|  |  |
| --- | --- |
| **Joint Collaborative Team on Video Coding (JCT-VC)**  **of ITU-T SG16 WP3 and ISO/IEC JTC1/SC29/WG11**  13th Meeting: Incheon, KR, 18–26 April 2013 | Document: JCTVC-M0033  WG 11Number :\_ m28433 |

|  |  |  |  |
| --- | --- | --- | --- |
| *Title:* | On secondary transforms for Intra\_BL residue | | |
| *Status:* | Input Document to JCT-VC | | |
| *Purpose:* | Proposal | | |
| *Author(s) or Contact(s):* | Ankur Saxena  Felix C. Fernandes  1301 E. Lookout Drive,  Richardson, TX – 75082, USA. | Tel: Email: | 1-9727617761 [asaxena@sta.samsung.com](mailto:asaxena@sta.samsung.com),  [felix.f@sta.samsung.com](mailto:felix.f@sta.samsung.com) |
| *Source:* | Samsung Electronics Co., Ltd | | |

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Abstract

In this contribution, a secondary transform scheme is provided for Intra\_BL residue. A Rate-Distortion based secondary transform scheme is applied for the Luma component of Intra\_BL residue in the enhancement layer (EL) at block sizes 8x8 and larger of scalable video coding. For the Chroma component and 4x4 Luma Intra\_BL residues, the standard DCT and DST (transforms already in SHM 1.0) are retained. Two sets of results are presented in this contribution: first when only one secondary transform is applied, and second when either of the two secondary transforms are applied. Results are also shown when a low-complexity 8x8 Rotational Transform is used as a secondary transform. Simulation results show that average Luma gains of upto 1.4 % and 0.9 % for (BL+EL) are obtained for All Intra 2x and All Intra 1.5x settings respectively for the secondary transforms scheme presented in this contribution.

# Introduction

In JCTVC-L0330 [1], an alternate transform scheme for Intra\_BL residue was provided for Luma component of the Intra\_BL residue. Currently, in the S-HEVC Test Model [3], for the Luma component of the Intra\_BL prediction residue, the DCT Type 2 [4] transform is applied at block sizes 8, 16 and 32. At size 4, the DST-Type 7 [6] transform is used because the coding efficiency of DST-Type 7 and DCT are almost the same in SHM 1.0, but DST is used as the transform for Intra 4x4 Luma Transform Units in the base layer [2]. For the chroma component of Intra\_BL residue, the DCT is always used across all block sizes (Note that unless specified, DCT refers to DCT-Type 2 in this contribution).

In [1,2], it was shown that different transforms other than DCT-Type 2, when applied on the Intra\_BL block residue can provide substantial gains. Specifically, at sizes 4 to 32, transforms DCT- Type 3 and DST-Type 3 were also used in addition to the conventional DCT-Type 2 transform. At the encoder, a Rate-Distortion (R-D) search was performed and one of the following transforms was chosen: DCT-Type 2, DCT-Type 3, and DST-Type 3. The transform choice is signaled by a flag (which can take 3 values for each of the 3 transforms) to the decoder. At the decoder, the flag is parsed and the corresponding inverse transform is used.

However, the scheme in [1,2] will require two additional transform cores at each size 4, 8, 16 and 32. This means 8 new, additional, transform cores (2 transforms x 4 sizes = 8) are required. Furthermore, additional transform cores such as at size 32x32 are extremely expensive to implement in hardware.

Hence, to avoid large alternate transforms for inter-prediction residues, there is a need for a low-complexity transform method that can be applied efficiently on the Intra\_BL residues. To avoid the above shortcomings, and to improve the coding efficiency of SHM, which is the test model for scalable extensions of HEVC, in this contribution we propose to use secondary transforms [5,8], which we describe in detail in the next section.

# Secondary Transforms

## Motivation

## Since most of the energy in the DCT coefficients is concentrated in the upper-left coefficients (see Fig. 1), it is sufficient to perform operations only on a small fraction of the DCT output (typically the 4x4 or 8x8 upper-left block). These operations can be performed by simply using a secondary transform of size 4x4 or 8x8 on the upper-left block. Furthermore, the same secondary transform derived for a block size (such as 8x8) can be applied at higher block sizes (such as 16x16 or 32x32). This re-utilization at higher block sizes is one of the most important advantages of secondary transforms.

Low frequency components

**Fig. 1**: Low-frequency components of a DCT transformed block

## Rate-Distortion based secondary transform for Intra\_BL residue

In [1,2], it was shown that primary alternative transroms: DCT-Type 3, and DST-Type 3 can be used in addition to DCT-Type 2. One of the three possible transforms is selected via a Rate-Distortion search at the encoder, and the choice is signaled at the decoder via a flag. At the decoder, the flag is parsed and the corresponding inverse transform is used.

Next, we show how a low-complexity secondary transform can be used for Intra\_BL residues. In this section, we will show the derivation and usage of secondary transforms via secondary transform sizes of K\*K, and K=4 or 8, but this can be trivially extended to other block sizes.

Let us begin with a secondary transform at size 4. At size 4, we would always like to use DCT-Type 2 as the primary transform.

Corresponding to the DCT-Type 3, we derive a secondary transform as follows:

Let **C** denote the DCT Type-2. DCT Type-3, which is simply the inverse (or transpose) of DCT-Type 2 is then given by **CT** (Note that we ignore the normalization factors such as sqrt (2) in definition of DCT’s, which is a common practice in literature). Also let **S** denote DST-Type 3.

For an alternate primary transfom **A,** an equivalent secondary transform **M,** we have **C \*M = A**,

or M = C T \* A. If alternate transform is DCT-Type 3, i.e., CT , then M= C T \* C T. For DST-Type 3, M would be C T \* S.

The secondary transform matrices (after 7-bit shifts, and rounding) at size 4x4 and 8x8 are:

1. Secondary transform for DCT-T3 at size 4x4 with basis vectors along rows are:

[ 123 24 -24 -5

-24 123 5 -24

24 -5 123 -24

5 24 24 123]

1. Secondary transform for DCT-T3 at size 8x8 with basis vectors along rows are:

[ 120 39 -20 -4 -9 -4 -4 -2

-33 114 37 -29 -1 -11 -2 -3

26 -26 116 22 -33 -2 -10 -1

-9 29 -14 119 4 -33 -2 -7

14 -6 27 -3 120 -15 -30 -3

-1 16 0 26 8 118 -33 -22

8 2 14 6 26 20 114 -45

4 9 7 13 14 27 36 118 ]

1. Secondary transform for DST-T3 at size 4x4 with basis vectors along rows are:

[ 123 -24 -24 5

24 123 -5 -24

24 5 123 24

-5 24 -24 123 ]

1. Secondary transform for DST-T3 at size 8x8 with basis vectors along rows are:

[ 120 -39 -20 4 -9 4 -4 2

33 114 -37 -29 1 -11 2 -3

26 26 116 -22 -33 2 -10 1

9 29 14 119 -4 -33 2 -7

14 6 27 3 120 15 -30 3

1 16 0 26 -8 118 33 -22

8 -2 14 -6 26 -20 114 45

-4 9 -7 13 -14 27 -36 118 ]

**Comments on Secondary Transforms**

1. The secondary transforms derived using DCT-Type 3 and DST-3 have the same coefficients in magnitude, and only a few coefficiencts have alternate signs. This will reduce secondary transform hardware-implementation cost, where the same core with few sign changes can be used as the second secondary transform.
2. It has been shown by Loeffler [7] that an 8x8 DCT-Type 2 can be implemented via 11 multiplications, and 29 additions. DCT-Type 3, which is transpose of DCT-Type 2, can therefore also be implemented via 11 multiplications and 29 additions.

The 8x8 secondary transform M = CT  \* CT, can therefore be considered as a cascade of two DCT’s, and therefore can be computed via 22 multiplications and 58 additions, which is less than a full matrix multiplication at size 8x8, i.e., 64 multiplications and 56 additions.

Similarly, the secondary transform corresponding to DST-Type 3 (which can be obtained by changing signs of some transform coefficients of the previous secondary transform matrix M) can also be implemented via 22 multiplications and 58 additions.

1. For the 4x4 secondary transform, we show a fast factorization in Section 2.4.

## Rotational Transforms

Rotational transforms [9] were derived for Intra residue in the context of HEVC. In fact, the rotational transforms are special cases of secondary transforms, and can also be used as secondary transforms for Intra\_BL residues.

Specifically, in this contribution, we use the following rotational transform matrices from [9] as secondary transforms:

1. Rotational Transform 1

{ 87, -93, 12, 0, 0, 0, 0, 0,},

{ 91, 79, -44, 0, 0, 0, 0, 0,},

{ 25, 38, 120, 0, 0, 0, 0, 0,},

{ 0, 0, 0, 118, -50, -5, 0, 0,},

{ 0, 0, 0, -50, -118, -13, 0, 0,},

{ 0, 0, 0, 1, 14, -128, 0, 0,},

{ 0, 0, 0, 0, 0, 0, 128, 0,},

{ 0, 0, 0, 0, 0, 0, 0, 128,},

1. Rotational Transform 2, which is the transpose of Rotational Transform 1

{ 87, 91, 25, 0, 0, 0, 0, 0,},

{ -93, 79, 38, 0, 0, 0, 0, 0,},

{ 12, -44, 120, 0, 0, 0, 0, 0,},

{ 0, 0, 0, 118, -50, 1, 0, 0,},

{ 0, 0, 0, -50, -118, 14, 0, 0,},

{ 0, 0, 0, -5, -13, -128, 0, 0,},

{ 0, 0, 0, 0, 0, 0, 128, 0,},

{ 0, 0, 0, 0, 0, 0, 0, 128,},

Note that due to the structure of rotational transform matrices, there are only 20 non-zero elements at size 8x8. So they can be implemented via only 20 multiplications, and 12 additions, which is much smaller than 22 mults and 58 adds that would be required for 8x8 secondary transform implementation in the previous section. Further the number of operations for the 8x8 rotational transform can be reduced via using a lifting implementation of rotational transforms as described in [9].

We next describe how the secondary transform would be implemented at block sizes 8, 16 and 32 in a practical video codec at the encoder and decoder end in Embodiments 2 and 3.

**Encoder**

1. Select one from the following 3 transform choices (a), (b) or (c) for all the Transform Unit’s in a Coding Unit (CU) via a Rate-distortion search:
2. 2-d DCT
3. 2-d DCT followed by secondary transform M1
4. 2-d DCT followed by secondary transform M2
5. Based on the choice of transform scheme in (1) above, encode a flag with the appropriate transform choice in (1): DCT, or DCT + M1, or DCT + M2
6. Encode the coefficients by transform choice in (1), and encode the flag with appropriate value (note that in [1,2] there is a mechanism to not encode the flag under certain conditions. We can follow the same convention here as well.)

**Decoder**

1. Decode the flag, obtain the transform choice: DCT, or DCT + M­1, or DCT + M2
2. If transform choice == DCT {

Apply Inverse DCT }

else if (transform choice == secondary transform 1){

Apply inverse secondary transform M1

Apply Inverse DCT

}

else (transform choice == secondary transform 2){

Apply inverse secondary transform M2

Apply Inverse DCT

}

// End

Note that, in the above algorithm, we have assumed only 2 secondary transform choices. This can be trivially extended to different transform sizes, and block sizes: (for example, the secondary transform can be applied only at block sizes 16, 32 etc.), and size of secondary transform (this can KxK, where K =4, 8 etc.), and also a Rotational transform core can also be used as a secondary transform.

**Trigger Conditions**

Inverse Quantizer

**Decoder Operations**

NxN Inverse DCT

TU >=8 ; Mode == Intra\_BL ; Text\_Luma;

and useSecTransFlag ==1

K-Point Inverse 2-d Secondary Transform

Inverse Quantizer

NxN Inverse DCT

## Fast factorization of 4x4 secondary transforms

In general, the 4x4 matrix M will require 16 multiplications and 12 additions for implementation. In this section, we show that actually the implementation of M (and hence its transpose MT  = C \* C can be performed in only 6 multiplications, and 14 additions.

For the multiplications, which reduce from 16 to 6, this is [(16-6)/16] \* 100 = 62.5 % reduction. For the additions, there is a slight increase of [(14-12)/12] \* 100 = 16.67 %. But implementation complexity, especially from mults is the main barrier to transform deployment in image/video coding. The 4x4 matrix can be implemented via the following method for 4-point input **x**, and 4-point output **y.**



Note that there are 4 additional shifts due to rounding operations in the computation of the transform, but shifts are generally cheap in hardware as compared to mults and adds.

* + 1. **Further comments on fast factorization**

1. The 4x4 secondary matrix obtained from DST-Type 3, i.e., M2 can similarly be evaluated using only 6 mults, and 14 adds, since some of its elements have sign-changes than M.
2. The inverse of the matrices M and M2 can also be computed using 6 mults, and 14 adds, since they are simply the transpose of M and M2 respectively, and the steps in computation of MT can simply be reversed as compared to that of M.

# Simulation Results

In this section, we present 4 set of results when the 8x8 secondary transforms are applied to Intra\_BL residue at sizes 8x8, 16x16 and 32x32 Luma component. For the other block sizes, we retain the standard transforms in SHM 1.0. Detailed results for the tests are in the attached excel sheets. We would like to thank Qualcomm for cross-checking our proposal in JCTVC-M0307.

**Test 1: One secondary transform presented in (b) in Section 2.2**

**Test 2: Two secondary transforms presented in (b) and (d) in Section 2.2**

**Test 3: One rotational transform presented in (e) in Section 2.3**

**Test 4: One rotational transform and its transpose presented in (e) and (f) in Section 2.3**

**Test 5: Two secondary transforms presented in (b) and (d) in Section 2.2 on TU sizes 16 and 32 only**

**Test 6: Two rotational transform presented in (e) and (f) in Section 2.3 on TU sizes 16 and 32 only**

**Test 1:** Anchor is SHM 1.0. Test is one secondary transform



**Test 2:** Anchor is SHM 1.0. Test is two secondary transforms



**Test 3:** Anchor is SHM 1.0. Test is one Rotational transform



**Test 4:** Anchor is SHM 1.0. Test is one Rotational transform, and its transpose



**Test 5:** Anchor is SHM 1.0. Test is two secondary transform on TU sizes 16 and 32 only



**Test 6:** Anchor is SHM 1.0. Test is two rotational transform on TU sizes 16 and 32 only



From the above experimental results, average BD rate gains of 1.4%, 0.9 %, 0.9 %, 0.5 % and 0.9 % for All Intra 2x, All Intra 1.5x, Random Access 2x, Random Access 1.5x, and Random Access SNR settings for the case of two 8x8 secondary transforms (Table 2). In the case, a single secondary transform is used (Table 1), the gains are 1.0 %, 0.6%, 0.6 %, 0.3 %and 0.6 % respectively for the above 5 settings.

For the Rotational Transform (Table 3), average BD rate gains of 0.8%, 0.4 %, 0.4 %, 0.2 % and 0.4 % for All Intra 2x, All Intra 1.5x, Random Access 2x, Random Access 1.5x, and Random Access SNR settings. When the Rotational transform and its transpose are used, average BD-Rate gains of 1.1 % , 0.6%, 0.6 %, 0.3 % and 0.6 % are obtained (Table 4). For tests 5 and 6, we perform tests on 10 frames for Intra settings only to find out the gains when the transforms are applied at sizes 16 and 32 only. The results for full-frame tests exhibit a similar behavior, and are in the attached excel sheets.

## Training of Matrices

No training was performed on any video sequence.

## Complexity

**Bit Shift at Intermediate Levels:** In our implementation, we shift the output of secondary transform by by 7 bits after vertical or horizontal transform. The bit-precision for all the transforms is 16 bits, similar to in SHM 1.0.

**Table 3. Worst-case and Average-Case operations count summary for the secondary transforms**



From the worst-case, and average-case operation counts, it can be easily seen that the additional complexity for applying the secondary transforms or ROT on 16x16 and 32x32 block is almost marginal.

## Techniques for reducing latency for secondary transform in hardware

We refer the reader to [10] for various strategies to reduce the latency in secondary transforms in hardware.

# Conclusions

In this contribution, we have presented various secondary transforms for Intra\_BL residue coding instead of the conventional DCT at block sizes 8x8 and higher. The proposed secondary transform scheme requires the storage of only 1 or 2 additional transform cores depending on whether one or two secondary transforms are used. Simulation results show significant gains in compression performance as compared to SHM 1.0 anchors. We recommend adopting this proposal into SHM.

# References

1. L. Guo, M. Karczewicz, J. Sole, and J. Chen, “Transform Selection for Inter-Layer Texture Prediction in Scalable Video Coding”, JCTVC-K0321, Shanghai, China, Oct 2012.
2. L. Guo, M. Karczewicz, and J. Chen, “Test of Transform Selection for Inter-Layer Texture Prediction on SMuC 0.1.1”, JCTVC-L0330, Geneva, Switzerland, Jan 2013.
3. SHVC Test Model 1 (SHM 1), JCTVC-L1007, Geneva, Switzerland, Jan 2013.
4. V. Britanak, P. Yip and K. Rao, Discrete Cosine and Sine Transforms: General Properties, Fast Algorithms and Integer Approximations, Elsevier, 2007.
5. A. Saxena and F. Fernandes, “On secondary transforms for prediction residual,” Orlando, USA, ICIP 2012.
6. A. Saxena and F. Fernandes, “Mode-dependent DCT/DST without 4\*4 full matrix multiplication for intra prediction,” JCTVC-E125, Geneva, Switzerland, March 2011.
7. C. Loeffler, A. Lightenberg, and G. Moschytz, “Practical fast 1-D DCT algorithms with 11 multiplications,” IEEE ICASSP, vol. 2, pp. 988–991, Feb. 1989
8. E. Alshina, A. Alshin and F. Fernandes, “Rotational transform for image and video compression”, IEEE ICIP, Brussels, Belgium, Sept 2011.
9. Z. Ma, F. Fernandes, E. Alshina, A. Alshin, “Experimental Results for the ROT”, JCTVC-G304, Geneva, Switzerland, Nov 2011.
10. A. Saxena, Y. Shibahara, F. Fernandes and T. Nishi, “On secondary transforms for intra prediction residual,” JCTVC-H0125, San Jose, USA, Feb 2012.

# Patent rights declaration(s)

**Samsung Electronics Co., Ltd., may have IPR relating to the technology described in this contribution and, conditioned on reciprocity, is prepared to grant licenses under reasonable and non-discriminatory terms as necessary for implementation of the resulting ITU-T Recommendation | ISO/IEC International Standard (per box 2 of the ITU-T/ITU-R/ISO/IEC patent statement and licensing declaration form).**