

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 binary digits (0s and 1s) and what looks like a snippet of code. On the right side, there's a complex network diagram with many nodes (small circles) connected by lines, with a bright light source in the center of the network.

AHG8: RPR Scaling Window Issues

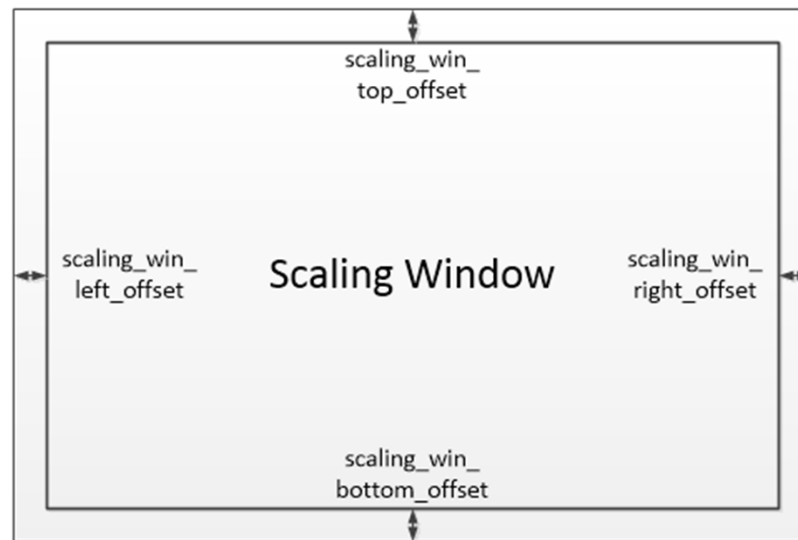
JVET-Q0487

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 two issues with current definition

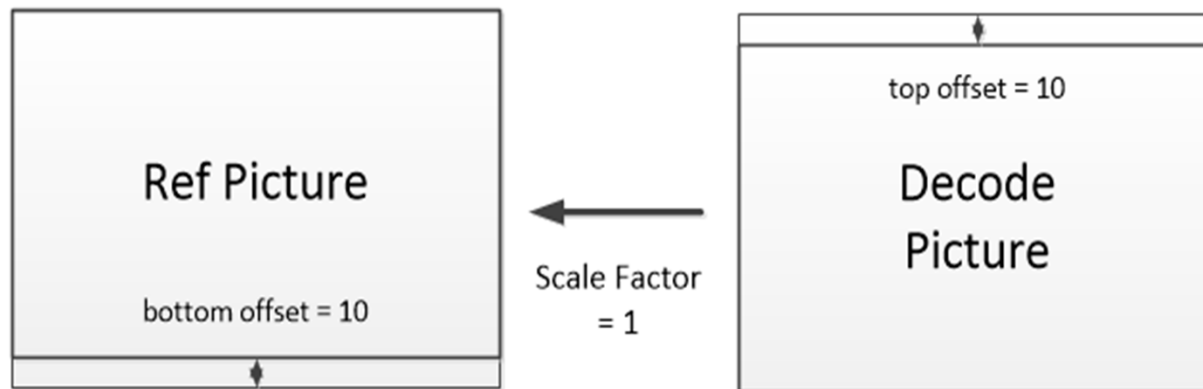


Several cases resulting in 1:1 scaling factors

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_10_flag? 0 : 1 ][ collocated_ref_idx ]` shall be equal to 0.

- Prevents TMVP for case #3 earlier but allows it for case #2 (same coded picture width and height, same scaling window width and height, but different scaling window offsets)
- Even though case #2 may intuitively be suboptimal TMVP coding, there is nothing broken or undefined in this case

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



BROADCOM[®]

connecting everything[®]