

The background is a solid red color. Overlaid on this are various semi-transparent white and light red graphics. On the left, there are faint icons of a code editor with a '</>' symbol, a waveform graph, and a circular target-like graphic. In the center and right, there are complex network diagrams consisting of numerous small dots connected by thin lines, with some dots being larger and more prominent. The overall aesthetic is high-tech and digital.

AHG8: Constraints on Scaling Window Offset Parameters

JVET-Q0486

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



Introduction

- Scaling window adopted for deriving RPR scaling ratios at Geneva October 2019 meeting
- Horizontal and vertical scaling factors computed by comparing scaling windows of current and reference pictures

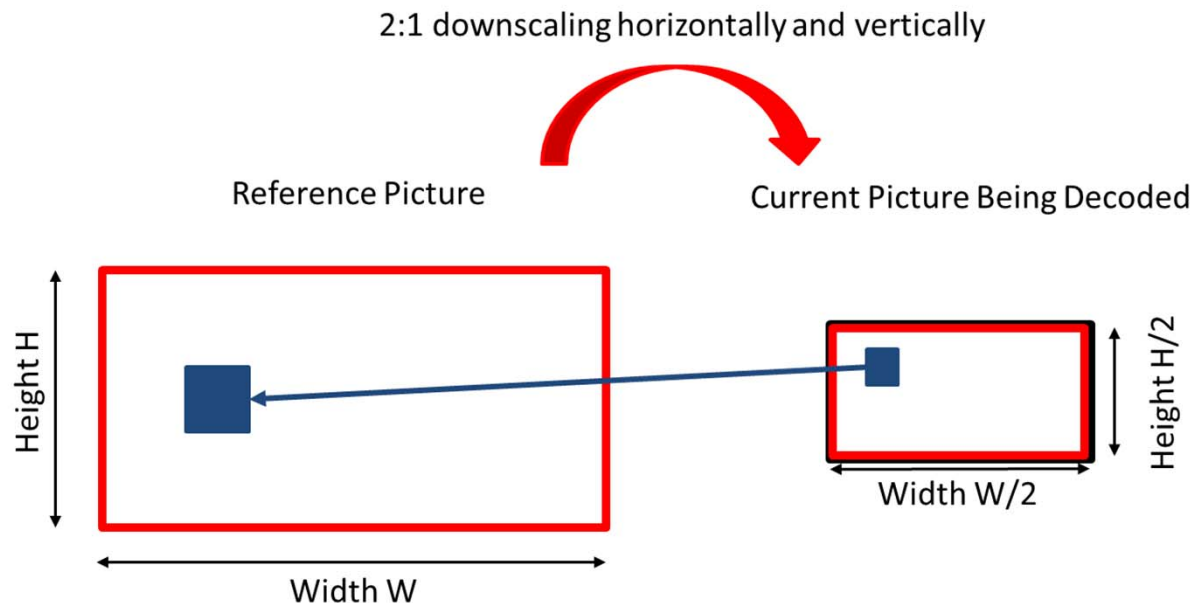
pic_parameter_set_rbsp() {	Descriptor
.....	
scaling_window_flag	u(1)
if(scaling_window_flag) {	
scaling_win_left_offset	ue(v)
scaling_win_right_offset	ue(v)
scaling_win_top_offset	ue(v)
scaling_win_bottom_offset	ue(v)
}	
.....	u(1)

Scaling Window Offsets

- Main intent was to enable RPR for picture widths or heights that were not exactly a multiple of $\text{Max}(8, \text{MinCbSizeY})$ luma samples
- Scaling window determines the applicable scaling ratio
- Important to note processing is applied to the whole picture (which includes pixels outside the scaling window)
- Lack of constraints on offset values inadvertently allow corner case configurations with implementational complexity
 - Only constraints are $(\text{left} + \text{right}) < \text{width}$ and $(\text{top} + \text{bottom}) < \text{height}$

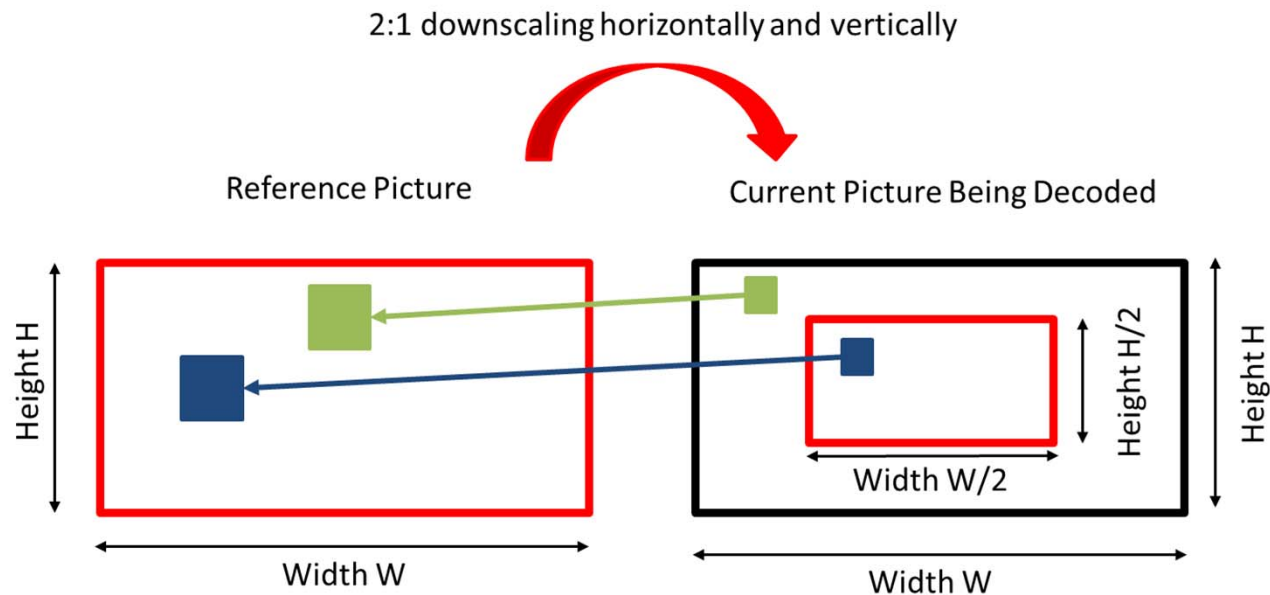
Example: Scaling window is entire picture

- Processing requirements are similar to a HxW picture without RPR (1/4 the pixels with 4X the memory bandwidth)



Example: Scaling window is subset of the entire picture

- Processing requirements are **4X(!!!)** of a HxW picture without RPR (same number of pixels with 4X the memory bandwidth)



Discussion

- Original RPR design was to add RPR without significantly increasing the complexity burden compared to a non-RPR decoder
- Functionality identified was support for picture sizes that were not exactly a multiple of $\text{Max}(8, \text{MinCbSizeY})$ luma samples
- Restricting offsets to be no greater than $\text{Max}(8, \text{MinCbSizeY})$ enables desired functionality without creating problematic corner case configurations to support

Proposed changes in PPS semantics

`scaling_win_left_offset`, `scaling_win_right_offset`, `scaling_win_top_offset`, and `scaling_win_bottom_offset` specify the offsets, in units of luma samples, that are applied to the picture size for scaling ratio calculation. The values of `scaling_win_left_offset`, `scaling_win_right_offset`, `scaling_win_top_offset`, and `scaling_win_bottom_offset` shall be in the range of 0 to $\text{Max}(8, \text{MinCbSizeY})$, inclusive. When `scaling_window_flag` is equal to 0, the values of `scaling_win_left_offset`, `scaling_win_right_offset`, `scaling_win_top_offset`, and `scaling_win_bottom_offset` are inferred to be equal to 0.



BROADCOM[®]

connecting everything[®]