

CLAIMS

1. A method of video processing, comprising:
 - performing a conversion between a video comprising one or more video pictures comprising one or more subpictures and a bitstream of the video,
 - wherein the conversion conforms to a rule that specifies that one or more parameters for a scaling window applicable to a subpicture are determined from one or more syntax elements during a subpicture sub-bitstream extraction process.
2. The method of claim 1, wherein the one or more syntax elements are included in an original picture parameter set and/or an original sequence parameter set.
3. The method of claim 1 or 2, wherein the original picture parameter set and/or the original sequence parameter set is changed with a calculation and a rewriting of the one or more parameters.
4. The method of any of claims 1 to 3, wherein the one or more parameters include at least one of a left offset, a right offset, a top offset or a bottom offset of the scaling window applicable to the subpicture.
5. The method of any of claims 1 to 4, wherein the left offset of the scaling window (subpicScalWinLeftOffset) is derived as follows:

$$\text{subpicScalWinLeftOffset} = \text{pps_scaling_win_left_offset} - \text{sps_subpic_ctu_top_left_x}[\text{spIdx}] * \text{CtbSizeY} / \text{SubWidthC}, \text{ and}$$
 wherein pps_scaling_win_left_offset indicates left offsets applied to a picture size for scaling ratio calculation, sps_subpic_ctu_top_left_x[spIdx] indicates a x-coordinate of a coding tree unit located at a top-left corner of the subpicture, CtbSizeY is a width or a height of a luma coding tree block or coding tree unit, and the SubWidthC indicates a width of a video block and is obtained from a table according to a chroma format of a picture including the video block.

6. The method of any of claims 1 to 4, wherein the right offset of the scaling window, subpicScalWinRightOffset, is derived as follows:

subpicScalWinRightOffset = (rightSubpicBd >= sps_pic_width_max_in_luma_samples) ?
 pps_scaling_win_right_offset : pps_scaling_win_right_offset –
 (sps_pic_width_max_in_luma_samples – rightSubpicBd) / SubWidthC, and

wherein sps_pic_width_max_in_luma_samples specifies a maximum width, in units of luma samples, of each decoded picture referring to a sequence parameter set,

pps_scaling_win_right_offset indicates right offsets applied to a picture size for scaling ratio calculation, SubWidthC indicates a width of a video block and is obtained from a table according to a chroma format of a picture including the video block, and

rightSubpicBd = (sps_subpic_ctu_top_left_x[spIdx] + sps_subpic_width_minus1[spIdx] + 1) * CtbSizeY, and

wherein sps_subpic_ctu_top_left_x[spIdx] indicates a x-coordinate of a coding tree unit located at a top-left corner of the subpicture, sps_subpic_width_minus1[spIdx] indicates a width of the subpicture, and CtbSizeY is a width or a height of a luma coding tree block or coding tree unit.

7. The method of any of claims 1 to 4, wherein the top offset of the scaling window, subpicScalWinTopOffset, is derived as follows:

subpicScalWinTopOffset = pps_scaling_win_top_offset –
 sps_subpic_ctu_top_left_y[spIdx] * CtbSizeY / SubHeightC,

wherein pps_scaling_win_top_offset indicates top offsets applied to a picture size for scaling ratio calculation, sps_subpic_ctu_top_left_y[spIdx] indicates a y-coordinate of a coding tree unit located at a top-left corner of the subpicture, CtbSizeY is a width or a height of a luma coding tree block or coding tree unit, and SubHeightC indicates a height of a video block and is obtained from a table according to a chroma format of a picture including the video block.

8. The method of any of claims 1 to 4, wherein the bottom offset of the scaling window, subpicScalWinBotOffset, is derived as follows:

$$\text{subpicScalWinBotOffset} = (\text{botSubpicBd} \geq \text{sps_pic_height_max_in_luma_samples}) ?$$

$$\text{pps_scaling_win_bottom_offset} : \text{pps_scaling_win_bottom_offset} -$$

$$(\text{sps_pic_height_max_in_luma_samples} - \text{botSubpicBd}) / \text{SubHeightC}, \text{ and}$$
 wherein $\text{sps_pic_height_max_in_luma_samples}$ indicates a maximum height, in units of luma samples, of each decoded picture referring to a sequence parameter set, $\text{pps_scaling_win_bottom_offset}$ indicates bottom offsets applied to a picture size for scaling ratio calculation, SubHeightC indicates a height of a video block and is obtained from a table according to a chroma format of a picture including the video block, and
 wherein $\text{botSubpicBd} = (\text{sps_subpic_ctu_top_left_y}[\text{spIdx}] + \text{sps_subpic_height_minus1}[\text{spIdx}] + 1) * \text{CtbSizeY}$, and
 wherein $\text{sps_subpic_ctu_top_left_y}[\text{spIdx}]$ indicates a y-coordinate of a coding tree unit located at a top-left corner of the subpicture and CtbSizeY is a width or a height of a luma coding tree block or coding tree unit.

9. The method of any of claims 5-8, wherein $\text{sps_subpic_ctu_top_left_x}[\text{spIdx}]$, $\text{sps_subpic_width_minus1}[\text{spIdx}]$, $\text{sps_subpic_ctu_top_left_y}[\text{spIdx}]$, $\text{sps_subpic_height_minus1}[\text{spIdx}]$, $\text{sps_pic_width_max_in_luma_samples}$, and $\text{sps_pic_height_max_in_luma_samples}$ are from the original sequence parameter set, and $\text{pps_scaling_win_left_offset}$, $\text{pps_scaling_win_right_offset}$, $\text{pps_scaling_win_top_offset}$, and $\text{pps_scaling_win_bottom_offset}$ are from the original picture parameter set.

10. The method of any of claims 1 to 3, wherein the rule specifies to rewrite values of $\text{pps_scaling_win_left_offset}$, $\text{pps_scaling_win_right_offset}$, $\text{pps_scaling_win_top_offset}$, and $\text{pps_scaling_win_bottom_offset}$ in all referenced picture parameter set (PPS) network abstraction layer (NAL) units to be equal to $\text{subpicScalWinLeftOffset}$, $\text{subpicScalWinRightOffset}$, $\text{subpicScalWinTopOffset}$, and $\text{subpicScalWinBotOffset}$, respectively, and
 wherein $\text{pps_scaling_win_left_offset}$, $\text{pps_scaling_win_right_offset}$, $\text{pps_scaling_win_top_offset}$, and $\text{pps_scaling_win_bottom_offset}$ respectively indicate left offsets, right offsets, top offsets, bottom offsets that are applied to a picture size for scaling ratio calculation, and

wherein subpicScalWinLeftOffset, subpicScalWinRightOffset, subpicScalWinTopOffset, and subpicScalWinBotOffset respectively indicate a left offset, a right offset, a top offset and a bottom offset of the scaling window applicable to the subpicture.

11. The method of any of claims 1 to 4, wherein the bottom offset of the scaling window, subpicScalWinBotOffset, is derived as follows:

$$\text{subpicScalWinBotOffset} = (\text{botSubpicBd} \geq \text{pps_pic_height_in_luma_samples}) ?$$

$$\text{pps_scaling_win_bottom_offset} : \text{pps_scaling_win_bottom_offset} -$$

$$(\text{pps_pic_height_in_luma_samples} - \text{botSubpicBd}) / \text{SubHeightC}, \text{ and}$$

wherein pps_pic_height_max_in_luma_samples indicates a maximum height, in units of luma samples, of each decoded picture referring to a picture parameter set,

pps_scaling_win_bottom_offset indicates bottom offsets applied to a picture size for scaling ratio calculation, SubHeightC indicates a height of a video block and is obtained from a table

according to a chroma format of a picture including the video block, and

wherein $\text{botSubpicBd} = (\text{sps_subpic_ctu_top_left_y}[\text{spIdx}] +$

$\text{sps_subpic_height_minus1}[\text{spIdx}] + 1) * \text{CtbSizeY},$

wherein sps_subpic_ctu_top_left_y[spIdx] indicates a y-coordinate of a coding tree unit located at a top-left corner of the subpicture and CtbSizeY is a width or a height of a luma coding tree block or coding tree unit.

12. The method of claim 11, wherein pps_pic_height_in_luma_samples is from the original picture parameter set.

13. The method of claim 1, wherein the rule further specifies to rewrite a reference sample enabled flag in a sequence parameter set that indicates an applicability of a reference picture resampling.

14. The method of claim 1, wherein the rule further specifies to rewrite a

a resolution change allowed flag in a sequence parameter set that indicates a possible picture spatial resolution change within a coded layer video sequence (CLVS) referring to the sequence parameter set.

15. The method of any of claims 1 to 14, wherein the conversion includes encoding the video into the bitstream.

16. The method of any of claims 1 to 14, wherein the conversion includes decoding the video from the bitstream.

17. The method of any of claims 1 to 14, wherein the conversion includes generating the bitstream from the video, and the method further comprises: storing the bitstream in a non-transitory computer-readable recording medium.

18. A video processing apparatus comprising a processor configured to implement a method recited in any one or more of claims 1 to 17.

19. A method of storing a bitstream of a video, comprising, a method recited in any one of claims 1 to 17, and further including storing the bitstream to a non-transitory computer-readable recording medium.

20. A computer readable medium storing program code that, when executed, causes a processor to implement a method recited in any one or more of claims 1 to 17.

21. A computer readable medium that stores a bitstream generated according to any of the above described methods.

22. A video processing apparatus for storing a bitstream representation, wherein the video processing apparatus is configured to implement a method recited in any one or more of claims 1 to 17.