MediaTek Inc., Beijing, China
[4]School of Biomedical Engineering, Capital Medical University, Beijing, China
{jtfan, sswang, rqxiong, swma}@jdl.ac.cn, {jicheng.an, shawmin.lei}@mediatek.com, zhangnan@ccmu.edu.cn

***Abstract*—High Efficiency Video Coding (HEVC) adopts hybrid transform coding scheme to improve the transform performance. However, it does not consider the influence of the prediction unit boundary. This paper proposes an adaptive boundary dependent transform optimization scheme for HEVC. Based on the transform unit boundary type identification by checking whether it lies at a prediction unit boundary or not, some additional transform kernels, including DCT-IV, are incorporated for transform. Moreover, the additional adopted transform kernels are generated by re-using the existing kernels in HEVC. Therefore, the butterfly structure for transform implementation can be preserved to facilitate parallel computing, and computational complexity increasing can be avoided. Experimental results show that the coding performance improvements can be up to 0.96% and 0.64% for low delay P and random access testing configurations respectively.**

I. INTRODUCTION

Transform coding is widely adopted in image and video compression. Its main aim is to reduce the inherent spatial redundancy between adjacent pixels. It has been proved that the Karhunen-Loeve Transform (KLT) is the optimal transform coding method for video and image compression that can reduce the correlation [1]. However, practical use of KLT is limited due to its high computational complexity. Discrete Cosine Transform (DCT) and Discrete Sine Transform (DST) have been testified to offer good energy compaction for image compression and can achieve coding performance verge on KLT [2]. However, DCT and DST consist of a bunch of different transform kernels, it is always a popular research topic to determine the optimal transform kernel [3].

In the past researches, it has been studied that a very few typical transform kernels could be practically adopted in image and video coding. The DCT-II is one of these. It was first adopted into H.264/AVC, bounded with a hybrid transform coding scheme with typical block sizes from 4×4 to 16×16 depending on the size of prediction blocks after intra/inter prediction [4] [5]. And it was definitely used for High Efficiency Video Coding (HEVC) as well, with an extensional range of block size, including 4×4, 8×8, 16×16 and 32×32 [6]-[8].

In recent researches, many research efforts have been devoted to optimizing the transform kernel determination to fully exploit signal correlation to achieve performance improvement. In [9] and [10], based on the distribution of intra-predicted residuals which is denoted as boundary information, an optimization scheme for transform kernel determination is proposed. It is stated that DST-VII is more effective for the transform unit at the boundary. DST-VII was finally adopted to HEVC [8]. However, due to high complexity of DST-VII, it was only applied for 4x4 Transform Units (TU). Thus, the overall HEVC intra-predicted hybrid transform coding scheme switches between DST-VII and the conventional DCT-II for 4x4 transform units. For other transform units, only DCT-II is employed. In [11], Mode Dependent Directional Transform (MDDT) is fully investigated and it can be seen that MDDT can bring much performance improvement with high computational complexity. In [12], Zhao etc. propose a Rate Distortion Optimization Transform (RDOT) by providing more transform kernels. Better performance improvement can be achieved with much higher computational complexity.

In order to avoid high computational complexity, only 4-point (4×4) fast DST-VII is adopted into HEVC [8], and Cohen etc. propose a simple MDDT scheme for intra-predicted transform with little performance loss [13]. In [14] [15], fast algorithms for DCT and IDCT (Inverse Discrete Cosine Transform) are proposed for butterfly structured implementation considering parallel computing and hardware design.

So, the adoption of multiple types of transform kernels improves compression performance significantly. However, computational complexity increases much. In this paper, we propose a Boundary Dependent Transform (BDT) optimization scheme. For the given inter prediction unit, different DST/DCT kernels are adaptively determined for transform based on the boundary information to improve the coding performance without any computational complexity increasing.

The rest of this paper is organized as follows. Section II gives a detailed description of inter-predicted residual analysis. Then, the proposed adaptive boundary dependent transform scheme is provided in section III. Section IV shows our experimental results. Finally, we conclude this paper in section V.

II. INTER-PREDICTED RESIDUAL ANALYSIS

Inter prediction plays a key role in video coding. It can improve coding performance significantly. However, for a

Prediction Unit (PU), the distribution of inter-predicted residuals is uneven.

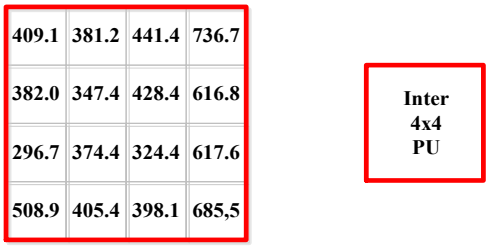


Figure 1. The square error distribution of inter 4×4 PU (POC (Picture Order Count) = 2)

Fig. 1 shows the square error distribution for all inter 4×4 PUs in a frame of sequence *BasketballPass* under low delay P configuration. Each data is the average of square error of the same position in a 4×4 PU using luminance component. It can be seen that the residual values near the boundary are much larger than those in the middle of the PU.

When the size of PU is larger than that of TU in the current CU (Coding Unit), similar observation can be achieved on the TU boundary as shown in Fig. 2. It can also be seen that the inter-predicted square errors near PU boundaries are larger than those near TU boundaries but not near the boundaries of PU.

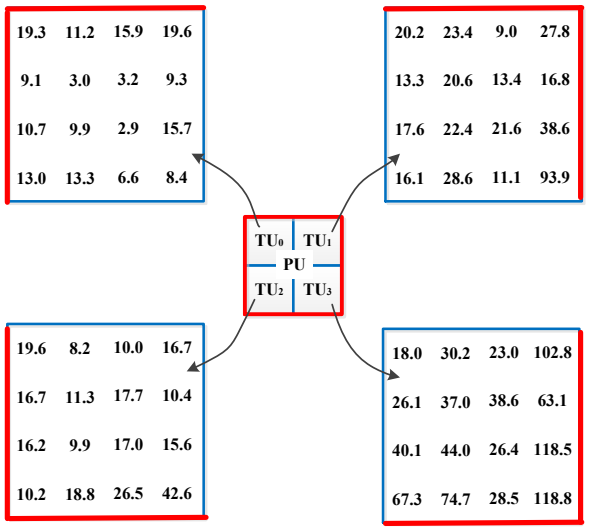


Figure 2. The square error distribution of inter 8×8 PU when the size of its optimal TU is 4x4 and POC of this frame equals to 4.

For this boundary based uneven distribution in a PU as mentioned before, the specific transform kernel, DST-VII, is more effective for that its basis vectors tend to vanish at the known boundaries and maximize energy at unknown boundaries [3]. However, considering the high computational complexity when DST-VII is applied to TUs with larger size, like 8×8 and 16×16, we utilize an alternative transform kernel, DCT-IV, to replace DST-VII. So performance improvement and low computational complexity can be expected

III. Proposed Boundary Dependent Transform

In this section, an adaptive boundary dependent transform scheme is proposed to adapt the above-mentioned uneven error distribution in a single PU. Firstly, for the PU with different types of boundaries, different transform kernels are utilized for transform to improve the coding performance. Then, a particular method is provided to derive additional needed transform kernels to avoid the computational complexity increasing.

A. Boundary Based Transform Kernels Determination

In [9], it is stated that DST-VII is more appropriate to compact energy for residual blocks with known boundary, compared with other transform kernels. However, the disadvantage of using DST is that the computational complexity is high. The bigger TU size is, the greater computational complexity is. Moreover, butterfly structures are difficult to implement for DST, which makes parallel processing hardly possible. These disadvantages limit one and only one 4-point fast DST-VII processed in HEVC. In [3], it is proved that DCT-IV plays a significant role in gathering energy for matrices with obvious boundary information, as the same as DST-VII partly, and the computational complexity for DCT-IV transform is as good as that for the original DCT-II transform on account of parallelizablity. Thus, for inter-predicted residual blocks we replace DCT-IV for DCT-II in large size of TU, such as 8×8 and 16×16, and employ DST-VII instead of DCT-II for 4×4 TU when the specific situations occur, which is followed.

In order to improve the coding performance and flexible parallelization, we propose to utilize DCT-II, DCT-IV and DST-VII for inter-predicted transform according to PU and TU boundaries. The detailed scheme can be presented as follows.

In HEVC, two-dimensional transform is incorporated by implementing horizontal transform and vertical transform. Therefore, the proposed scheme contains the following two aspects.

1) Horizontal transform. As illustrated in Table I, for each 4×4 TU, when its left boundary is not a boundary of PU and its right boundary is a boundary of PU, the DST-VII kernel will be applied directly for transform. On the contrary, when the left boundary is a boundary of PU and its right boundary is not a boundary of PU, the Flipping DST-VII (F-DST-VII) kernel is utilized to replace DCT-II kernel. For other circumstances, DCT-II remains unchangeable. For each 8×8 or 16×16 TU, F-DCT-IV (Flipping DCT-IV) and DCT-IV are directly utilized to substitute for DST-VII and F-DST-VII respectively.

2) Vertical transform. As shown in Table II, for each 4×4 TU, when its top boundary is not a boundary of PU and its bottom boundary is a boundary of PU, the DST-VII kernel will be applied in transform processing. On the contrary, when its top boundary is a boundary of PU and its bottom boundary is not a boundary of PU, the F-DST-VII kernel replaces DCT-II kernel either. Similarly, DCT-II is unchanged when both top and bottom boundaries are boundaries of PU or neither of them is not a boundary of PU. For each 8×8 or 16×16 TU, F-DCT-IV and DCT-IV substitute for DST-VII and F-DST-VII respectively.

In addition, NPU in Table I and Table II indicts that the boundary of TU is not a boundary of PU, while PU indicates a PU-boundary.

TABLE I. TRANSFORM KERNELS DETERMINATION FOR HORIZONTAL TRANSFORM

TU Boundary		4×4 TU	TU Boundary		8×8/16×16 TU
Left	*Right*		*Left*	*Right*	
NPU	PU	DST-VII	NPU	PU	F-DCT-IV
PU	NPU	F-DST-VII	PU	NPU	DCT-IV
PU	PU	DCT-II	PU	PU	DCT-II
NPU	NPU	DCT-II	NPU	NPU	DCT-II

TABLE II. TRANSFORM KERNELS DETERMINATION FOR VERTICAL TRANSFORM

TU boundary		4×4 TU	TU boundary		8×8/16×16 TU
Top	*Bottom*		*Top*	*Bottom*	
NPU	PU	DST-VII	NPU	PU	F-DCT-IV
PU	NPU	F-DST-VII	PU	NPU	DCT-IV
PU	PU	DCT-II	PU	PU	DCT-II
NPU	NPU	DCT-II	NPU	NPU	DCT-II

Fig. 3 shows an example of the proposed BDT for different PU and TU partitions. Once a 16×16 inter PU is divided into four 8×8 TUs by quad-tree, naming TU_0, TU_1, TU_2 and TU_3, each TU employs newly transform kernel for horizontal or vertical transform according to the determination strategy.

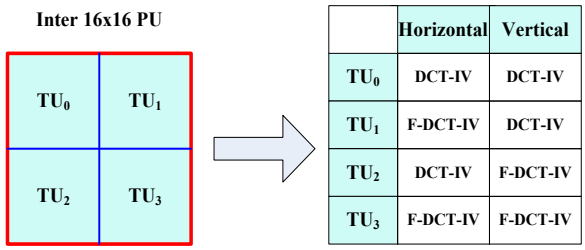


Figure 3. An example of BDT for an Inter 16×16 PU.

B. Generation of Additional Transform Kernels

In the proposed BDT scheme, additional transform kernels are incorporated, including F-DST-VII, DCT-IV and F-DCT-IV. If all the additional kernels are conducted separately with other kernels, much computational complexity will be brought in. In our paper, we propose to utilize the original kernels in HEVC, including 4-point fast DST-VII, 16-point and 32-point DCT-II transform kernels, to generate the additional kernels. Thus, the increasing computational complexity can be definitely avoided.

For the generation of F-DST-VII transform kernel, 4-point fast DST-VII for intra-predicted transform in HEVC is employed as follows. For the forward transform, firstly, we flip the input data. Then the DST-VII is re-used for forward F-DST-VII as illustrated in Fig. 4. Furthermore, for inverse F-DST-VII, we re-use the DST-VII first and then flip the output data from left to right, as shown in Fig. 5.

For the generation of DCT-IV transform kernel, DCT-II can be employed, since their definitions have much correlation [16]. Equation (1) and (2) show the definitions of N-point (N×N) DCT-II and DCT-IV of signal $f[n]$, respectively.

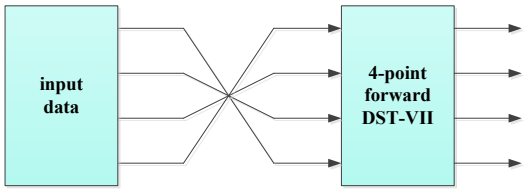


Figure 4. 4-point forward F-DST-VII using 4-point DST-VII.

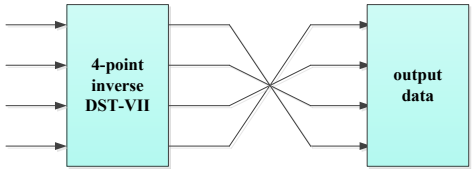


Figure 5. 4-point inverse F-DST-VII using 4-point DST-VII.

$$\hat{f}_{\text{DCT-II}}[k]=\lambda_k\frac{2}{\sqrt{N}}\sum_{n=0}^{N-1}f[n]\cos[\frac{k\pi}{N}(n+\frac{1}{2})],\ k=0,1,2,...,N-1,\qquad \lambda_k=\begin{cases}2^{-0.5}, & k=0\\ 1, & k\neq 0\end{cases}. \tag{1}$$

$$\hat{f}_{\text{DCT-IV}}[k]=\frac{2}{\sqrt{N}}\sum_{n=0}^{N-1}f[n]\cos[\frac{\pi}{N}(k+\frac{1}{2})(n+\frac{1}{2})],\ k=0,1,...,N-1. \tag{2}$$

Considering the parity of k, (1) can be equally presented as follows.

$$\hat{f}_{\text{DCT-II}}[2k]=\lambda_k\frac{2}{\sqrt{N}}\sum_{n=0}^{N/2-1}(f[n]+f[N-1-n])\cos[\frac{k\pi}{N/2}(n+\frac{1}{2})],\quad k=0,1,2,...,N/2-1. \tag{3}$$

$$\hat{f}_{\text{DCT-II}}[2k+1]=\frac{2}{\sqrt{N}}\sum_{n=0}^{N/2-1}(f[n]-f[N-1-n])\cos[\frac{\pi}{N/2}(k+\frac{1}{2})(n+\frac{1}{2})],\quad k=0,1,2,...,N/2-1. \tag{4}$$

Combining (1) with (3), it can be concluded that the coefficients of the N-point DCT-II with even indices are equal to the coefficients of the N/2-point (N/2×N/2) DCT-II of signal $f[n]+f[N-1-n]$. And with combination of (1), (2) and (4), it can be seen that the coefficients of the N-point DCT-II with odd indices are equal to that of the DCT-IV of N/2-point signal $f[n]-f[N-1-n]$. Therefore, the 8-point DCT-IV can share the partial logic of 16-point DCT-II, and 16-point DCT-IV can share the partial logic of 32-point DCT-II.

Then, DCT-IV could be implemented efficiently by butterfly structure with better parallelization as the same as DCT-II, and forward and inverse transform using DCT-IV are depicted in Fig. 6 and Fig. 7. Therefore, the low implementation cost for DCT-IV can be expected. And no more computational complexity will be added.

For the generation of F-DCT-IV transform kernel, similar to F-DST-VII, the newly DCT-IV is employed. For the forward transform, we reorder the input data first. Then DCT-IV is re-used for 8-point and 16-point forward F-DCT-IV transform as illustrated in Fig. 8. Moreover, for 8-point and 16-point inverse

F-DCT-IV, we re-use the DCT-IV first and then reorder the output data as shown in Fig. 9.

Based on the above analysis, all the additional transform kernels for proposed BDT can be generated by re-using the logic of the existing transform kernels in HEVC. Thus, computational complexity can be reduced efficiently.

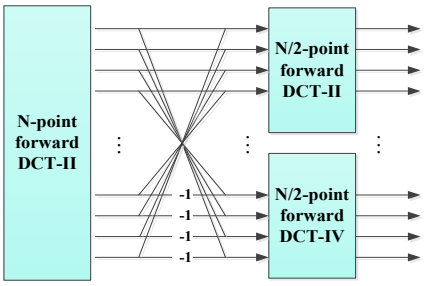


Figure 6. N-point forward DCT-II by using N/2-point DCT-II and N/2-point DCT-IV.

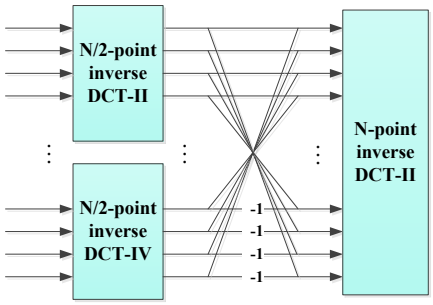


Figure 7. N-point inverse DCT-II by using N/2-point DCT-II and N/2-point DCT-IV.

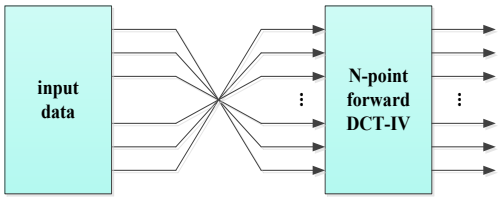


Figure 8. 8-point/16-point forward F-DCT-IV using 8-point/16-point DCT-IV.

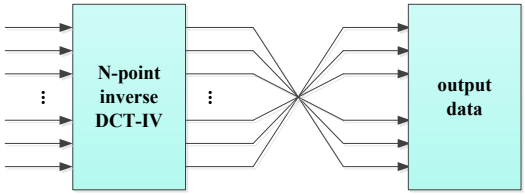


Figure 9. 8-point/16-point inverse F-DCT-IV using 8-point/16-point DCT-IV.

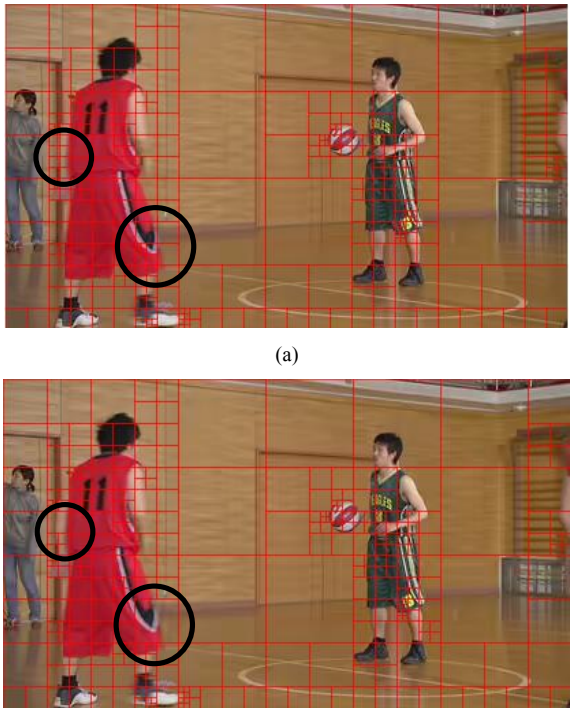
IV. EXPERIMENTAL RESULTS AND ANALYSIS

A. Rate-distortion Performance

In order to verify the proposed scheme, experiments are implemented on top of the HM16.2. The testing configurations including low delay P (LDP) and random access (RA) are the common test conditions as in [17]. To evaluate the rate-distortion performance, all sequences under common test conditions are coded using QP values of 22, 27, 32 and 37, respectively. Table III shows the rate-distortion performance using BD-rate as measurement [18].

TABLE III. RATE-DISTORTION PERFORMANCE OF THE PROPOSED BDT FOR DIFFERENT CONFIGURATIONS

Resolution	Sequence		LDP (BD-rate)	RA (BD-rate)
1600P	CLASS A	Traffic	-0.71%	-0.32%
		PeopleOnStreet	-0.96%	-0.64%
1080P	CLASS B	Kimono	-0.73%	-0.35%
		ParkScene	-0.94%	-0.44%
		Cactus	-0.61%	-0.19%
		BasketballDrive	-0.19%	-0.05%
		BQTerrace	-0.37%	-0.11%
480P	CLASS C	BasketballDrill	-0.47%	-0.18%
		BQMall	-0.68%	-0.23%
		PartyScene	-0.32%	-0.15%
		RaceHorses	-0.44%	-0.25%
240P	CLASS D	BasketballPass	-0.38%	-0.12%
		BQSquare	-0.26%	-0.18%
		BlowingBubbles	-0.49%	-0.21%
		RaceHorsesC	-0.44%	-0.28%
720P	CLASS E	FourPeople	-0.64%	-0.13%
		Johnny	-0.13%	-0.01%
		KristenAndSara	-0.46%	-0.08%
Average			***-0.51%***	***-0.22%***



(a)

(b)

Figure 10. TU partition distribution for coded stream. (a) BDT. (b) HM16.2

From Table III, it can be seen that the proposed BDT scheme can improve the coding performance much for all the sequences, compared with the reference software HM16.2.

For LDP configuration, the performance gain is 0.51% on average. Especially, for sequences in *PeopleOnStreet* and *ParkScene*, the performance gain can be approaching to 1%. For RA configuration, the average performance gain is 0.22%. Fig. 10 shows the TU partition distribution of *BasketballPass* coded streams by using the BDT scheme and the HM16.2 anchor. It can be seen that the TU partition distribution with BDT scheme is more appropriate for the actual texture features.

B. Coding Complexity Analysis

In this sub-section, coding complexity analysis is provided. Complexity reduction is calculated as follows.

$$T = \left(\frac{T_{anchor} - T_{proposed}}{T_{anchor}} \right) \times 100\ \% \qquad (6)$$

where T_{anchor} indicates the encoding time for HM16.2 anchor, $T_{proposed}$ is the encoding time for the proposed BDT scheme. The positive values of T indicate that the proposed BDT scheme achieves complexity reduction for encoder-side.

Table IV provides that coding complexity analysis of the proposed BDT, compared with HM16.2 anchor. It can be seen that our proposed scheme can obtain the coding complexity reduction by 0.4% for both LDP and RA configurations. It can be attributed to the proposed generation scheme for the additional kernels.

TABLE IV. CODING COMPLEXITY ANALYSIS OF BOUNDARY DEPENDENT TRANSFORM

Resolution	Sequence	LDP (%)	RA *(%)*
1600P	CLASS A	1%	0%
1080P	CLASS B	-1%	0%
480P	CLASS C	1%	0%
240P	CLASS D	1%	1%
720P	CLASS E	0%	1%
OVERALL		***0.4%***	***0.4%***

V. CONCLUSION

In this paper, an adaptive boundary dependent transform is proposed to improve the coding performance of HEVC. Depending on whether the TU boundary is a PU boundary or not, the proposed transform kernels are adaptively utilized for the inter-predicted residual blocks. Experimental results show that the proposed BDT scheme can improve the coding performance for all testing sequences, compared to HM16.2. Moreover, the proposed scheme does not bring coding complexity increasing.

ACKNOWLEDGEMENT

This work was supported in part by the Major State Basic R esearch Development Program of China (2015CB351800) and in part by the National Natural Science Foundation of China (61322106, 61390515, 61210005 and 61272255). This work was also supported in part by the Science and Technology Program of Beijing Municipal Commission of Education (KM201310025009).

REFERENCES

[1] N. Ahmed, T. Natarajan, and K. R. Rao, "Discrete cosine transform," IEEE Trans. On Computers, vol. C-23, no. 1, pp. 90 –93, January 1974.

[2] C. Zhang, D. Florêncio, "Analyzing the optimality of predictive transform coding using graph-based models," IEEE Signal Processing Letters, vol. 20, no.1, pp. 106-109, January 2013.

[3] V. Britanak, P. C. Yip, K.R. Rao, "Discrete cosine and sine transforms: general properties, fast algorithms and integer approximations, " Academic Press, 2010.

[4] ITU-T and ISO/IEC, "Advanced Video Coding for Generic Audiovisual Services," ITU-T Recommendation H.264 and ISO/IEC 14496-10(MPEG-4 AVC), Version 8, July 2007.

[5] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira,T. Stockhammer und T. Wedi, "Video coding with H.264/AVC: Tools, Performance, and Complexity," IEEE Circuits and Systems Magazine, vol. 4, no. 1, pp. 7–28, April 2004.

[6] G. J. Sullivan, J. R. Ohm, W.-J. Han, T. Wiegand, "Overview of the High Efficiency Video Coding (HEVC) Standard," IEEE Trans. on Circuits and Systems for Video Technology, vol. 22, no.10, pp.1649–1668, December 2012.

[7] B. Bross, W. J. Han, J. R. Ohm, G. J. Sullivan, Y. K. Wang, T. Wiegand, "High Efficiency Video Coding (HEVC) text specification draft 10 (for FDIS & Consent)," Joint Collaborative Team on Video Coding (JCTVC) of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11 Document JCTVC-L1003, 12th Meeting, Geneva, CH, January 2013.

[8] M. Budagavi, A. Fuldseth, G. Bjontegaard, V. Sze, M. Sadafale, "Core Transform Design for the High Efficiency Video Coding (HEVC) Standard," IEEE Journal of Selected Topics in Signal Processing, vol. 7, no.6, pp. 1029-1041, December 2013.

[9] A. Saxena and F. Fernandes, "CE7: Mode-dependent DCT/DST for intra prediction in video coding," Committee input document JCTVC-D033,ISO/IEC/ITU-T Joint Collaborative Team on Video Coding, Daegu, Korea, January 2011.

[10] F. Fernandes, A. Saxena and Y. Reznik, "Fast transforms for intra-prediction-based image and video coding," in IEEE Data Compression Conference (DCC), pp. 13-22, March 2013.

[11] S. Ma, S. Wang, Q. Yu, J. Si, W. Gao, "Mode Dependent Coding Tools for Video Coding", IEEE Journal of Selected Topics in Signal Processing, vol. 7, no.6, pp. 990-1000, December 2013.

[12] X. Zhao, L. Zhang, S. Ma, W. Gao, "Video coding with rate-distortion optimized transform." IEEE Transactions on. Circuits and Systems for Video Technology, vol. 22, no. 1, pp. 138-151, January 2012.

[13] R. Cohen, C. Yeo, R. Joshi, W. Ding, S. Ma, A. Tanizawa, H. Yang, X. Zhao, L. Zhang, "Tool experiment 7: MDDT simplification," Committee input document JCTVC-B307, ISO/IEC/ITU-T Joint Collaborative Team on Video Coding, Geneva, Switzerland, 2010.

[14] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," IEEE Transactions on Signal Processing, vol. 40, no.9, pp. 2174-2193, September 1992.

[15] T. Ma, C. Liu, Y. Fan, X. Zeng, "Fast 8×8 IDCT algorithm for HEVC," 2013 IEEE 10th International Conference on. ASIC (ASICON), pp. 1-4, 2013.

[16] Y. A. Reznik, "Relationship between DCT-II, DCT-VI, and DST-VII transforms," IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), pp. 5642-5646, 2013.

[17] K. McCann, C. Rosewarne, B. Bross, M. Naccari, K. Sharman, G. J. Sullivan, "High Efficiency Video Coding (HEVC) Test Model 16 (HM16) Improved Encoder Description", Committee input document JCTVC-S1002, France, October 2014.

[18] F. Bossen, "Common HM test conditions and software reference configurations," Committee input document JCTVC-K1100, Shanghai, China, October 2012.