

SHARP

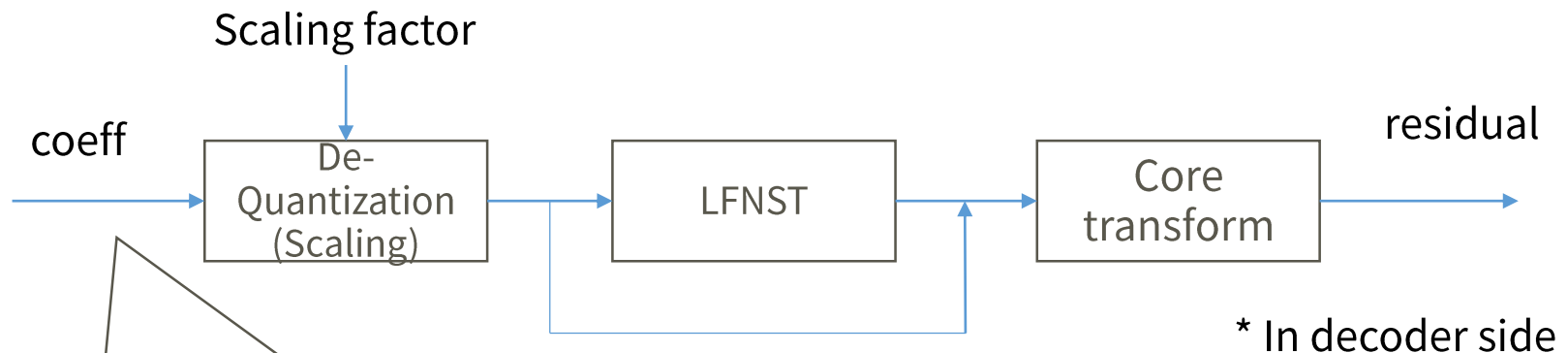
Be Original.

JVET-P0292

AHG15: Scaling process for LFNST case

Tomonori Hashimoto
Eiichi Sasaki
Tomohiro Ikai

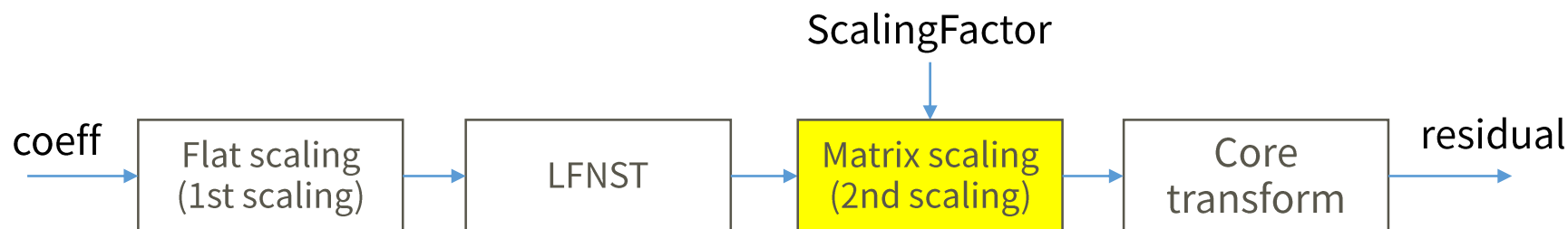
- Scaling matrices are applied even in the case of LFNST mode in WD6.
 - It would be wrong since LFNST's coefficient does not correspond to conventional transform coefficients in spatial frequency meaning.



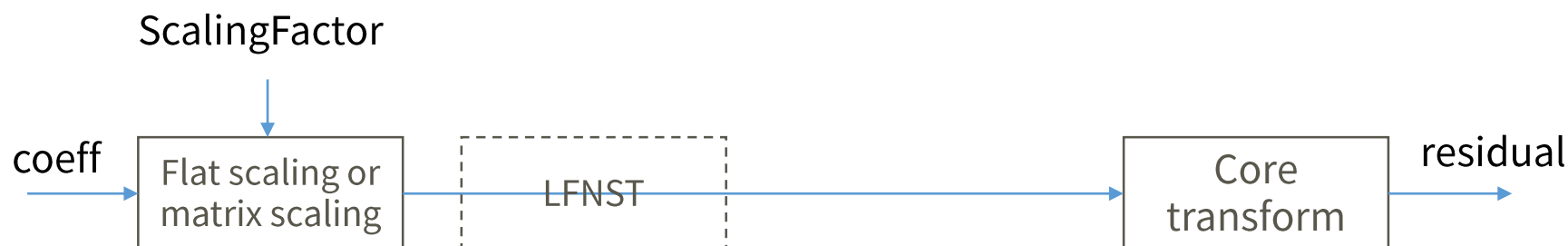
Not normal spatial coefficient domain
in LFNST case

Proposed method

- If LFNST on and scaling is enabled, newly introduced 2nd scaling is applied after LFNST.



- If LFNST off or no scaling List, no changes



- If LFNST case, flat matrix is applied first

8.7.3 Scaling process for transform coefficients

- If one or more of the following conditions are true, $m[x][y]$ is set equal to 16:
 - `sps_scaling_list_enabled_flag` is equal to 0.
 - `transform_skip_flag[xTbY][yTbY]` is equal to 1.
 - `lfnst_idx[xTbY][yTbY]` is not equal to 0 and both `nTbW` and `nTbH` are greater than or equal to 4

- In LFNST and scaling is enabled, second scaling is applied

•8.7.5.1 Transformation process for scaled transform coefficients

- The values of the list $u[x]$ with $x = 0..nonZeroSize - 1$ are derived as follows:

$$xC = \text{DiagScanOrder}[2][x][0] \quad (8-970)$$

$$yC = \text{DiagScanOrder}[2][x][1] \quad (8-971)$$

$$u[x] = d[xC][yC] \quad (8-972)$$

- The one-dimensional low frequency non-separable transformation process as specified in clause 8.7.4.2 is invoked with the input length of the scaled transform coefficients $nonZeroSize$, the transform output length $nTrS$ set equal to $nLfnstOutSize$, the list of scaled non-zero transform coefficients $u[x]$ with $x = 0..nonZeroSize - 1$, the intra prediction mode for LFNST set selection $predModeIntra$, and the LFNST index for transform selection in the selected LFNST set $lfnst_idx[xTbY][yTbY]$ as inputs, and the list $v[x]$ with $x = 0..nLfnstOutSize - 1$ as output.

- The array $d[x][y]$ with $x = 0..nLfnstSize - 1$, $y = 0..nLfnstSize - 1$ is derived as follows:

- If $predModeIntra$ is less than or equal to 34, the following applies:

$$d[x][y] = (y < 4) ? v[x + (y << \log2LfnstSize)] : d[x][y] \quad (8-973)$$

$$((x < 4) ? v[32 + x + ((y - 4) << 2)] : d[x][y])$$

- Otherwise, the following applies:

$$d[x][y] = (x < 4) ? v[y + (x << \log2LfnstSize)] : d[x][y] \quad (8-974)$$

$$((y < 4) ? v[32 + y + ((x - 4) << 2)] : d[x][y])$$

- When `scaling_list_enabled_flag` is equal to 1, The second scaling process for transform coefficients as specified in clause 8.7.4 is invoked with the transform block location $(xTbY, yTbY)$, the transform block width $nTbW$ and the transform block height $nTbH$, the colour component variable $cIdx$ and the bit depth of the current colour component $bitDepth$ as inputs, and the output is an $(nTbW) \times (nTbH)$ array of modified scaled transform coefficients d .

- In LFNST and scaling is enabled, second scaling is applied

Second scaling process for transform coefficients

...

The variables `bdShift` and `bdOffset` are derived as follows:

$$\text{bdShift} = 4 \quad (8\text{-xxxx})$$

$$\text{bdOffset} = (1 \ll \text{bdShift}) \gg 1 \quad (8\text{-xxxx})$$

For the derivation of the scaled transform coefficients $d[x][y]$ with $x = 0..nTbW - 1$, $y = 0..nTbH - 1$, the following applies:

– The scaling factor $m[x][y]$ is derived as follows:

$$m[x][y] = \text{ScalingFactor}[\text{sizeIdW}][\text{sizeIdH}][\text{matrixId}][x][y] \quad (8\text{-XXX})$$

Where `sizeIdW` is set equal to $\text{Log2}(nTbW)$, `sizeIdH` is set equal to $\text{Log2}(nTbH)$ and `matrixId` is specified in Table 7-4 for `sizeId` set equal to $\text{Max}(\text{sizeIdW}, \text{sizeIdH})$, `CuPredMode[xTbY][yTbY]` and `cIdx`, respectively.

– The value $\text{dnc}[x][y]$ is derived as follows:

$$\text{dnc}[x][y] = (d[x][y] * m[x][y] + \text{bdOffset}) \gg \text{bdShift} \quad (8\text{-xxxx})$$

– The scaled transform coefficient $d[x][y]$ is derived as follows:

$$d[x][y] = \text{Clip3}(\text{CoeffMin}, \text{CoeffMax}, \text{dnc}[x][y])$$

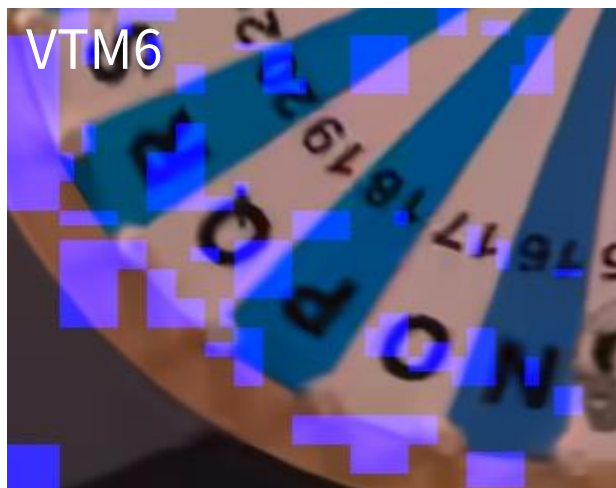
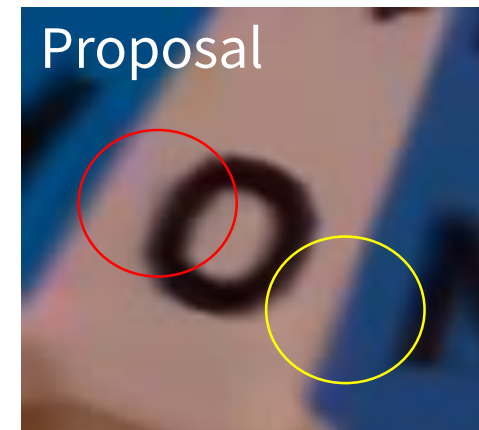
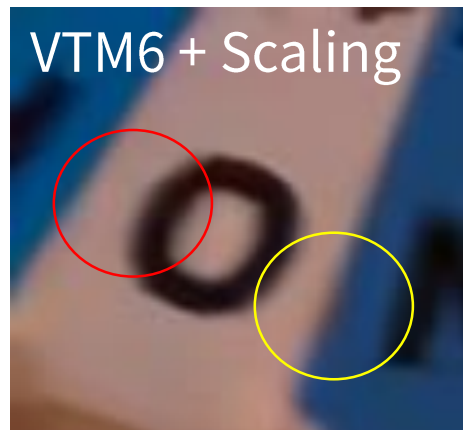
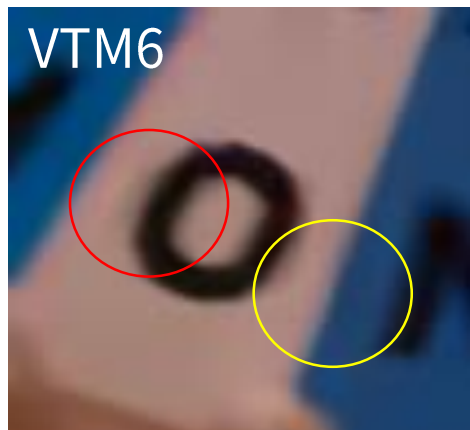
- Support transform coefficient control capability for LFNST transform as well
 - Scaling matrices are always applied in the conventional transform domain even if LFNST is applied.
 - Some improvements are observed.
- No additional signaling

- Proposed method is implemented on top of version 7977eb8e VTM.
- Scaling matrix attached in this contribution is used.
- Proposal + Qmat vs. VTM + Qmat

	All Intra Main10					Random access Main10				
	Over VTM-6.0 (7977eb8e) scaling matrix enabled					Over VTM-6.0 (7977eb8e) scaling matrix enabled				
	Y	U	V	EncT	DecT	Y	U	V	EncT	DecT
Class A1	-0.01%	-0.01%	0.12%	102%	101%	-0.03%	0.06%	0.00%	100%	100%
Class A2	-0.02%	0.14%	0.23%	102%	101%	-0.03%	-0.05%	0.05%	100%	100%
Class B	0.01%	0.08%	0.16%	101%	101%	0.00%	0.24%	0.15%	100%	102%
Class C	0.01%	0.15%	0.22%	101%	103%	0.01%	0.26%	0.25%	100%	105%
Class E	0.01%	0.23%	-0.06%	100%	102%					
Overall	0.00%	0.11%	0.14%	101%	101%	-0.01%	0.15%	0.13%	100%	102%
Class D	0.00%	0.17%	-0.04%	100%	107%	0.00%	0.21%	-0.02%	100%	117%
Class F	0.03%	0.08%	0.23%	100%	102%	0.01%	0.10%	0.07%	100%	104%

No obvious loss was found.

Visual quality check



VTM6



VTM6 + Scaling



Proposal



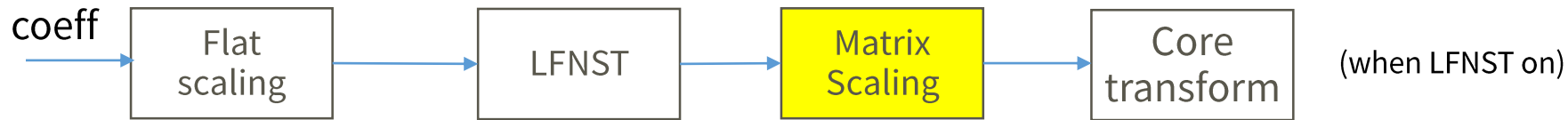
- We propose a scaling process for LFNST case.
- The proposed method provides transform coefficient control capability in all conventional and LFNST transform cases.
- It is recommended to adopt this method in the next VVC WD and VTM software.

Thank Bytedance for crosscheck (JVET-P0657).

SHARP

Be Original.

- In LFNST on case, this and previous one propose the same process



- In LFNST off case, previous proposal changed the process while this proposal doesn't change.

P00282 (this proposal), no change to VVC6



00383 (previous proposal)

