The background is a solid red color. Overlaid on this are various faint, semi-transparent technical graphics. On the left, there's a small icon of a code editor with a '</>' symbol. Next to it is a circular radar-like graphic. In the center, there are some faint, illegible text fragments that appear to be code or technical specifications. On the right side, there's a complex network diagram with many nodes (small circles) connected by lines, with a bright light source emanating from one of the nodes.

AHG8: RPR Scaling Window Issues

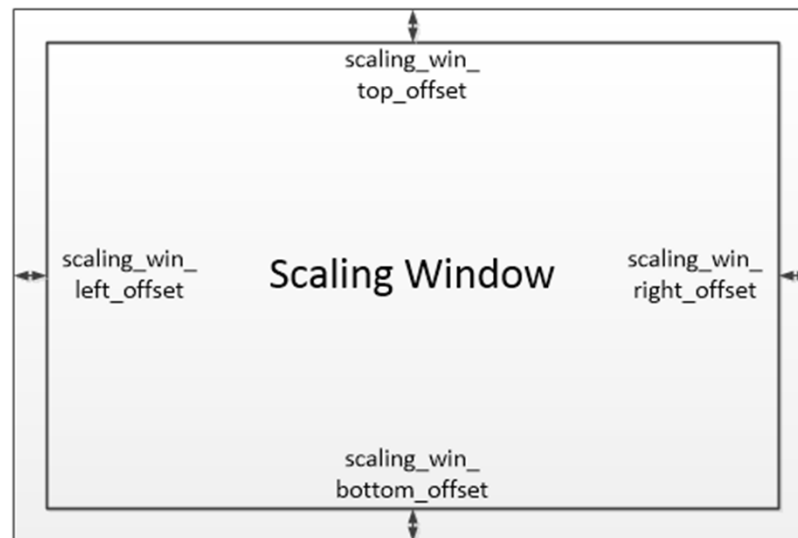
JVET-Q0487

W. Wan, T. Hellman, B. Heng, P. Chen
Brussels, Belgium, January 2020 meeting



Introduction

- A per-picture scaling window used to calculate RPR scaling factors between current picture and its reference pictures
- Contribution discuss four issues with current definition

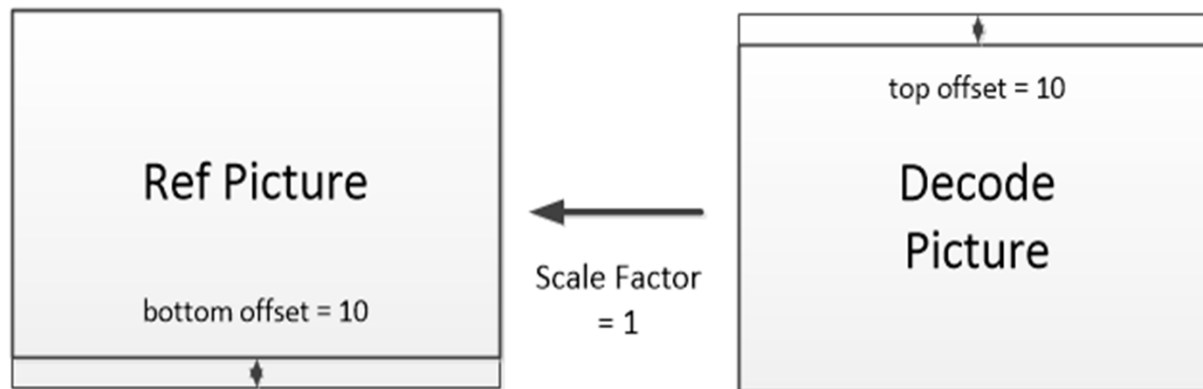


Several cases resulting in 1:1 scaling factors

1. Obvious case (same coded picture width and height, same scaling window offsets)
2. Same coded picture width and height, same scaling window width and height, but different scaling window offsets
3. Completely different coded picture width and height, but same scaling window width/height.

Case #2

- Same coded picture width and height, same scaling window width and height, but different scaling window offsets



Issue #1: Temporal Motion Vector Prediction (TMVP) Constraint

- TMVP needs to be disabled when the co-located reference picture differs in coded size and scale factor from the current picture, otherwise undefined temporal motion vectors could be referenced.
- Semantics of `collocated_ref_idx` in slice header of current text:

It is a requirement of bitstream conformance that the resolutions of the reference picture referred to by `collocated_ref_idx` and the current picture shall be the same and `RefPicIsScaled[collocated_from_10_flag? 0 : 1][collocated_ref_idx]` shall be equal to 0.

- The term “resolutions” is ambiguous and not defined anywhere
- Important to clarify that the true (i.e. coded) width and height of the current and co-located pictures must be the same

Issue #1: TMVP Constraint (continued)

- Proposed change

It is a requirement of bitstream conformance that the ~~resolutions of the reference picture referred to by `collocated_ref_idx` and the current picture~~ values of `pic_width_in_luma_samples` and `pic_height_in_luma_samples` in the current picture and in the reference picture referred to by `collocated_ref_idx` shall be the same and `RefPicIsScaled[collocated_from_l0_flag? 0 : 1][collocated_ref_idx]` shall be equal to 0.

Issue #2: Code/Spec Mismatch when Scale Factor is 1

- Specification: In Section 8.5.6.3.1, RPR offsets and scaling is applied to all motion vectors, regardless of whether the scaling factor is 1:1 or not. The 1:1 scale factor (i.e. 2^{14}) is treated the same as any other scale factor. Therefore, during this process any left_offset or top_offset will be applied accordingly, even if the scaling factor is 1:1.
- VTM 7.x Code: In the xPredInterBlkRPR() function, all RPR-related logic is only applied if the scaling factor is not 1:1. Specifically:

```
const bool scaled = scalingRatio != SCALE_1X;

if( scaled )
{
    "RPR prediction"
}
else
{
    "standard prediction"
}
```

The standard inter-prediction process is the same as it was before RPR, and makes no adjustments for RPR left/top scaling window offsets. Therefore, when the scalingRatio is 1:1 above, any left/top offsets will be ignored and motion compensation will fail to match the specification text.

Issue #2: Code/Spec Mismatch (continued)

- Logically seems specification is correct since application of the offset shouldn't depend on whether the scale factor is 1:1 or not.
- Can provide SW to modify the source code so it aligns with the specification

Issue #3: Scaling Window Offsets in DMVR

- v1 of contribution thought spec for Case #2 was OK
- Upon further review, discovered DMVR not properly specified
 - None of the motion vector refinement process (Section 8.5.3) uses the scaling window left or top offsets
 - None of the bounding box derivation (Section 8.5.6.3.1) accounts for these offsets either.
 - Unintended use case / not well studied, maybe there are other issues?
- Assert no practical reason and potential complexity to support switching padding from top to bottom as in Case #2
- Propose disabling DMVR for case #2

Issue #4: Half-Sample Chroma Offsets

- Scaling window offsets specified in units of luma samples
 - Thus, half-sample chroma offsets possible (odd luma values)
 - When attempting to fix SW, unusual results seen in these cases
- Section 8.5.6.3.1: the precision of the current picture's left scaling offset (**red**) is treated with less precision than the reference picture's left scaling offset (**blue**)????

“The variable **fRefLeftOffset** is set equal to (**scaling_win_left_offset** << 10), where **scaling_win_left_offset** is the **scaling_win_left_offset** for the reference picture.

$$\text{refxSb}_C = (((\text{xSb} - \text{scaling_win_left_offset}) / \text{SubWidthC} \ll 5) + \text{refMvLX}[0]) * \text{scalingRatio}[0] + \text{addX}$$
$$\text{refx}_C = ((\text{Sign}(\text{refxSb}_C) * ((\text{Abs}(\text{refxSb}_C) + 256) \gg 9) + \text{x}_C * ((\text{scalingRatio}[0] + 8) \gg 4)) + \text{fRefLeftOffset} / \text{SubWidthC} + 16) \gg 5$$

Issue #4: Half-Sample Chroma Offsets (continued)

- Since all inter-prediction tools use the process described in the text above, all tools would need to support this unintentional half-sample chroma shift, even if those tools do not support RPR scaling.
- Assert half-sample chroma left/top offsets is unnecessary
- Propose specifying `scaling_win_left_offset` and `scaling_win_right_offset` in units of `SubWidthC` luma samples, and specifying `scaling_win_top_offset` and `scaling_win_bottom_offset` in units of `SubHeightC` luma samples to avoid half-sample chroma to be used



BROADCOM[®]

connecting everything[®]