One of the VVC tools heavily contributed to compression performance improvement is a flexible Versatile Video Coding also known by its abbreviation DC mode. One of these collinear modes is a WAIP integer tree (3) that is identical to chroma DCT (6). It relates to the new coding structure (3):

$$\left( \begin{array}{l} H \\text{DC} \\ \hline \end{array} \right)$$

(3)

Intra Prediction in the Emerging VVC Video Coding Standard

| Block size (the larger tap DCT predictors are filtered in HEVC, propagation depth is increased and becomes dependent on filters applied to reference samples). In this paper, we focus on two sequentially invoking reference sample smoothing and interpolation filtering that could result in a latency issue.

<table>
<thead>
<tr>
<th>Overall: CTC</th>
<th>28.26%</th>
<th>25.90%</th>
<th>28.26%</th>
<th>25.90%</th>
<th>28.26%</th>
<th>25.90%</th>
</tr>
</thead>
</table>

The individual tools, including WAIP, 4-up interpolation filtering and PDPC are tested and the results are shown in Table 2.

**Non-functional intra prediction DC mode**

Design of an intra prediction mode that is considered to be the least complex has to be kept hardware-friendly for both square and rectangular blocks. In particular, division operation trivially implemented by shift-operation only for square shape can be an issue in the case of rectangular shapes. To skip this computationally complex operation, a couple of methods were proposed. Fig. 7 illustrates the technique adopted into VVC [19]. This solution, on the one hand, provides the design space exploration with two blocks: one block is a rectangular block, and the other is a square block. The rectangular block is calculated by a DC value only along a longer side.

**PDPC**

Yet, as observed for intra prediction in AVC/HEVC, only a few directional and non-directional modes interplay discontinuities along block boundaries [14]. To remedy this problem, post-prediction filtering was applied to DC, horizontal and vertical modes [17]. In VVC, the number of the post-prediction filtering was extended to a total of six, whereas in HEVC, it was applied to a wider subset of intra prediction modes (e.g., modes -16, 14, and 58.8417[17]). Thus, the post-prediction filter known in literature as Block and Line Smoothing (BLS) [17] is applied to the current block with 16 discontinuities near a left and above boundaries within the predicted blocks as shown in Fig. 6. The following formulas is used to update the predicted sample $p_{i,j}$:

$$p_{i,j} = \frac{1}{4}(p_{i-1,j} + p_{i+1,j} + p_{i,j-1} + p_{i,j+1}) - \alpha DCT(p_{i,j})$$

$$\alpha = \begin{cases} 0.001 & \text{if } \left| \frac{w(h)}{h(w)} \right| < 2 \sqrt{2} \\ 0.005 & \text{otherwise} \end{cases}$$

where $\alpha$ denotes reference samples, $p_{i,j}$ denotes predicted samples, $DCT(p_{i,j})$ is a clipping function that prevents exceeding the sample range by an output value and is actually required only for these two modes in other blocks. The updated sample is a weighted sum of predicted samples and a reference sample.

**Reference sample filtering**

Similar to HEVC, in VVC prediction in VVC has 2 filtering methods applied to reference samples. Therefore, one reference sample smoothing and interpolation filtering (see Fig. 3) [14]. Reference sample smoothing applied only to integer-slope modes in luma blocks modifies reference samples by using the finite impulse response filter $1, 2, 3$ that does not require multiplication operations and allows keeping the design of integer-slope modes simple as compared to fractional-slope ones, which use an interpolation filter. It is worth noting that such a design enables multiplication-free implementations of “lazy” encoders if among directional modes, only horizontal, vertical and integer-slope modes are checked. More detailed explanations on handling reference samples could be found in [15].

**Table 1: Objective performance of VTM6.0 over HM16-20.**

<table>
<thead>
<tr>
<th>Block size</th>
<th>101%/58%</th>
<th>102%/59%</th>
<th>101%/58%</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Overall: CTC</th>
<th>28.26%</th>
<th>25.90%</th>
<th>28.26%</th>
<th>25.90%</th>
<th>28.26%</th>
<th>25.90%</th>
</tr>
</thead>
</table>

**References**