



(51) International Patent Classification:

H04N 19/17 (2014.01) H04N 19/44 (2014.01)
H04N 19/119 (2014.01) H04N 19/70 (2014.01)
H04N 19/167 (2014.01)

(21) International Application Number:

PCT/SE2024/050961

(22) International Filing Date:

08 November 2024 (08.11.2024)

(25) Filing Language:

English

(26) Publication Language:

English

(30) Priority Data:

63/548,061 10 November 2023 (10.11.2023) US

(71) Applicant: **TELEFONAKTIEBOLAGET LM ERICSSON (PUBL)** [SE/SE]; SE-164 83 Stockholm (SE).

(72) Inventors: **DAMGHANIAN, Mitra**; Isafjordsgatan 14E, 164 80 Stockholm (SE). **PETTERSSON, Martin**; Isafjordsgatan 14E, 164 80 Stockholm (SE). **SJÖBERG, Rickard**; Isafjordsgatan 14E, 164 80 Stockholm (SE).

(74) Agent: **GIRDZIJAUSKAS, Ivana**; Ericsson AB, PCA1 PU Kista2&India, Isafjordsgatan 14, 164 80 Stockholm (SE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CV, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IQ, IR, IS, IT, JM, JO, JP, KE, KG,

KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, MG, MK, MN, MU, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SA, SC, SD, SE, SG, SK, SL, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, WS, ZA, ZM, ZW.

(84) Designated States (unless otherwise indicated, for every kind of regional protection available): ARIPO (BW, CV, GH, GM, KE, LR, LS, MW, MZ, NA, RW, SC, SD, SL, ST, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, ME, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, KM, ML, MR, NE, SN, TD, TG).

Published:

- with international search report (Art. 21(3))
- in black and white; the international application as filed contained color or greyscale and is available for download from PATENTSCOPE

(54) Title: VIDEO CODING AND DECODING INVOLVING MASTER SUBPICTURES

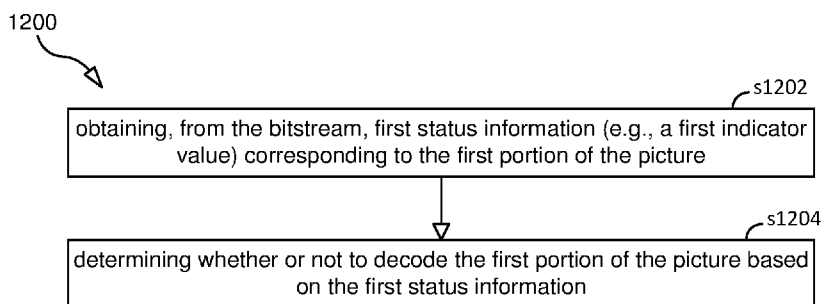


FIG. 12

(57) Abstract: A method for decoding a picture from a bitstream, the picture comprising a first portion and a second portion. The method includes obtaining, from the bitstream, first status information (e.g., a first indicator value) corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion. The method also includes determining whether or not to decode the first portion of the picture based on the first status information.

VIDEO CODING AND DECODING INVOLVING MASTER SUBPICTURES

TECHNICAL FIELD

[001] Disclosed are embodiments related to video coding and decoding.

5

BACKGROUND

[002] Versatile Video Coding (VVC)

[003] VVC and its predecessors Advanced Video Coding (AVC) and High Efficiency Video Coding (HEVC) are block-based video codecs standardized and developed jointly by ITU-T and MPEG. The codecs utilize both temporal and spatial prediction to generate predicted pixel data. VVC, HEVC and AVC are similar in many aspects. Spatial prediction is achieved using intra (I) prediction from within a current picture. Temporal prediction is achieved using uni-directional (P) or bi-directional inter (B) prediction on the block level from previously decoded reference pictures.

[004] In the encoder, the difference between the original pixel data and the predicted pixel data, referred to as the residual, is transformed into the frequency domain, quantized and then entropy coded (e.g., coded using arithmetic coding) before transmitted together with necessary prediction parameters such as prediction mode and motion vectors, also entropy coded (e.g., also arithmetically coded). To reconstruct a picture the decoder obtains the residual for the picture (i.e., the decoder performs decoding (e.g., arithmetic decoding), inverse quantization and inverse transformation to obtain the residual) and then adds the residual for the picture to an intra or inter prediction.

[005] Video, Picture, and Components

[006] A video (a.k.a., video sequence) consists of a series of pictures where each picture consists of one or more components. A picture in a video sequence is sometimes denoted 'image' or 'frame'. Each component in a picture can be described as a two-dimensional rectangular array of picture sample values (or "samples values" for short). It is common that a picture in a video sequence consists of three components; one luma component Y where the sample values are luma values and two chroma components Cb and Cr, where the sample values are chroma

25

values. Other common representations include ICtCb, IPT, constant-luminance YCbCr, YCoCg and others. It is also common that the dimensions of the chroma components are smaller than the luma components by a factor of two in each dimension. For example, the size of the luma component of an HD picture would be 1920x1080 and the chroma components would each have the dimension of 960x540. Components are sometimes referred to as ‘color components’, and other times as ‘channels’.

[007] Blocks and Units

[008] In many video coding standards, such as HEVC and VVC, each component is split into blocks and a coded video bitstream produced by the encoder consists of a series of coded blocks. A block is a two-dimensional array of sample values (or “samples” for short). It is common in video coding that the picture is split into units that cover a specific area of the picture. Each unit consists of all blocks from all components that make up that specific area and each block belongs fully to one unit. The macroblock in H.264 and the Coding unit (CU) in HEVC are examples of units.

[009] A block may be a “transform block” or a “prediction block.” A transform block is a block to which a transform used in coding is applied, and a “prediction block” is a block to which a single prediction mode is applied.

[0010] VVC block structure

[0011] The VVC video coding standard uses a block structure referred to as quadtree plus binary tree plus ternary tree block structure (QTBT+TT) where each picture is first partitioned into square blocks called coding tree units (CTU). The size of all CTUs are identical and the partition is done without any syntax controlling it. Each CTU is further partitioned into coding units (CU) that can have either square or rectangular shapes. The CTU is first partitioned by a quad tree structure, then it may be further partitioned with equally sized partitions either vertically or horizontally in a binary structure to form coding units (CUs). A block could thus have either a square or rectangular shape. The depth of the quad tree and binary tree can be set by the encoder in the bitstream. The ternary tree (TT) part adds the possibility to divide a CU into three partitions instead of two equally sized partitions; this increases the possibilities to use a block structure that better fits the content structure in a picture.

[0012] NAL units

[0013] Both VVC and HEVC define a Network Abstraction Layer (NAL). All the data, i.e., both Video Coding Layer (VCL) data and non-VCL data in HEVC and VVC is encapsulated in NAL units. A VCL NAL unit contains data that represents picture sample values. A non-VCL NAL unit contains additional associated data (a.k.a., metadata) such as parameter sets and supplemental enhancement information (SEI) messages. The NAL unit in VVC and HEVC begins with a header called the NAL unit header. The syntax for the NAL unit header for HEVC starts with a forbidden_zero_bit that shall always be equal to 0 to prevent start code emulations. Without it, some MPEG systems might confuse the HEVC video bitstream with other data, but the 0 bit in the NAL unit header makes all possible HEVC bitstreams uniquely identifiable as HEVC bitstreams. The NAL unit header in VVC is very similar to the one in HEVC but uses 1 bit less for the nal_unit_type and instead reserves this bit for future use. The nal_unit_type, nuh_layer_id and nuh_temporal_id_plus1 code words specify the NAL unit type of the NAL unit that identifies what type of data is carried in the NAL unit, the scalability layer ID and the temporal layer ID for which the NAL unit belongs to. The NAL unit type indicates and specifies how the NAL unit should be parsed and decoded. The rest of the bytes of the NAL unit is payload of the type indicated by the NAL unit type. A bitstream consists of a series of concatenated NAL units (i.e., an ordered set of NAL units).

[0014] A decoder can conclude how the NAL unit should be handled, e.g., parsed and decoded, after looking at the NAL unit header. All VVC or HEVC bitstreams consist of a series of concatenated NAL units. The decoding order is the order in which NAL units shall be decoded, which is the same as the order of the NAL units within the bitstream. The decoding order may be different from the output order, which is the order in which decoded pictures are to be output, such as for display, by the decoder.

[0015] Slices

[0016] The concept of slices in HEVC divides a picture into independently coded slices, where decoding of one slice in a picture is independent of other slices of the same picture. Different coding types could be used for slices of the same picture, i.e. a slice could either be an I-slice, P-slice or B-slice. One purpose of slices is to enable resynchronization in case of data loss. In HEVC, a slice is a set of CTUs.

[0017] In VVC, a slice is defined as an integer number of complete tiles or an integer number of consecutive complete CTU rows within a tile of a picture that are exclusively contained in a single NAL unit. In VVC, a picture may be partitioned into either raster scan slices or rectangular slices. A raster scan slice consists of a number of complete tiles in raster scan order. A rectangular slice consists of a group of tiles that together occupy a rectangular region in the picture or a consecutive number of CTU rows inside one tile. Each slice has a slice header comprising syntax elements. Decoded slice header values from these syntax elements are used when decoding the slice. Each slice is carried in one VCL NAL unit (a.k.a., coded slice NAL unit). In an early draft of the VVC specification, slices were referred to as tile groups.

[0018] Slice and macroblock structure in earlier standards

[0019] The H.263 and H.264 standards use the macroblock entity which is a 16x16 block size processing unit and define a slice as an integer number of macroblocks in a particular order. H.263 and H.264 use the term slice in a different context compared to HEVC and VVC.

[0020] Annex K in H.263 specification describes the optional slice structured mode which includes two submodes:

[0021] (1) The Rectangular Slice submode (RS). When RS is in use, the slice shall occupy a rectangular region of width specified by the slice width indication (SWI) parameter of the slice header in units of macroblocks, and contains a number of macroblocks in scanning order within the rectangular region. When the Rectangular Slice submode is not in use, the SWI field is not present in the slice header and a slice contains a number of macroblocks in scanning order within the picture as a whole.

[0022] (2) The Arbitrary Slice Ordering submode (ASO). When ASO is in use, the slices may appear in any order within the bitstream. When ASO is not in use, the slices must be sent in the (unique) order for which the MacroBlock Address (MBA) field of the slice header is strictly increasing from each slice to each subsequent slice in the picture.

[0023] The H.263 specification describes a coded macroblock indication (COD) as one bit at the start of macroblock syntax, which when set to “0” signals that the macroblock is coded and if set to “1”, signals that no further information is transmitted for this macroblock and the

macroblock is handled as “skipped.” Skipped macroblocks in H.263 are decoded as INTER macroblocks with motion vector for the whole block equal to zero and with no coefficient data.

[0024] The H.264 specification (T-REC-H.264-202108) also defines a skipped macroblock as a macroblock for which no data is coded other than an indication that the macroblock is to be decoded as “skipped.” This indication may be common to several macroblocks. The elements such as motion vector components and reference indices are derived for the skipped macroblock according to some defined derivation processes.

[0025] The Slice data semantics in H.264 includes:

TABLE 1

mb_skip_run specifies the number of consecutive skipped macroblocks for which, when decoding a P or SP slice, **mb_type** shall be inferred to be P_Skip and the macroblock type is collectively referred to as a P macroblock type, or for which, when decoding a B slice, **mb_type** shall be inferred to be B_Skip and the macroblock type is collectively referred to as a B macroblock type. The value of **mb_skip_run** shall be in the range of 0 to $\text{PicSizeInMbs} - \text{CurrMbAddr}$, inclusive.

mb_skip_flag equal to 1 specifies that for the current macroblock, when decoding a P or SP slice, **mb_type** shall be inferred to be P_Skip and the macroblock type is collectively referred to as P macroblock type, or for which, when decoding a B slice, **mb_type** shall be inferred to be B_Skip and the macroblock type is collectively referred to as B macroblock type. **mb_skip_flag** equal to 0 specifies that the current macroblock is not skipped.

[0026] Tiles

[0027] The VVC video coding standard includes a tool called tiles that divides a picture into rectangular spatially independent regions. Tiles in VVC are similar to the tiles used in HEVC. Using tiles, a picture in VVC can be partitioned into rows and columns of CTUs where a tile is an intersection of a row and a column.

[0028] The tile structure is signaled in the picture parameter set (PPS) by specifying the thicknesses of the rows and the widths of the columns. Individual rows and columns can have different sizes, but the partitioning always span across the entire picture, from left to right and top to bottom respectively.

[0029] There is no decoding dependency between tiles of the same picture. This includes intra prediction, context selection for coding and motion vector prediction. One exception is that in-loop filtering dependencies are generally allowed between tiles.

[0030] In the rectangular slice mode in VVC, a tile can further be split into multiple slices where each slice consists of a consecutive number of CTU rows inside one tile. FIG. 1A shows an example of a VVC tile partitioning and FIG. 1B shows an example of a rectangular slice partitioning using the VVC tile partitioning.

[0031] Subpictures

[0032] Subpictures are supported in VVC where a subpicture is defined as a rectangular region of one or more slices within a picture. This means a subpicture contains one or more slices that collectively cover a rectangular region of a picture. In the VVC specification subpicture location and size are signaled in the Sequence Parameter Set (SPS). Boundaries of a subpicture region may be treated as picture boundaries (excluding in-loop filtering operations) conditioned to a per-subpicture flag `subpic_treated_as_pic_flag[ i ]` in the SPS. Also loop-filtering on subpicture boundaries is conditioned to a per-subpicture flag `loop_filter_across_subpic_enabled_flag[ i ]` in the SPS.

[0033] This is done by a first codeword, `sps_num_subpics_minus1`, that specifies how many subpictures there are. Then there is a loop that for each subpicture index `i` specifies the location and size for a subpicture. This loop associates each subpicture to its subpicture index value `i`. The SPS also specifies for each subpicture whether its boundaries shall be treated as picture boundaries or not for the motion compensation and/or in-loop filtering processes. Boundaries of a subpicture region may be treated as picture boundaries (excluding in-loop filtering operations) conditioned to a per-subpicture flag `subpic_treated_as_pic_flag[ i ]` in the SPS. Also loop-filtering on subpicture boundaries is conditioned to a per-subpicture flag `loop_filter_across_subpic_enabled_flag[ i ]` in the SPS.

[0034] In the VVC slice header, there is one subpicture ID syntax element and one slice address syntax element. The subpicture ID specifies the subpicture ID of the subpicture and the slice address specifies the slice address within the subpicture. All slices that belong to the same subpicture must have the same subpicture ID value. In order to support bitstream extraction and merge operations without a need to rewrite slice headers, there is a mapping mechanism in VVC

between subpicture ID values and subpicture index values. This mapping is signalled by either the `sps_subpic_id[i]` or the `pps_subpic_id[i]` syntax elements in the SPS and PPS, respectively. For example, a `sps_subpic_id[1]` value equal to 3 means that subpicture ID value 3 is mapped to the subpicture index value 1. By using this mapping mechanism, only SPS or PPS syntax elements needs to be rewritten during a bitstream extraction or merge operation. Table 2 shows the subpicture syntax in the SPS in VVC.

TABLE 2 – Subpicture syntax in the SPS in the VVC specification

<code>seq_parameter_set_rbsp() {</code>	Descriptor
<code>...</code>	
<code>sps_subpics_info_present_flag</code>	<code>u(1)</code>
<code>if(sps_subpics_present_flag) {</code>	
<code>...</code>	
<code>sps_num_subpics_minus1</code>	<code>ue(v)</code>
<code>for(i = 0; sps_num_subpics_minus1 > 0 && i <=</code> <code>sps_num_subpics_minus1; i++) {</code>	
<code>...</code>	
<code>sps_subpic_ctu_top_left_x[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_ctu_top_left_y[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_width_minus1[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_height_minus1[i]</code>	<code>u(v)</code>
<code>...</code>	
<code>sps_subpic_treated_as_pic_flag[i]</code>	<code>u(1)</code>
<code>sps_loop_filter_across_subpic_enabled_flag[i]</code>	<code>u(1)</code>
<code>...</code>	
<code>}</code>	
<code>}</code>	
<code>...</code>	

[0035] Bitstream extraction and merge operations

10 **[0036]** Some video applications involve operations made on bitstreams, including bitstream extraction and bitstream merge operations. A bitstream extraction operation takes an input bitstream as input and produces an output bitstream, where the output bitstream represents a subset of the video content of the input bitstream. A bitstream merge operation takes multiple input bitstreams as input and produces a single output bitstream, where the output bitstream

represents the union of the video content of the input bitstreams. For bitstream merge to work, it is very common that the input bitstreams are created with merge support in mind. Bitstream merge operations where the bitstreams are encoded fully independently by e.g. different encoders is typically not possible.

5 **[0037]** It is desirable that the bitstream format supports bitstream extraction and merge operations without requiring low-level bitstream changes, since low-level bitstream changes infer substantial implementation burden and increased computational complexity. One common design goal is to enable these operations without any need to manipulate coded slice data. For example, in VVC it would be acceptable to re-write parameter set data but not acceptable to re-
10 write slice data or picture header data.

[0038] For extraction and merge to work, the video format needs to support spatial partitioning of the content into independently coded segments. Slices, tiles and subpictures are examples of such segments. In HEVC, motion-constrained tile sets (MCTSs) are typically used, where slices containing one or more complete tiles are encoded such that the slices are
15 independent. Special care is taken into the HEVC motion compensation processes, where the HEVC encoder needs to select motion parameters carefully so that the slices become independent, hence the name MCTS. In VVC, the subpicture partitioning type was introduced, for which support of less complex bitstream extraction and merge handling was a key reason. Compared to HEVC MCTSs, VVC subpictures are more compression efficient and do not
20 require special care related to motion compensation.

[0039] Bitstream extraction and merge operations for subpictures in VVC could for instance comprise extracting one or more subpictures from a first bitstream, extracting one or more subpictures from a second bitstream and merging the extracted subpictures into a new third bitstream.

25 **[0040]** Picture order count (POC)

[0041] Pictures in HEVC are identified by their picture order count (POC) values, also known as full POC values. Each slice contains a code word, `pic_order_cnt_lsb`, that shall be the same for all slices in a picture. `pic_order_cnt_lsb` is also known as the least significant bits (lsb) of the full POC since it is a fixed-length code word and only the least significant bits of the full
30 POC is signaled. Both encoder and decoder keep track of POC and assign POC values to each

picture that is encoded/decoded. The `pic_order_cnt_lsb` can be signaled by 4-16 bits. There is a variable `MaxPicOrderCntLsb` used in HEVC which is set to the maximum `pic_order_cnt_lsb` value plus 1. This means that if 8 bits are used to signal `pic_order_cnt_lsb`, the maximum value is 255 and `MaxPicOrderCntLsb` is set to $2^8 = 256$. The picture order count value of a picture is called `PicOrderCntVal` in HEVC. Usually, `PicOrderCntVal` for the current picture is simply called `PicOrderCntVal`.

[0042] Reference picture management

[0043] Reference picture management in HEVC is done using reference pictures sets (RPS). The reference picture set is a set of reference pictures that is signaled in the slice headers. When the decoder has decoded a picture, it is put together with its POC value in a decoded picture buffer (DPB). When decoding a subsequence picture, the decoder parses the RPS syntax from the slice header and constructs lists of reference picture POC values.

[0044] The reference picture management in the VVC specification differs slightly from the one in HEVC. In HEVC, the RPS is signaled and the reference picture lists to use for Inter prediction is derived from the RPS. In the VVC specification, the reference pictures lists (RPL) are signaled and the RPS is derived. However, in both specifications, signaling of what pictures to keep in the DPB, what pictures should be short-term and long-term is done. Using POC for picture identification and determination of missing reference pictures is done the same in both specifications.

[0045] Parameter Sets

[0046] HEVC and VVC specifies three types of parameter sets, the picture parameter set (PPS), the sequence parameter set (SPS) and the video parameter set (VPS). The PPS contains data that is common for a whole picture, the SPS contains data that is common for a coded video sequence (CVS) and the VPS contains data that is common for multiple CVSs, e.g. data for multiple scalability layers in the bitstream.

[0047] VVC also specifies one additional parameter set, the adaptation parameter set (APS). The APS carries parameters needed for the adaptive loop filter (ALF) tool, the luma mapping and chroma scaling (LMCS) tool and the scaling list tool.

[0048] Both HEVC and VVC allow certain information (e.g. parameter sets) to be provided by external means. “By external means” should be interpreted as the information is not provided in the coded video bitstream but by some other means not specified in the video codec specification, e.g. via metadata possibly provided in a different data channel, as a constant in the decoder, or provided through an API to the decoder.

[0049] Decoding capability information (DCI)

[0050] In VVC there is a DCI NAL unit. The DCI specifies information that does not change during the decoding session and may be good for the decoder to know about early and upfront, such as profile and level information. The information in the DCI is not necessary for operation of the decoding process. In drafts of the VVC specification the DCI was called decoding parameter set (DPS).

[0051] The decoding capability information may also contain a set of general constraints for the bitstream, that gives the decoder information of what to expect from the bitstream, in terms of coding tools, types of NAL units, etc. In VVC, the general constraint information can be signaled in the DCI, VPS or SPS.

[0052] Picture header

[0053] In VVC, a coded picture contains a picture header structure. The picture header structure contains syntax elements that are common for all slices of the associated picture. The picture header structure may be signaled in its own non-VCL NAL unit with NAL unit type PH_NUT or included in the slice header given that there is only one slice in the coded picture. This is indicated by the slice header syntax element `picture_header_in_slice_header_flag`, where a value equal to 1 specifies that the picture header structure is included in the slice header and a value equal to 0 specifies that the picture header structure is carried in its own PH NAL unit. For a CVS where not all pictures are single-slice pictures, each coded picture must be preceded by a picture header that is signaled in its own NAL unit. HEVC does not support picture headers.

[0054] Layers, OLS and OPI

[0055] The value of the `nuh_layer_id` syntax element in the NAL unit header in VVC and HEVC specifies the layer ID to which the NAL unit belongs. In VVC, layers may be coded independently or dependently from each other. When the layers are coded independently, a layer

with e.g., nuh_layer_id 0 may not predict video data from another layer with e.g. nuh_layer_id 1. In VVC, dependent coding between layers may be used, which enables support for scalable coding with SNR, spatial and view scalability.

[0056] The VPS in VVC and HEVC contains scalability layer information that is needed for handling scalable bitstreams. For HEVC, a VPS must be present for all bitstreams, even single-layer ones, while for VVC the VPS may be omitted for single-layer bitstreams. The VPS defines output layer sets (OLSs), where an OLS is a description of a set of layers in the bitstream with indications of which of the layers in the OLS that should be output. In VVC, only the output layers are specified, and the full OLS is derived by using bitstream information for how layers reference other layers. This means that the full OLS is decodable even if all layers that are not included in the OLS are discarded. In other word, no layer in an OLS depends on any layer not in the OLS. This also means that some OLS layers may be required to be decoded but no pictures of those layers are output.

[0057] The operation point information (OPI) in VVC can be used to specify the OLS index of the target OLS for the decoder to decode. The OPI may additionally specify the highest temporal sublayer the decoder should decode. When a bitstream contains many layers and sublayers, the OPI can be useful to tell a decoder what parts of the bitstream to decode and/or what temporal sublayers that should be discarded when decoding. The target OLS and highest temporal sublayer to decode can alternatively be specified by external means. If that happens and there is an OLS in the bitstream, the decoder should use the information provided by external means and ignore the OLS information. The OPI is signaled in its own non-VCL NAL unit in VVC.

SUMMARY

[0058] Certain challenges presently exist. For instance, one problem with the existing technology described above is that all portions of a picture have the same “status” meaning that in the decoding process all decoded sample values in the picture are equally “valued.” The existing technology does not allow decoding only a portion of the picture and refraining from decoding another portion of the picture for a conformant decoder. In other words, the notion of different portions of the picture having different priority, importance, or status is not recognized in the existing technology. Hence, the existing technology does not allow a conformant decoder

to decode only portions of the picture based on, for example, the amount of resources (e.g., memory or processing power) available to the decoder, device settings, and/or preferences.

[0059] Another problem with the existing technology is that the existing skip mechanisms are acting only as encoding tools. For instance, the concept of skipped macroblocks in H.263 and H.264 is based on refraining from signalling sample values for a macroblock but encoding a skip indication and then on the decoder side have a decoding process to decode the macroblock as “skipped,” including a derivation process for motion vector components and reference indices for the skipped macroblocks in order to decode the sample values. This usage of skipped macroblocks as a coding tool only saves bits when the macroblock can be reconstructed on the decoder side by means other than signalling the sample values, rather than providing flexibility for the decoder to decode only portions of the picture indicated with a particular “status” in the bitstream. One can also say that the skip indication in H.263 and H.264 really represent a coding option or mode for the macroblock. In that sense, the macroblock is coded using the cheapest mode in terms of bit cost. It is not the case that the skipped macroblocks are not coded, they are coded by the skip indication.

[0060] Another problem with the existing technology is that the existing skip mechanisms applicable to portions of picture are bit saving mechanisms and not a mechanism to facilitate low delay use cases. It is not supported in the existing image and video coding standards to not decode parts of the picture, in order to satisfy the low delay or low latency requirements.

[0061] In applications such as video conferencing, sports broadcasting, and remote observation of a critical mission, there are use case scenarios where part of the picture is the most important part of the content, and, from a user perspective, being able to have that portion of the picture continuously decoded and displayed for every picture is preferred over not decoding and displaying entire frames. In the existing technology, in scenarios such as when one or more VCL NAL units in a picture are lost, or the decoder resources become scarce at one point or in case of connectivity or bandwidth issues during a session, one or more full pictures could be not decoded and consequently not displayed entirely, resulting in pauses and jumps in the decoded and displayed video sequence.

[0062] The video content in some applications has several independent portions in one frame. The existing technology such as MCTS in HEVC or independent slices and subpictures in VVC facilitate more efficient and flexible encoding and decoding of this type of content but still a picture needs to be decoded fully in all those independent portions before being displayed and hence it is not possible to decode and display only selective parts of the picture.

[0063] Accordingly, in one aspect there is provided a method for decoding a picture from a bitstream, the picture comprising a first portion and a second portion. The method includes obtaining, from the bitstream, first status information (e.g., a first indicator value) corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion. The method also includes determining whether or not to decode the first portion of the picture based on the first status information.

[0064] In another aspect there is provided a method for producing a bitstream comprising a picture (i.e., an encoded picture), the picture comprising a first portion and a second portion. The method includes determining whether the first portion is required to be decoded by a decoder. The method also includes, as a result of determining that the first portion is required to be decoded by a decoder, producing encoded status information for the first portion and including the encoded status information in the bitstream, where the encoded status information for the first portion does not correspond to the second portion, and the encoded status information is for informing the decoder that the decoder should not refrain from decoding the first portion of the picture.

[0065] In another aspect there is provided a method performed by a network node. The method includes receiving i) a first unit (e.g., a first VCL NAL unit) comprising encoded picture sample values corresponding to a first portion of a picture and ii) first status information (e.g., encoded first status information) corresponding to the first portion of the picture, wherein the first status information does not correspond to a second portion of the picture. The method also includes receiving a second unit (e.g., second VCL NAL unit) comprising encoded picture sample values corresponding to the second portion of the picture. The method also includes based on the first status information, deciding whether or not to forward the first unit to a receiving device.

[0066] In another aspect there is provided a computer program comprising instructions which when executed by processing circuitry of an apparatus causes the apparatus to perform any of the methods disclosed herein. In one embodiment, there is provided a carrier containing the computer program wherein the carrier is one of an electronic signal, an optical signal, a radio
5 signal, and a computer readable storage medium. In another aspect there is provided an apparatus that is configured to perform the methods disclosed herein. The apparatus may include memory and processing circuitry coupled to the memory.

[0067] An advantage of the embodiments disclosed herein is that they provide flexibility in choosing which portions of a picture to decode, which is useful when not all portions of the
10 picture are of interest or when there are insufficient resources for decoding the full picture. This introduces a new trade-off different from the existing technology where only full pictures can be decoded.

[0068] The embodiments further have the advantage of allocating resources (time, memory) to the more critical portions of the picture for instance in a low delay/low latency setup.
15 If the decoder is not going to decode at least some portions of a picture, that part of the signal can be skipped entirely, hence smaller buffers could be used.

[0069] Another advantage is that, if only some portions of the picture are going to be displayed or used in an application and not the fully decoded picture, then the decoder can decode only the necessary portions of the picture and not the unnecessary portions. The proposed
20 solution in this case will be more resource efficient and will reduce the unnecessary processing and power consumption.

[0070] Another advantage is that the user experience is improved in certain applications by prioritizing the decoding and display of the critical portion of the picture (e.g., in a sports video the critical portion may be player with the ball) over risking not being able to decode
25 whole frames due to causes such as packet loss or sudden bandwidth reduction especially in a low delay setup.

BRIEF DESCRIPTION OF THE DRAWINGS

[0071] The accompanying drawings, which are incorporated herein and form part of the specification, illustrate various embodiments.

- [0072] FIG. 1A shows an example of a tile partitioning in VVC.
- [0073] FIG. 1B shows an example of a rectangular slice partitioning using the tile partitioning in VVC.
- [0074] FIG. 1C illustrates a system according to an embodiment.
- 5 [0075] FIG. 2 is a schematic block diagram of an encoder according to an embodiment.
- [0076] FIG. 3 is a schematic block diagram of a decoder according to an embodiment.
- [0077] FIG. 4A illustrates an example picture.
- [0078] FIG. 4B illustrates an example of a first set of portions and a second set of portions.
- 10 [0079] FIG. 5 illustrates an example where two different subpicture layouts using the same master subpicture layout indicate two different portions of the picture with master status.
- [0080] FIG. 6 illustrates an example of how an update on the master subpicture layout affects the portion of the picture with master status.
- [0081] FIG. 7 illustrates an example where the picture is partitioned into subpictures 0 to 15 5 and subpictures 0 to 2 are indicated as master subpictures.
- [0082] FIG. 8 illustrates example master areas.
- [0083] FIG. 9 illustrates a subpicture being indicated as a master subpicture if it is fully or partially overlapping with the part of the picture indicated as master area.
- [0084] FIG. 10 illustrates a texture and depth picture with an ROI.
- 20 [0085] FIG. 11 illustrates a system according to an embodiment.
- [0086] FIG. 12 is a flowchart illustrating a process according to an embodiment.
- [0087] FIG. 13 is a flowchart illustrating a process according to an embodiment.
- [0088] FIG. 14 is a flowchart illustrating a process according to an embodiment.
- [0089] FIG. 15 is a schematic block diagram of a decoder according to an embodiment.
- 25 [0090] FIG. 16 is a block diagram of an encoding apparatus according to an embodiment.

DETAILED DESCRIPTION

[0091] FIG. 1C illustrates a system 100 according to an embodiment. System 100 includes an encoder 102 and a decoder 104, wherein encoder 102 is in communication with decoder 104 via a network 110 (e.g., the Internet or other network). Encoder 102 encodes a source video sequence 101 into a bitstream comprising an encoded video sequence and transmits the bitstream to decoder 104 via network 110. In some embodiments, encoder 102 is not in communication with decoder 104, and, in such an embodiment, rather than transmitting bitstream to decoder 104, the bitstream is stored in a data storage unit. Decoder 104 decodes the pictures included in the encoded video sequence to produce video data for display and/or further image processing (e.g. a machine vision task). Accordingly, decoder 104 may be part of a device 103 having an image processor 105 and/or a display 106. The image processor 105 may perform machine vision tasks on the decoded pictures. One such machine vision task may be identifying objects in the picture. The device 103 may be a mobile device, a set-top device, a head-mounted display, or any other device.

[0092] FIG. 2 illustrates functional components of encoder 102 according to some embodiments. It should be noted that encoders may be implemented differently so implementation other than this specific example can be used. Encoder 102 employs a subtractor 241 to produce a residual block which is the difference in sample values between an input block and a prediction block (i.e., the output of a selector 251, which is either an inter prediction block output by an inter predictor 250 (a.k.a., motion compensator) or an intra prediction block output by an intra predictor 249). Then a forward transform 242 is performed on the residual block to produce a transformed block comprising transform coefficients. A quantization unit 243 quantizes the transform coefficients based on a quantization parameter (QP) value (e.g., a QP value obtained based on a picture QP value for the picture in which the input block is a part and a block specific QP offset value for the input block), thereby producing quantized transform coefficients which are then encoded into the bitstream by encoder 244 (e.g., an arithmetic encoder) and the bitstream with the encoded transform coefficients is output from encoder 102. Next, encoder 102 uses the quantized transform coefficients to produce a reconstructed block. This is done by first applying inverse quantization 245 and inverse transform 246 to the transform coefficients to produce a reconstructed residual block and using an adder 247 to add the prediction block to the reconstructed residual block, thereby producing the reconstructed block, which is stored in the reconstruction picture buffer (RPB) 266. Loop filtering by a loop

filter (LF) stage 267 is applied and the final decoded picture is stored in a decoded picture buffer (DPB) 268, where it can then be used by the inter predictor 250 to produce an inter prediction block for the next picture to be processed. LF stage 267 may include any combination of the following three sub-stages: i) a deblocking filter, ii) a sample adaptive offset (SAO) filter, and
5 iii) an Adaptive Loop Filter (ALF).

[0093] FIG. 3 illustrates functional components of decoder 104 according to some embodiments. It should be noted that decoder 104 may be implemented differently so implementations other than this specific example can be used. Decoder 104 includes a bitstream parser 301 that receives the bitstream (e.g., retrieves the bitstream from a file or receives the
10 bitstream via network 110) and parses the bitstream to obtain each NAL unit in the bitstream. When bitstream parser 301 obtains a NAL unit comprising encoded picture sample values (e.g., the encoded quantized transform coefficient values) bitstream parser 301 provides the encoded sample values to a decoder module 361 (e.g., a function that performs arithmetic decoding) that decodes the sample values to obtain decoded sample values (e.g., the quantized transform
15 coefficient values). Decoder 104 also includes a reconstruction stage 398 in which the quantized transform coefficient values are subject to an inverse quantization process 362 and inverse transform process 363 to produce a residual block. This residual block is input to adder 364 that adds the residual block and a prediction block output from selector 390 to form a reconstructed block. Selector 390 either selects to output an inter prediction block or an intra
20 prediction block. The reconstructed block is stored in a RPB 365. The inter prediction block is generated by the inter prediction module 350 and the intra prediction block is generated by the intra prediction module 369. Following the reconstruction stage 398, a loop filter stage 367 applies loop filtering and the final decoded picture may be stored in a decoded picture buffer (DPB) 368 and output to image processor 105. Pictures are stored in the DPB for two primary
25 reasons: 1) to wait for picture output and 2) to be used for reference when decoding future pictures.

[0094] As described above, a challenge presently exists because, presently, a conformant decoder must decode all portions of a picture (i.e., cannot refrain from decoding one or more portions of the picture). In other words, the notion of different portions of the picture
30 having different status (e.g., different importance, different priority, etc.) is not recognized in the existing technology. Hence, the existing technology does not allow a conformant decoder to

refrain from decoding a portion of the picture based on, for example, the status of the portion and the amount of resources (e.g., memory or processing power) available to the decoder, device settings, and/or preferences.

[0095] Accordingly, this disclosure provides for, in one aspect, indicating a status for different portions of a picture. The status for a portion of the picture is proposed to be indicated in the bitstream, and, based on the status, the decoder may choose to decode or not decode the portion (e.g., choose to arithmetic decode or not arithmetic decode the arithmetic encoded data corresponding to the first portion such as quantized residual values).

[0096] Accordingly, as shown in FIG. 15, in one embodiment bitstream parser 301 includes a selector 1501 that, in one embodiment, determines, for one or more picture portions in the bitstream, a status of the portion (e.g., determines whether the picture has a master status, determines the group or groups to which the portion belongs, etc.) and then, based on the status (and possible other information), determines whether to select the portion for decoding (i.e., determines whether to provide the encoded sample values of the portion to decoding module 361). That is, if a picture portion is selected for decoding, then the encoded samples values of that picture portion are provided to decoding module 361, otherwise the encoded samples values are discarded (a.k.a., dropped or sent to the “bit bucket”).

[0097] The status may be indicated for a portion of the picture based on priority, importance, analysis of the content or other preferences related to that portion of the picture.

Hence, a first decoder may decode only those picture portions that are indicated with status A, a second decoder may decode only those picture portions which are indicated with status B, and a third decoder may decode only those picture portions which are indicated with either status A or B.

[0098] In one embodiment, the encoded bitstream includes several subpictures per picture (each subpicture consisting of one or more portions of the picture) and at least one of these subpictures is indicated as having a “master” status (i.e., the subpicture is a master subpicture). In one embodiment a subpicture having a master status means that the subpicture includes content with the highest importance to the user compared to other subpictures in the picture. A decoder with limited decoding resources may then decode only the one or more

portions of the picture corresponding to the master subpicture and not decoding other portions which are not part of a master subpicture.

[0099] Embodiment 1: A Method for Decoding A Picture

[00100] In one embodiment there is provided a method for decoding a picture from a
5 bitstream, wherein the picture comprises a first portion and a second portion. In one
embodiment, the first portion is “independent” from at least the second portion of the picture. As
used herein, a portion of a picture is “independent” from another portion of the picture when the
portion can be reconstructed correctly without any information from the other portion (i.e., the
independent portion does not predict from the other portion). The method comprises obtaining,
10 from the bitstream, first status information (e.g., a first status indicator value that indicates a
status, such as, for example, a priority level) corresponding to (i.e., related to) the first portion.

[00101] The method further includes, determining, based on the first status information
corresponding to the first portion, whether or not to decode the first portion. For instance, in one
embodiment, determining whether or not to decode the first portion based on the first status
15 information comprises determining whether the first status information is equal to a first value
and, if the first status information is equal to the first value, then it is determined to decode the
first portion of the picture. In one embodiment, if the first status information is not equal to the
first value, then it is determined not to decode the first portion (i.e., refrain from decoding the
first portion). In another embodiment, if the first status information is not equal to the first value,
20 then the determination as to whether or not to decode the first portion may be dependent on other
criteria being met (e.g., the decoder has sufficient resources).

[00102] The first value may be a value that is hardcoded in the decoder or obtained by the
decoder from a configuration file or from the bitstream or from user input.

[00103] In one embodiment, the first status information corresponds to the first portion
25 and not to the second portion. In one embodiment a portion of the picture is a subpicture, a slice,
a tile, a CTU, a set of CTUs, a tile set, or a tile group.

[00104] In one embodiment, the status information is the value of an indicator and is
decoded from a syntax element in the bitstream. In one embodiment, the indicator may for

instance be a binary flag where the status information is one of two possible values. In another embodiment, the indicator can encode one of three or more values.

[00105] In this disclosure, a status for a portion of a picture may be defined as a specific value of the indicator associated with that portion signaled in the bitstream. Indicating a status for a portion of a picture may specify that a process in the decoding of that portion or the whole picture shall be invoked. The decoder may choose alternative decoding steps for handling independent portions of a picture based on the status of each portion. One example is an indicator value in the bitstream indicating that a first portion of a picture has the status of “master” and the decoding method for decoding this picture comprises decoding the first portion in response to this portion having the status of “master.” A portion of a picture having a “master” status may be interpreted as that portion is preferred (e.g., prioritized) over other portions of the picture not having “master” status, e.g. in case of limited decoding resources. Another interpretation may be that the portion of the picture having “master” status, is to be decoded, whereas other portions of the picture not having “master” status are not to be or may not be decoded, and for which the bitstream may not even comprise any video coding layer data.

[00106] A portion of the picture with “master” status may be defined as a portion of the picture which is decoded in all flavors of the decoders. This means that the decoding of the portion is not optional, and that it is guaranteed that any correctly functioning compliant decoder must decode (or at least attempt to decode) the portion with “master” status.

[00107] The status may also be seen as a grouping of the one or more portions of the picture. In one example there are at least two portions in the picture and a status A or a status B has been assigned to each portion of the picture where at least one portion has status A and at least one portion has status B. In this example one can say that the picture is divided into two groups of picture portions, one group are portions with status A and another group are portions with status B. In one example the decoder decodes portions with status A and does not decode or may not decode portions with status B in the picture.

[00108] Statuses may bring advantage to the current art by grouping portions according to their properties, such that some portions of the picture sharing a particular property (e.g, portions that belong to the same particular view) are decoded while other portions of the picture not having the particular property (e.g., not belong to the particular view) may not be decoded

depending on other factors (e.g., user preferences, available resources, etc.). The properties may include the following:

[00109] (a) Portion view IDs, e.g. a status would indicate a particular portion view ID, where one or more portions of the picture having a particular status would comprise data for a particular view in a picture, where a view may be a view in a multi-view setup, a view captured at a specific angle of a certain content, such as a region of interest (ROI), or a view of a specific content among multiple contents, and/or

[00110] (b) Type of data, e.g., texture, alpha, depth, point cloud attribute or meta data, etc. E.g., a status value could correspond to the associated picture portion(s) comprising one of texture, alpha or depth data, and/or

[00111] (c) Priority of video data to be used for video analysis. E.g. some objects in a video may be more important to track than others.

[00112] In one example, the first portion is independent from other portions and is a first subpicture. In this example, an indicator value equal to a first value “master subpicture status” is decoded from the bitstream and in response to the indicator value being equal to master subpicture status the first subpicture is decoded.

[00113] As used herein, the phrase “a portion of the picture having a first status” is used interchangeably with the phrase “a first status portion.” As an example, “a subpicture having a master subpicture status” is the same as “a master subpicture” and “a subpicture with a master status.”

[00114] In another example, the picture is partitioned into at least a first portion and a second portion. In this example, a first status for the first portion is indicated with a first indicator value in the bitstream for the first portion and a second status for the second portion is indicated with a second indicator value in the bitstream for the second portion. The indicator values for the first status and the second status may be signaled in the same syntax element or in different syntax elements. The method comprises decoding the first portion of the picture in response to the first indicator value being equal to a first value or the first status being equal to a status A, and not decoding the first portion of the picture in response to the first indicator value being equal to a second value or the first status being equal to a status B, different from status A.

Likewise, the second portion of the picture is decoded in response to the second indicator value being equal to the first value or the second status being equal to a status A, and the second portion of the picture is not decoded in response to the second indicator value being equal to a second value or the second status being equal to status B.

5 **[00115]** In one embodiment, the first portion of the picture is a first subpicture in the picture, the second portion of the picture is a second subpicture in the picture, status A means master subpicture status, and status B means not master subpicture status. For example, when the first status corresponding to the first portion is equal to status A and the second status corresponding to the second subpicture is equal to status B, the status for the first subpicture is a master subpicture status, and the status for the second subpicture is not a master subpicture status.

15 **[00116]** In one embodiment, the first subpicture of the picture is decoded in response to determining that the first subpicture of the picture has a status A where status A is a master subpicture status. In one embodiment, the second subpicture of the picture is not decoded as a result of determining that the second subpicture has a status B, where status B is not master subpicture status. In another embodiment, the second subpicture is not decoded as a result of determining that the second subpicture has a status of “B” and determining a condition is satisfied (e.g., determining that the amount of available memory and/or available processing resources are below a threshold and/or determining that a latency requirement will not be met if an attempt is made to decode the second subpicture).

20 **[00117]** Device 103 may perform all or a subset of the following steps for decoding a picture from a bitstream:

25 **[00118]** (1) obtaining the bitstream (e.g., receiving the bitstream via network 110 or retrieving the bitstream from a storage unit), where the bitstream contains information for decoding a current picture from the bitstream;

[00119] (2) Decoding information from the bitstream indicating that the current picture comprises a first portion (the information may also indicate that the first portion is independent from other portions of the picture);

[00120] (3) obtaining a first indicator value;

[00121] (4) determining whether or not to decode the first portion based on the first indicator value (e.g., decoding the first portion in response to the first indicator value being equal to a first value, and not decoding the first portion in response to the first indicator value not being equal to the first value).

5 [00122] In another embodiment, device 103 may perform all or a subset of the following steps for decoding a picture from a bitstream:

[00123] (1) obtaining a bitstream containing information for decoding a current picture from the bitstream;

10 [00124] (2) Decoding information from the bitstream that the current picture is partitioned into at least a first portion and a second portion;

[00125] (3) obtaining a first indicator value for the first portion (first indicator value may be decoded from the bitstream);

[00126] (4) Determining a first status from the first indicator value;

15 [00127] (5) Decoding the first portion in response to the first indicator value being equal to a first value or the first status being equal to status A;

[00128] (6) Not decoding the first portion in response to the first indicator value being equal to a second value or the first status being equal to status B;

[00129] (7) obtaining a second indicator value for the second portion (the second indicator value may be decoded from the bitstream);

20 [00130] (8) Decoding the second portion in response to the second indicator value being equal to a first value or the second status being equal to status A;

[00131] (9) Not decoding the second portion in response to the second indicator value being equal to a second value or the second status being equal to status B.

25 [00132] In one example, a portion of a picture must have either a first or a second status and the first portion of the picture has a first status, and the second portion of the picture does not have the first status. This may be interpreted as the second portion of the picture has a second status that is not equal to the first status.

[00133] In another example, a picture has more than one portion, every portion of the picture has a first status or some status different than the first status (i.e., some portions may have the first status, while other portions do not have the first status), only the portions with the first status are expressly indicated such that any portion not expressly indicated as having the first status necessarily has another status other than the first status (that is, the other portions of the picture that have not been indicated with the first status are implied to not have the first status). For example, in one embodiment there is at least one portion in the picture with a status A and at least one portion in the picture with a status B and the status A and the status B are not the same.

[00134] In one embodiment, the first indicator value is decoded from a parameter set, a header, or an SEI message. A header may for instance be a picture header, slice header or NAL unit header.

[00135] In one embodiment, the first portion comprises two or more CTUs.

[00136] Encoder 102 may perform all or a subset of the following steps for encoding a picture into a bitstream:

[00137] (1) Determining that the picture comprises one or more portions;

[00138] (2) Determining that a first portion of the picture is indicated by an indicator value wherein the indicator value equal to a first value indicates that the first portion of the picture shall be decoded by the decoder and: (2a) the indicator value not equal to the first value indicates that the first portion of the picture may or may not be decoded by the decoder or (2b) the indicator value equal to a second value indicates that the first portion of the picture may or may not be decoded by the decoder;

[00139] (3) Encoding the indicator value for the first portion of the picture into the bitstream;

[00140] (4) If the indicator value is equal to the first value, encoding the first portion of the picture into the bitstream;

[00141] (5) If the indicator value is not equal to the first value or is equal to the second value, encoding the first portion of the picture into the bitstream or refraining from encoding the first portion of the picture.

[00142] In another embodiment, encoder 102 may perform all or a subset of the following steps for encoding a picture into a bitstream:

[00143] (1) Determining that the picture comprises at least a first portion and a second portion;

5 [00144] (2) Encoding a first indicator value for the first portion of the picture into the bitstream, where the indicator value equal to the first value indicates that the first portion of the picture shall be decoded by the decoder, and the first indicator value not being equal to the first value indicates that the first portion of the picture may or may not be decoded by the decoder;

10 [00145] (3) If the first indicator value is equal to the first value, encoding the first portion of the picture into the bitstream;

[00146] (4) If the first indicator value is not equal to the first value, encoding the first portion of the picture into the bitstream or refraining from encoding the first portion of the picture into the bitstream;

15 [00147] (5) Encoding a second indicator value for the second portion of the picture into the bitstream, where the second indicator value equal to the first value indicates that the second portion of the picture shall be decoded by the decoder, and the second indicator value not being equal to the first value indicates that the second portion of the picture may or may not be decoded by the decoder;

20 [00148] (6) If the second indicator value is equal to the first value, encoding the second portion of the picture into the bitstream;

[00149] (7) If the second indicator value is not equal to the first value, encoding the second portion of the picture into the bitstream or refraining from encoding the second portion of the picture into the bitstream.

25 [00150] Device 103 may in this example perform all or a subset of the following steps for decoding a picture comprising a first portion and a second portion from a bitstream, wherein the portions comprised in the picture are independent:

[00151] (1) Decoding a first value from the bitstream, the first value specifying that the first portion has a status A;

[00152] (2) Decoding a second value from the bitstream, the second value being different than the first value and the second value specifying that the second portion has a status B different than status A;

[00153] (3) Decoding the first portion in response to the first portion having status A; and

5 [00154] (4) Not decoding the second portion in response to the second portion having status B.

[00155] Embodiment 2: Extra actions for filling sample values without decoding

[00156] This embodiment describes additional action(s) the decoder may take for decoding a picture from a bitstream. In this embodiment, the decoder does not decode the first
10 portion in response to the first indicator value not being equal to the first value and in addition the decoder may:

[00157] (1) Generate sample values for the first portion of the picture using, for example: i) a constant sample value, ii) default sample values, and/or iii) previously decoded sample values; and/or

15 [00158] (2) Copy, interpolate or extrapolate sample values for the first portion of the picture from: i) collocated sample values in one or more previously decoded pictures, ii) other sample values in one or more previously decoded pictures, iii) other sample values in the current picture, a pre-determined picture, and/or part of a picture, and the decoder fills the sample values in the first portion with the generated, copied, interpolated, and/or extrapolated sample values.

20 [00159] In one example, two pictures P1 and P2 with the same sample values except in a first portion are encoded into bitstream B1 and bitstream B2, respectively, and the first portion in P1 and the first portion in P2 are indicated not to be decoded by the decoder. Using this embodiment, the decoder ignores the portions of the bitstream B1 and bitstream B2 which represent the first portion, however the decoded pictures P'1 and P'2 will have the same sample
25 values everywhere since the coded values corresponding to the first portion are ignored but the filling process is the same and so the first portion in P'1 and P'2 are filled with the same values.

[00160] In one example, the decoder has determined that the first subpicture is not a master subpicture and in response to that the decoder does not decode the first subpicture and in

addition to that the decoder copies the sample values of a collocating area of a previously decoded picture into the sample values of the first subpicture in the current picture.

[00161] In a variant of this embodiment, it is signalled in the bitstream how to generate, copy, interpolate or extrapolate the sample values for the portions of the picture which are not decoded. According to this embodiment, device 103 may perform all or a subset of the following steps shown in Table 3 for decoding a picture from a bitstream:

TABLE 3

1.	Receiving a bitstream containing information for decoding a current picture from the bitstream
2.	Decoding from the bitstream that the current picture is partitioned into at least a first and a second portion
3.	Decoding an indicator value for the first portion
4.	Determining a status from the indicator value
5.	Decoding the first portion in response to the indicator value being equal to a first value or the status being a first status
6.	Doing the following steps in response to the indicator value being equal to a second value or the status being a second status:
	a. Not decoding the first portion
	b. Decoding a second indicator from the bitstream
	c. Generate or copy sample values for the first portion of the picture using the second indicator value
	d. Filling the sample values of the first portion of the picture with the generated or copied sample values

[00162] In one example an indicator value in the bitstream equal to a first value indicates that the decoder does not decode portions of the picture which do not have master status and the decoder fills the sample values in all portions of the picture which are not master portions with gray. The indicator value equal to a second value in this example indicates that the decoder fills the sample values in all portions of the picture which are not master portions with black.

[00163] In one example the information to decode sample values for the portion which is not a master portion does not even exist in the bitstream.

[00164] Embodiment 3: Changing subpicture layout in the decoded or output picture

[00165] In this embodiment, the decoder rearranges the positioning and or rescales a
5 portion of the picture. In this embodiment, a first portion of a picture is decoded in response to determining that the first portion of the picture has a first status, and a second portion of the picture is not decoded in response to determining that the second portion of the picture has a second status, and in addition the decoder may: (1) relocate at least one of the portions in the picture; (2) scale or rescale at least one of the portions in the picture; and/or (3) change the
10 output resolution of the decoded picture or a decoded portion of the picture.

[00166] In one example, the decoder decodes a first subpicture in the picture in response to the first subpicture having a master subpicture status and the decoder does not decode a second subpicture in the picture in response to the second subpicture not having a master subpicture status, and, in addition the decoder scales the first subpicture to cover the full area in
15 the picture before outputting the picture.

[00167] In one version of this embodiment, the selection of subpictures that are decoded and output are based on the orientation of the display device, where the display device may for instance be a phone or TV set. In an example use case, a picture 400 (see FIG. 4A) consists of 9 subpictures (numbered from 0 to 8 in FIG. 4A) and the picture is larger than what is shown on
20 any screen. If shown vertically, only subpictures 1, 4, and 7 are displayed and if shown horizontally only subpictures 3, 4, and 5 are displayed. In this example, subpictures 1, 4 and 7 could be indicated as master subpictures for a vertical display and subpictures 3, 4 and 5 could be indicated as master subpictures for a horizontal display. Subpictures 0, 2, 6 and 8 are never displayed and in one version there is no VCL data in the bitstream for subpictures 0, 2, 6, 8. In
25 another version, only subpicture 4 has a master subpicture status and subpictures 1, 7 have a status X and subpictures 3, 5 have a status Y. A decoder may then select or be instructed to decode and output the master subpicture and either the subpictures with status X or the subpictures with status Y depending on, for example, orientation of the display.

[00168] This may be done without subpictures, but the advantage could be that the subpictures that will not be displayed do not need to be decoded. It would often be ok to wait one second if the display orientation is flipped.

[00169] According to this embodiment, device 103 may perform all or a subset of the steps shown in Table 4 for decoding a picture from a bitstream:

TABLE 4

1. Obtaining a bitstream containing information for decoding a current picture from the bitstream
2. Decoding from the bitstream that the current picture is partitioned into at least a first and a second portion
3. Decoding from the bitstream a first information about the location and or position of the first and or the second portion.
4. Decoding a first indicator value from the bitstream for the first portion and a second indicator value for the second portion
5. Determining a first status for the first portion based on the first indicator value and a second status for the second portion based on the second indicator value
6. Decoding the first portion in response to the first status being indicated for the first portion
7. Not decoding the second portion in response to the second status being indicated for the second portion.
8. Rescaling, rearranging, resizing, and/or changing the aspect ratio of the first portion so that the location, position, size, and/or aspect ratio of the first portion in the output picture is different from the location, position, size, and/or aspect ratio of the first portion as specified by the first information in the bitstream. The new location, position, size, and/or aspect ratio of the first portion may be specified in the bitstream or may be independent from the bitstream and determined in a pure post processing step.
9. Outputting the decoded picture

[00170] Embodiment 4: Sequence level control on allowing master subpictures

[00171] In this embodiment, it is indicated in the bitstream (e.g., via a control syntax element) whether it is allowed to decode or not decode a portion of a picture based on a signalled indicator value in the bitstream or not. In one variant, one value of a decoded control syntax element means that all portions shall be decoded regardless of their decoded status values.

- 5 Another value of a decoded control syntax element means that portions having a certain status may not or shall not be decoded. In another variant, one value of a decoded control syntax element means that there is no status value syntax elements in the bitstream (i.e. the presence of status value syntax elements in the bitstream is gated by the control syntax element).

- [00172] In a variant of this embodiment, it is indicated in the bitstream whether it is
10 allowed to indicate a status for a portion of a picture in the bitstream or not. In one variant, one value of a decoded control syntax element means that a certain status value shall not be signalled in any status value syntax element in the bitstream. Other status values than the certain status value may be allowed. Another value of a decoded control syntax element means that the certain status value may be signalled by status value syntax elements in the bitstream.

- 15 [00173] The decoded control syntax element may be a sequence level syntax element, and may be located in a parameter set such as a video parameter set or sequence parameter set.

- [00174] In one embodiment a third indicator is decoded from the bitstream and the value of the third indicator being equal to a third value, specifies that the first indicator for at least a first portion of the picture may be present in the bitstream and the value of the third indicator
20 being equal to a fourth value, specifies that the first indicator for at least a first portion of the picture is not present in the bitstream.

[00175] In one embodiment a sequence level indication specifies if the use of master subpictures is allowed.

- [00176] In another example the indication is a flag in the SPS, where the flag equal to 1
25 specifies that master subpictures may be indicated in the bitstream and the flag equal to 0 specifies that master subpictures shall not be indicated in the bitstream.

[00177] Embodiment 5: Layout information

[00178] In this embodiment, the bitstream contains i) first layout information (or a “first layout” for short) where the first layout specifies a first set of picture portions and ii) second

layout information (or a “second layout” for short) wherein the second layout specifies at least a second set of picture portions where at least one status is indicated for at least one of the portions in the second set.

[00179] In one example, the first layout signalled in the bitstream specifies, for each portion of the picture, the x-y coordinates of a corner of the portion and the portion’s dimensions (e.g., width and height). The x-y coordinates may be in tile units or block units.

[00180] In one example, the second layout specifies how the picture portions in the first set are grouped according to their status wherein the portions in each group have the same status. In a preferred version of this embodiment the first layout and the second layout differ.

[00181] FIG. 4B illustrates an example of a first set of portions (i.e., portions 0, 1, 2, and 3) that are specified by the first layout information and a second set of portions (i.e., portions 0 and 1) that are specified by the second layout information. In one embodiment, each portion that is a member of the second set of portions is treated as having a certain status (e.g., master status) by mere virtue of the fact that the portion is a member of the second set. But in another embodiment, the second layout not only expressly specifies which portions belong to the second set of portions, but also expressly specifies the status assigned to the second set of portions (e.g., master status), and, thereby, implicitly assigned to each portion that is a member of the second set of portions.

[00182] That is, for example, in one specific embodiment in which portions are subpictures, the first layout specifies subpicture partitioning and the second layout specifies a group of the subpictures with the master status. This is illustrated in FIG. 4B, where the first layout specifies partitioning of the picture into subpicture 0 to 3 and the second layout specifies subpicture 0 and 1 as the group of subpictures that have master status.

[00183] In one embodiment, the first layout and second layout are decoded from the bitstream. In one embodiment, decoding the first layout comprises decoding a code word specifying the total number of picture portions and then decoding, for each portion, a set of information specifying the portion (e.g., x-y coordinates, dimensions, and ID). Table 5 shows an exemplary syntax for the first layout.

TABLE 5

Exemplary syntax for identifying picture portions

parameter_set_rbsp() {	Descriptor
...	
num_portions_minus1	u(8)
for(i = 0; i <= num_portions_minus1; i++) {	
portion_ctu_top_left_x[i]	u(v)
portion_ctu_top_left_y[i]	u(v)
portion_width_minus1[i]	u(v)
portion_height_minus1[i]	u(v)
portion_ID[i]	u(v)
...	
}	
...	

[00184] In one embodiment, decoding the second layout comprises decoding a code word specifying the size of the second set of portions (i.e., the total number of portions that are included in the second set), and then decoding S number code words, where each one of the S code words identifies a portion that is a member of the second set of portions. Table 6A shows an exemplary syntax for the second layout:

TABLE 6A

Exemplary syntax for specifying the portions that belong to the second set

parameter_set_rbsp() {	Descriptor
...	
S_minus1	u(8)
for(i = 0; i <= S_minus1; i++) {	
second_set [i]	u(v)
}	
...	

[00185] That is, the array “second_set” will have S elements. In one embodiment, each element of the second_set array (which may also be called the “master_set” array) will include a portion ID. In this way the portions that belong to the second set can be identified.

[00186] Table 6B shows another exemplary syntax for the second layout:

TABLE 6B

Exemplary syntax for Specifying the portions that belong to the second set

parameter_set_rbsp() {	Descriptor
...	
second_set	u(v)
...	

[00187] In the embodiment shown in Table 6B, the variable `second_set` is a bitmask (i.e., a string of bits) identifying the portions that are a member of the second set. For instance, if the total number of portions included in the first set is N, the bitmask will have a length of N where each bit in the bitmask corresponds to one of the N portions, and, if the bit for a portion is set to 1, then the portion is a member of the `second_set`, otherwise it is not a member of the second set. As a specific example, if `second_set = 0101` then portions 0 and 2 are a member of the second set, and if `second_set = 1000`, then only portion 3 is a member of the second set.

[00188] Alternatively, it is indicated for each portion included in the first set of portions whether or not the portion is a member of the second set of portions. In this alternative, the number of portions included in the second set need not be indicated by a dedicated syntax element but derived from the indications for each portion. Table 7 shows an exemplary syntax for this embodiment.

TABLE 7

parameter_set_rbsp() {	Descriptor
...	
num_portions_minus1	u(8)
for(i = 0; i <= num_portions_minus1; i++) {	
portion_ctu_top_left_x[i]	u(v)
portion_ctu_top_left_y[i]	u(v)
portion_width_minus1[i]	u(v)
portion_height_minus1[i]	u(v)
portion_ID[i]	u(v)
portion_member_of_second_set_flag[i]	u(1)
...	
}	
...	

[00189] In another embodiment, a first portion that is included in the first set of portions is included in a second set of portions (a.k.a., second group or status A) and a second portion that is included in the first set is included in a third set of portions (a.k.a., third group or status B). In such an embodiment, decoding the second layout comprises decoding a first code word specifying a total number of groups (NG) (i.e., total number of distinct statuses), and, for each group G_i , $0 \leq i \leq NG-1$ in the second layout, the portions of the first layout that are included in G_i are identified. Table 8 shows an exemplary syntax for this embodiment.

TABLE 8

parameter_set_rbsp() {	Descriptor
...	
NG_minus1	u(8)
for(i = 0; i <= NG_minus1; i++) {	
group_members[i]	u(v)
group_status[i]	u(v)
}	
...	

[00190] As demonstrated by Table 8, the “group_members” array will have NG elements.

In one embodiment, each element of the group_members array will include a bitmask (i.e., a string of bits) identifying the portions that are a member of the group. For instance, if the total number of portions included in the first set is N, then each element in the group_members array will be an N bit bitmask where each bit corresponds to one of the N portions, and, if the bit for a portion is set to 1, then the portion is a member of the group, otherwise it is not a member of the group. As a specific example, if group_members[0] = 0101 then portions 0 and 2 are a member of group 0, and if group_members[1] = 1000, then only portion 3 is a member of group 1.

[00191] In another embodiment, each element of the group_members array will include a list of portion IDs (as shown in Table 7, each portion is given a unique portion ID). That is, the second layout uses portion IDs that are different from portion index but unique to each portion to indicate the groupings. In one example, the portion index is specified by index i in the Table 5 above, as the loop counter value for each portion in that particular order, so the first portion gets index 0 and the next portion gets index 1, etc. The portion ID is then signalled in the bitstream (e.g., as portion_ID) to create a mapping between each portion index into a unique ID value. A

benefit of this method could be that a status update is possible with only updating the mapping between index and IDs, without the need to resend the portion partitioning or status values. In one example of this variant there is a one-to-one mapping of a portion index into a portion ID and the indication is signalled in a loop over the portion IDs.

- 5 **[00192]** Additionally, as shown in Table 8, the bitstream may include, for each one of the NG groups, a code word containing information indicating the status assigned to the group (but this is not needed in embodiment where each group has an implicit status). Accordingly, a decoder may obtain from the bitstream: picture portion information specifying a set of picture portions, group membership information indicating the number of picture portion groups and, for
10 each picture portion group, the members of the group and the status assigned to the group, if needed.

- [00193]** Alternatively, it is indicated for each portion P1 of the first layout which group in the second layout portion P1 belongs to. In this alternative, the number of groups there are in the second layout may not be indicated by a dedicated syntax element but derived from the
15 indications for each portion P1. Table 9 shows an exemplary syntax for this embodiment.

TABLE 9

parameter_set_rbsp() {	Descriptor
...	
num_portions_minus1	u(8)
for(i = 0; i <= num_portions_minus1; i++) {	
portion_ctu_top_left_x[i]	u(v)
portion_ctu_top_left_y[i]	u(v)
portion_width_minus1[i]	u(v)
portion_height_minus1[i]	u(v)
portion_groupID[i]	u(v)
...	
}	
...	

- [00194]** In the example above, the second layout is signalled in the same loop as the first layout.

[00195] FIG. 5 illustrates an example where two different subpicture layouts using the same master subpicture layout indicate two different portions of the picture with master status (the portions with master status are the portions within the dotted rectangle). One advantage of this embodiment is that to update the area of the picture that has a status A it is enough to update the picture portions which are in the group specified with status A and the partitioning of the picture into portions does not need to be modified.

[00196] FIG. 6 illustrates an example of how an update on the master subpicture layout affects the portion of the picture with master status.

[00197] In one example a subpicture layout is signalled in the bitstream and a flag or an index syntax element in a loop in a header or parameter set (e.g. the PH or PPS) specifies which of the subpictures are master subpictures. In one variant, an index syntax element in the SPS or PPS or PH specifies which of the subpictures are master subpictures. A similar example uses slices as portions of the picture.

[00198] In one example, a subpicture layout and a master subpicture layout are signalled in the SPS.

[00199] In another example a subpicture layout is signalled in the SPS and a master subpicture layout (a.k.a., master subpicture identification information) is signalled in the PPS. In this example the master subpicture layout may be updated in the bitstream by signalling a new PPS with a new master subpicture layout (i.e., new information identifying which portions are master portions) without the need to update the subpicture layout in the SPS. In another example the subpicture layout is signalled in the SPS or PPS and the master subpicture is signalled in a picture header or slice header. In this example the master subpicture layout may be updated for each picture or slice.

[00200] In another example the second layout is signalled in the same loop as the first layout is signalled. For instance, an indication that the i -th subpicture is the master subpicture is signalled in the same loop where subpicture layout is signalled. In one example there is a flag for each subpicture in the SPS subpicture layout to specify if this subpicture is a master subpicture or not. Table 10 shows an exemplary syntax for indicating whether the i -th subpicture is a master subpicture using the flag `sps_master_subpic_flag [ i ]` in the same loop where the subpictures layout is signalled.

TABLE 10

Exemplary syntax for indicating whether the i-th subpicture is a master subpicture in the same loop where the subpictures layout is signalled.

seq_parameter_set_rbsp() {	Descriptor
...	
sps_num_subpics_minus1	u(8)
for(i = 0; i <= sps_num_subpics_minus1; i++) {	
sps_subpic_ctu_top_left_x[i]	u(v)
sps_subpic_ctu_top_left_y[i]	u(v)
sps_subpic_width_minus1[i]	u(v)
sps_subpic_height_minus1[i]	u(v)
sps_master_subpic_flag[i]	u(1)
...	
}	
...	

[00201] In a variant of this embodiment, a first layout may become units for specifying a second layout. In one example, the picture is partitioned into a grid of rectangular subpictures and a second layout uses the units of the grid of the first layout to specify portions of the picture which have a master status. FIG. 7 illustrates an example where the picture is partitioned into subpictures 0 to 5 and subpictures 0 to 2 are indicated as master subpictures by signalling the index of the start and the index of the end subpicture that are indicated as master subpictures, so in this example values 0 and 2 are signalled.

[00202] Embodiment 6: Signalling a Master Area

[00203] In this embodiment, the spatial position of a master area is indicated in the picture independent from the subpicture partitions and the subpicture(s) located in that master area will be the master subpicture(s). A master area is defined as an area in the picture where all portions intersecting with this area have a master status. An example is illustrated in FIG. 8 where a master area is indicated in the left. A first and a second subpicture layouts are assumed where for the first subpicture layout with 4 subpictures in this example subpicture 0 is indicated a master subpicture since subpicture 0 is located in the master area. For the second subpicture layout this time with 6 subpictures both subpictures 0 and 1 are in the master area and so both are indicated as master subpictures.

[00204] In a variant of this embodiment a subpicture is indicated as a master subpicture if it is fully or partially overlapping with the part of the picture indicated as master area, as illustrated in FIG. 9, where subpictures 0, 1, 3 and 4 are indicated master subpictures since either fully or partially overlap with the part of the picture indicated as master area.

5 **[00205]** Embodiment 7: There is exactly one portion in the picture with status A

[00206] In this embodiment there is one portion in the picture that has status A and when one portion in the picture is indicated with status A, no other portion in the picture is indicated with status A. One example is when it is indicated in the bitstream that only one subpicture has a master status.

10 **[00207]** In one variant of this embodiment, when one portion in the picture is indicated with status A, it is inferred that all other portions of the picture do not have status A.

[00208] In one embodiment there is only one subpicture in the picture that has a master status and all other subpictures in the picture do not have a master status. In one example an identifier value is signalled in a header or parameter set to specify which subpicture is the one
15 subpicture with master status.

[00209] In another variant of this embodiment there is exactly one portion in the picture that has status A. In one embodiment a syntax element in a header or parameter set equal to a first value indicates that there is one master subpicture per picture.

[00210] In a variant of this embodiment, it is the first portion of the picture that is
20 indicated to have status A. The first portion may be defined as the first portion in the bitstream or the first portion in a raster scan order or the portion with the lowest ID. In one example a flag in a parameter set specifies that the subpicture with subpicture index equal to 0 is a master subpicture.

[00211] In one example, exactly one portion in the picture has status A or a master status,
25 and the decoder for decoding the picture in this example takes the following steps:

[00212] (1) Decoding a first index value from a first index syntax element in the bitstream;

[00213] (2) Determining, using the index value, the portion of the picture with the status equal to status A or the master status;

[00214] (3) Decoding the portion of the picture with the status equal to status A in response to the first portion having status A or the master status;

[00215] (4) Not decoding any other portions in the picture.

[00216] In another variant of this embodiment the picture may have other portions but
5 other portions in the picture do not have a status A. In one example, the one portion with a master status is the top left subpicture in the picture. In another example the one portion with a master status is the subpicture with subpicture ID equal to value X.

[00217] In one embodiment, the portion of the picture is a subpicture and the existence of only one master subpicture is known to the decoder by default or without a dedicated signalling.

10 [00218] In another example of this embodiment, a syntax element in the bitstream equal to a first value indicates that there is only one portion in the picture with master status. This syntax element may be signalled in a header or a parameter set.

[00219] Embodiment 8: Master subpicture indication in the header itself

[00220] In this embodiment, an indication is signalled in the header of the current portion
15 which indicates that the current portion has status A or not. In one example, the portion is a slice and the indication that the current slice has status A or not is signalled in the slice header.

[00221] In a variant of this embodiment the portion that does not have status A carries only the indication that it does not have status A and no more coded sample data. In one example of this variant, a slice that is not a master slice carries only the indication that this is not a master
20 slice in the slice header, and no other slice data.

[00222] In another variant of this embodiment the indication specifies if the current portion of the picture has status A or status B (in this variant more than one status exists).

[00223] Embodiment 9: More than one level, subpicture hierarchy: status A, B, ...

[00224] In this embodiment, based on any of the previous embodiments, there is more
25 than one status in the picture, e.g., status A, status B and status C, each describing a specific status or feature.

[00225] In one embodiment, there are more than one possible levels of hierarchy for the status. For example, a portion with status A has a higher status than a portion with status B, and a

portion with status B has a higher status than a portion with status C. The different levels of status may here be seen as different priorities.

[00226] One example with multiple statuses includes means to prioritize what to send/pass on if the network conditions are limited or prioritize what to decode, e.g., if the decoding lags for some reason.

[00227] Another example use case of using multiple statuses could be when merging videos with different frame rates, such as a collage of surveillance videos where the cameras have different frame rates. Let's for instance say that a subpicture stream cam1 of 60 fps is merged side-by-side with a subpicture stream cam2 of 30 fps. Cam1 is the main feed and is marked with status A in the merged-side-by-side video, while every other frame of cam2 in the merged side-by-side video is marked as status B or status C. The frames with status B has coded sample data and the frames with status C has no coded sample data and are thus not decoded.

[00228] In one version of this embodiment the multiple statuses are grouped into representations (or subsets of the picture). For example, a first representation comprises portions only with status A, a second representation comprises portions with status A or B and a third representation comprises portions with status A or C.

[00229] In one example, imagine a picture in the video with texture and depth packed side-by-side, each with a region of interest (ROI) area as illustrated in FIG. 10.

[00230] One could then divide this picture into portions with four different statuses: (1) Status A: Texture ROI (portion 1); (2) Status B: Texture non-ROI (portion 2); (3) Status C: Depth ROI (portion 3); and (4) Status D: Depth non-ROI (portion 4). Representations for subsets of this picture could then be: Representation 1: status A (ROI texture); Representation 2: status A+B (Full texture); Representation 3: status A+C (ROI texture with depth); Representation 4: status C+D (Full depth); and Representation 5: status A+B+C+D (Full texture with depth).

[00231] In another version of this embodiment, a portion is allowed to have multiple statuses at the same time. This can be seen as an alternative to grouping statuses. With the same example picture as above the following statuses could be used: (1) Status A: Texture; (2) Status B: Depth; (3) Status C: ROI, where the portions would have the following statuses: Portion 1: Status A and C; Portion 2: Status A; Portion 3: Status B and C; and Portion 4: Status B.

[00232] The representations then comprise of the portions having the following combinations of statuses:

TABLE 11

Representation	Combinations of statuses
1. ROI texture	Both status A and C
2. Full texture	At least status A
3. ROI texture with depth	At least status C
4. Full depth	At least status B
5. Full picture with depth	Any combination of status A, B and C

[00233] Another example of having multiple statuses is the orientation example from embodiment 3. In that example the following two statuses could be defined: Status A: Vertical alignment and Status B: Horizontal alignment, where the subpicture 4 would have status A and B, subpictures 1 and 7 have status A only, subpictures 3 and 5 have status B only, and subpictures 0, 2, 6 and 8 would have no status.

[00234] Embodiment 10: Gating condition:

[00235] In this embodiment, the decoding of the portion of the picture with status equal to status X is conditional for instance by another syntax element in the bitstream. In one embodiment, a master subpicture status is indicated for at least one portion of the picture, and, if one or more criteria are satisfied (i.e., a criteria is satisfied), only subpictures in the picture with master subpicture status are decoded otherwise all subpictures are decoded. In one embodiment, the criteria is satisfied if a flag in a parameter set is set to a specific value (e.g., 1), the profile, tier, and/or level of the bitstream satisfies a condition (e.g., if bitstream has a level below a threshold, then the condition is satisfied), and/or the decoder has the resources to timely decode all portions. In another example, whether nor not the criteria is satisfied depends on one or more syntax elements in the bitstream.

[00236] In one example, when the encoder does not encode a first portion of the picture a status A is signalled for that portion of the picture in the bit stream and a flag in a parameter set is set equal to a first value. In this example the decoder decodes the flag being equal to a first value and based on that does not decode the first portion of the picture which has a status A. In other words, the decoder is configured such that, if the decoder determines that a particular portion has a status of A and the flag is equal to the first value, then the decoder does not decode

the particular portion, otherwise (i.e., status is not A or flag not equal to first value) the decoder decodes the particular portion.

[00237] Embodiment 11: A mechanism for parameter sets when there are master subpictures indicated: put a parameter set on hold and retrieve another parameter set, switch
5 between mode 1 and 2 each mode using a different parameter set, signalled action and default action where signalled action involves using one parameter set and the default action involves using another parameter set in the decoding of the picture.

[00238] In this embodiment there is a first and a second parameter set in the bitstream, and, if there are any picture portions with status A in the picture, then the first parameter set is
10 used in the decoding process of the picture, otherwise (i.e., there is no picture portion with status A in the picture) the second parameter set is used in the decoding process of the picture. The parameter set that is not used for decoding of the picture is denoted as an “on hold” parameter set, which may then be retrieved (used again for the decoding process of a new picture) based on the status of the picture portions in the new picture to be decoded.

[00239] For example, the following may apply for putting parameter set A on hold and
15 then retrieving it based on having or not having a master subpicture in the picture being decoded:

[00240] a) Parameter set A is valid for a particular picture (a picture having $POC = i$) which does not have any master subpictures,

[00241] b) Master subpicture(s) is indicated in another picture (e.g., a picture having
20 $POC = i + 1$), parameter set A is not valid for picture $POC = i + 1$, other means or inference mechanism or another parameter set B is being used for the picture $POC = i + 1$,

[00242] c) Master subpicture(s) is not indicated in picture $POC = i+2$, parameter set A is again valid for picture with $POC = i+2$ without being signaled again.

[00243] In one embodiment, two decoding parameter sets (DPS or decoding capability
25 information (DCI) NAL unit) are signalled in the bitstream which contain the decoding capability information, one to be used when there are master subpictures in the picture and another to be used when there are no master subpictures in the picture.

[00244] In one embodiment, the first and the second parameter sets have equal identifier values or parameter set IDs.

[00245] In one embodiment, the statuses of the picture portions are signalled in the PPS, and there are two SPSs with the same parameter set IDs and one is used in the decoding process if there is at least a portion with status A in the picture (statuses for the portions are signalled in the PPS for the picture) while the other SPS is used in the decoding process of the picture if there is no portion with status A in the picture (statuses for the portion are signalled in the PPS for the picture). In this example, the syntax elements in the higher parameter set, here SPS, that are needed for correct parsing of the lower parameter set, PPS here, are the same in both SPSs.

[00246] In another example, there are portions with status A that the decoder may refrain from decoding. If the decoder decides not to decode those portions, then the decoder uses one arrangement for the portions specified in a first SPS, but if the decoder decides to decode those portions, then a second portion arrangement specified in a second SPS is used. Both SPSs are in the bitstream with the same ID.

[00247] In yet another example, an indicator in the picture header specifies if the picture has a portion with status A or not. If the picture has at least one portion with status A, then a first PPS is used for decoding the picture, otherwise a second PPS is used for decoding this picture.

[00248] Embodiment 12: An input indicates which subpictures will be decoded.

[00249] In this embodiment, decoder 104 may decode only subpictures indicated as having status A in the bitstream in response to an input about device 103. The input may specify aspects related to decoding resources, screen size/resolution/orientation/ratio, memory or processing aspects, power or screen settings, etc. In one example, device 103, which hosting decoder 104, is running on power saving mode, and, as a result of receiving input indicating that the device is in the power saving mode, the decoder decodes only subpictures in the bitstream that are marked as master subpictures.

[00250] Embodiment 13: Decoder guaranties bit exact decode and output of only master portions.

[00251] In one embodiment, the decoder guarantees to decode only the picture portions that are indicated as having status A. In another embodiment, the decode for the subset of the picture portions that are indicated with status A is guaranteed to be bit exact, which means the sample values in the decoded portions are guaranteed to have the intended sample values by the encoder, but the decode of the picture portions which do not have status A may not be bit exact which means the sample values in the portions without status A may be different from the intended sample values by the encoder. In one embodiment, even if there is all the information for decoding a portion without status A in the bitstream, the decoder will not decode that portion.

[00252] In one example, only subpictures which are indicated as master subpictures are decoded by the decoder. The encoder might encode sample values for all the portions or only the portions with master status.

[00253] In a variant of this embodiment, the decoder may not decode picture portions which do not have status A. In another variant of this embodiment, the decoder shall not decode picture portions which do not have status A.

[00254] Embodiment 14: Encoder aspects

[00255] In this embodiment, the encoder analyzes different portions of the picture and indicates status values e.g. defines a priority level such as high, middle and low priority and indicates for each portion a status A if the portion has a high priority, a status B if the portion has a middle priority or a status C if the portion has a low priority. The decoder then decodes all portions or only a set of the portions based on their indicated status values.

[00256] In one embodiment one or more portions in the picture are indicated by the encoder to have a status value and other portions do not have any status value.

[00257] In another example there are at least two status values A and B and every portion in the picture is indicated by the encoder with status A or status B.

[00258] In one embodiment, the indication of the status value for a portion of the picture is based on the properties of the sample values in that picture or the relative variation of the sample values from one picture to the other, such as movement of the objects in the picture.

[00259] In one variant of this embodiment, the encoder takes in inputs that are then used by the encoder to partition the picture into portions and/or indicating the status for one or more

portions in the picture. In one example this input comes from another process that processes the picture content or from an external setting such as a configuration file or preference of a user such as a particular area in the picture is specified to the encoder as a region of interest (RoI) or having the more interesting content compared to other portions of the picture.

5 **[00260]** Embodiment 15: Preprocessing for setting portion partitions and status values

[00261] In this embodiment, a pre-processing step (e.g. a neural network (NN) module) processes the content and sets the partitions in the pictures (e.g. subpicture partitions) that partitions the picture into picture portions, and/or sets the status regarding each picture portion. This preprocessing may be part of the encoder or may be performed in a separate entity. The
10 picture partition and status values may be assigned once for the entire length of a video sequence or may be updated during a video sequence. The encoder may take in the partitioning and the status values from the pre-processing step as it is or may modify it before using in the encoding process.

[00262] Embodiment 16: Network aspects

15 **[00263]** In this embodiment, a network node 1104, illustrated in FIG. 11, receives portions of a picture (e.g., an encoded picture) from a sender 1102 (e.g., encoder 102) and decides which portions of the picture to forward to device 103 (labeled “receiver” in FIG. 11). The decision is based on the status of each portion. In one embodiment, when there is reduced network capacity, network node 104 can decide to only forward the portions having a high status (e.g., only those
20 portions having an indicator value that is greater than or equal to a threshold value). For instance, when there is reduced network capacity, network node 104 can decide to only forward the portions that are indicated as being a “master” subpicture or forward only portions that have a status that is included in a determined set of statuses (e.g., medium priority and high priority are forwarded, but not low priority).

25 **[00264]** In the example shown in FIG. 11, sender 1102 sends towards receiver 103 a bitstream comprising six subpictures of a picture (e.g., six VCL NAL units, each one corresponding to a different one of the six subpictures). Network node 1104, which is on the communication path between sender 1102 and receiver 103, detects that there is reduced network capacity to device 103, determines from the bitstream that there are two subpictures of a higher
30 status, e.g. master subpicture status, in the picture and determines to prioritize the two

subpictures with higher status and only pass through the NAL units necessary to decode these two subpictures to device 103.

[00265] In one version of this embodiment, the priority of the NAL units needed to decode the subpictures of higher status has further been marked with higher priority at the systems layer, the network layer, and/or the radio access network (RAN) layer. In one version, the data packets comprising these NAL units have additionally been marked using LFS (Low Latency Low Loss Scalable throughput). Any node in RAN serving an L4S capable data flow may set the Explicit Congestion Notification (ECN) bits in the IP header for the flow if congestion is experienced. In another version, the data packets comprising the NAL units (or portions thereof) have been marked with a priority indicator in a Packet Data Unit (PDU) set signaling in the RTP protocol, such that the NAL units needed to decode the two subpictures of higher status gets a higher priority than the NAL units for the other subpictures.

[00266] Embodiment 17: Output Portion Set (OPS)

[00267] In one embodiment, an active OPS specifies picture portions that shall be decoded. An active OPS may also specify picture portions that shall be decoded and output.

[00268] In one embodiment, the OPSs are signaled for an entire sequence of pictures, e.g., in a parameter set such as SPS, VPS, or in DCI. In another embodiment, the OPSs may be signaled more frequently, e.g., for a single picture or a set of pictures in a sequence. In this embodiment, the OPSs may be signaled in a header, such as the picture header or slice header or in a parameter set such as the PPS or APS. In one embodiment the OPSs are signaled in one or more SEI messages.

[00269] In one embodiment, all non-Video Coding Layer (non-VCL) data, e.g. parameter sets, picture header, picture level and higher level SEI messages, are decoded from the bitstream, but for VCL data, e.g. coded slice data, only the VCL data corresponding to a picture portion that is identified by an active OPS is decoded.

[00270] In one embodiment, an OPS comprises a set of portion IDs. An example is shown in TABLE 12 and example syntax of how this can be signalled is shown in TABLE 13.

TABLE 12

OPS ID	Portion IDs
--------	-------------

OPS0	1-5
OPS1	1-5, 9-12
OPS2	1-12
OPS3	1-8

TABLE 13

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
}	
}	
}	

[00271] Here “num_opss_minus1” plus 1 specifies the number of OPSs signalled in the output_portion_sets_structure(). This structure may for instance be signalled in a parameter set like the SPS or the VPS. “num_portions_minus1 [i]” plus 1 specifies the number of active portions in the i-th OPS. The active portions are the portions that should be decoded for the OPS. “portion_id [i][j]” specifies the portion ID of the j-th active portion in the i-th OPS.

[00272] In another embodiment (see example in Table 14), an indicator value, e.g., a flag e.g. portion_active_flag[i][j], is signalled in the OPS for each portion of a picture, wherein a first value of the flag, e.g. 1, specifies that the portion is to be decoded, and a second value, e.g. 0, specifies that the portion is not to be decoded.

TABLE 14

output_portion_sets_structure() {	
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	

num_portions_in_picture_minus1 [i]	u(v)
for(j = 0; j <= num_portions_in_picture_minus1[i]; j++) {	
portion_active_flag [i][j]	f(1)
}	
}	
}	

[00273] In one embodiment, each OPS of the one or more OPSs is used to specify which portions a conforming decoder shall decode in order to output a set of portions for a given operation point. When an OPS is activated (or selected), it is said to be an active OPS, and a conforming decoder shall decode the specified set of portions for the corresponding operation point. In one embodiment, a set of portions that a decoder shall be able to decode in order to produce a valid output is the same as the set of outputted portions. In another embodiment, a set of portions that a decoder shall be able to decode may include more portions than what is output, i.e., not all the decoded portions are output. In this embodiment, for each portion included in the OPS, it is also specified whether the portion is output or not. In one embodiment, it is explicitly signaled for each portion to be decoded whether it should also be outputted. As an example, see TABLE 15.

TABLE 15

	Portions to decode	Portions to output
OPS0	1-5	1-5
OPS1	1-5, 9-12	1-5, 9-12
OPS2	1-12	1-10
OPS3	1-8	1-6

[00274] An example syntax of how this can be signalled is shown in TABLE 16.

TABLE 16

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1 [i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	

portion_id [i][j]	u(v)
output_portion_flag [i][j]	u(1)
}	
}	
}	

[00275] The “**portion_id** [i][j]” as in the previous syntax specifies the portion ID of the j-th active portion in the i-th OPS and the “**output_portion_flag** [i][j]” specifies whether that portion should be output or not, for example only decoded but not output.

[00276] A decoder may perform all or a subset of the following steps for decoding a

5 picture comprising at least two picture portions:

1. Obtain a bitstream.
2. Obtain information of at least one output portion set (OPS), where an OPS specifies a set of portions of the picture. An OPS may specify what portions of a picture to decode for a given operation point. The OPS may further comprise information of which of the
10 decoded portions to also output from the decoder. A portion may be one or more of a portion coded independently from other portions, a subpicture, a slice, a tile, a CTU, a tile set or a tile group. The OPS information may be decoded from the bitstream, e.g. from a header such as a slice header or picture header or in a parameter set such as SPS, VPS, PPS, APS, in a DCI or in an SEI message. An OPS may specify an operating point
15 for the decoder.
3. Select a first OPS from the at least one OPSs. The indication of which OPS to use may be signalled in the bitstream or determined by other external means, e.g., by user input, hardcoded in the decoder or specifies by external specifications.
4. Determine from the first OPS that a first portion is to be decoded. This may be indicated
20 with the presence of an identifier for the first portion in the first OPS, or this may be indicated with an indicator for the first portion in the first OPS having a first value, e.g., 1.
5. Determine from the first OPS that a second portion is not to be decoded. This may be indicated with the lack of presence of an identifier for the second portion in the first OPS,
25 or this may be indicated with an indicator for the second portion in the first OPS having a second value, e.g., 0.

6. In response to determining from the first OPS that a first portion is to be decoded, decode the first portion.
7. In response to determining from the first OPS that a second portion is not to be decoded, do not decode the second portion.

5 **[00277]** Groupings

[00278] In one embodiment, the portions are grouped into one or more groups where the groups are signaled in the OPS. A group here can be seen as that the picture portions in a group has a particular status. An example of grouping portions is illustrated in FIG. 4B comprising four portions numbered from 0 to 3, where portion 0 and 1 belong to one group. The grouped portions
10 0 and 1 can be said to have a specific status. Portions 2 and 3 does not have the specific status.

[00279] The status (or group belonging) may be signaled for each picture portion, e.g., in the slice header or the NAL unit header. The status may also be signaled for each picture portion in a picture in a loop of a header, e.g., the picture header, or in a parameter set, such as the VPS, SPS, PPS or APS.

15 **[00280]** In one variant, each OPS is expressed as a status. That is, by specifying a particular status, one is specifying a set of picture portions (i.e., the picture portions having said particular status). In another variant, each picture portion set is expressed as having one or more statuses.

[00281] In one embodiment, the decoder decodes a first portion of the picture in response
20 to the first portion having a specific status (belonging to a certain group) and it does not decode a second portion of the picture in response to the second portion not having the specific status (not belonging to the certain group).

[00282] For example, a bitstream may contain OPS information where one OPS is associated with a particular status or group ID. The bitstream may additionally contain picture
25 partitioning information where pictures are partitioned into portions. This picture partitioning information may associate each partition with a status or group ID. Furthermore, the picture partitioning information may associate each partition with a partition ID. The bitstream additionally contains coded pictures where each coded picture contains coded portions. The coded portions contain portion information comprising a partition ID for each portion. Now, a

decoder that uses the one OPS when decoding the bitstream would for each coded portion check the partition ID and by using the status or group ID determine whether or not to decode the coded portion. If the status or group ID matches with the one OPS, then the decoder decodes the coded portion.

5 **[00283]** In one example of this embodiment, the group of portions that are decoded and output are based on the orientation of the display device, where the display device may for instance be a phone or TV set. In an example use case, as illustrated in FIG. 4A, a picture consists of 9 subpictures (numbered from 0 to 8 in FIG. 4A) and the picture is larger than what is shown on any screen. If shown vertically, only portions 1, 4, and 7 are displayed and if shown
10 horizontally only portions 3, 4, and 5 are displayed. In this example, portions 1, 4 and 7 could be indicated as having a specific first status (belonging to a first group) for a vertical display and portions 3, 4 and 5 could be indicated as having a specific second status (belonging to a second group) for a horizontal display. Portions 0, 2, 6 and 8 are never displayed and in one embodiment there is no VCL data for portions 0, 2, 6, 8. In another embodiment, portion 4 has a specific
15 status A and portions 1, 7 have a specific status B and portions 3, 5 have a specific status C. A decoder may then select or be instructed to decode and output the portion with status A and either the portion with status B or the portion with status C depending on, for example, orientation of the display.

[00284] This may be done without portions of the picture, but the advantage could be that
20 the portions that will not be displayed do not need to be decoded. It would often be ok to wait one second if the display orientation is flipped.

[00285] In another embodiment, there are more than one status in the picture, e.g., status A, status B and status C, each describing a specific status or feature, and the features may be combined.

25 **[00286]** One example with multiple statuses includes means to prioritize what to send/pass on if the network conditions are limited or prioritize what to decode, e.g., if the decoding lags for some reason.

[00287] Another example use case of using multiple statuses could be when merging of videos with different frame rates, such as a collage of surveillance videos where the cameras
30 have different frame rates. Let's for instance say that a subpicture stream cam1 of 60 fps is

merged side-by-side with a subpicture stream cam2 of 30 fps. Cam1 is the main feed and is marked with status A in the merged-side-by-side video, while every other frame of cam2 in the merged side-by-side video is marked as status B or status C. The frames with status B have coded sample data and the frames with status C have no coded sample data and are thus not decoded.

[00288] In one embodiment, the multiple statuses are grouped into representations (or subsets of the picture). For Example, a first representation comprises portions only with status A, a second representation comprises portions with status A or B and a third representation comprises portions with status A or C.

[00289] In another example, imagine a picture in the video with texture + depth packed side-by-side, each with a region of interest (ROI) area (see, e.g., FIG. 10). One could divide this picture into portions with four different statuses:

Status A: Texture ROI (portion 1)
 Status B: Texture non-ROI (portion 2)
 Status C: Depth ROI (portion 3)
 Status D: Depth non-ROI (portion 4)

[00290] Representations for subsets of this picture could then be signaled in OPSs:

OPS 1: status A (ROI texture)
 OPS 2: status A+B (Full texture)
 OPS 3: status A+C (ROI texture with depth)
 OPS 4: status C+D (Full depth)
 OPS 5: status A+B+C+D (Full texture with depth)

[00291] The set of picture portion sets, or OPS, may for instance be called an “output status set (OSS)” in this embodiment. In one variant, all non-Video Coding Layer (non-VCL) data, e.g. parameter sets, picture header, picture level and higher level SEI messages, are decoded, but for VCL data, e.g. coded slice data, only those with a status included in the OPS are decoded, in this case the OPS consists of one status.

[00292] In yet another example, consider a picture comprising 12 portions, where portions 1-5 have status A (i.e., belong to group A), portions 6-8 have status B (i.e., belong to group B), and portions 9-12 have status C (i.e., belong to group C). In this example, the bitstream defines four OPSs with different combination of statuses according to TABLE 17 below. An example

syntax of how this can be signalled is shown in TABLE 18. The statuses in the OPS specifies for which portions VCL data should be decoded. In this example, all portions have one status. In one variant of the embodiment, a portion may have more than one status (i.e., a portion may belong to more than one group).

5

TABLE 17

	Status (group)
OPS0	A
OPS1	A and C
OPS2	A, B and C
OPS4	A and B

TABLE 18

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_groups_in_ops_minus1[i]	u(v)
for(j = 0; j <= num_groups_in_ops_minus1[i]; j++) {	
group_id [i][j]	u(v)
}	
}	
}	

[00293] Here “num_opss_minus1” plus 1 specifies the number of OPSs signalled in the output_portion_sets_structure(). “num_groups_in_ops_minus1 [i]” plus 1 specifies the number of groups (with a specific status) there are in the i-th OPS. The groups are the groups of portions that should be decoded for the OPS. “group_id [i][j]” specifies the group ID (status ID) of the j-th group in the i-th OPS.

[00294] In another embodiment, both the status and the indicator value of the first embodiment is signalled as shown in Table 19.

15

TABLE 19

	Status	Portions
OPS0	A	1-5
OPS1	A and C	1-5, 9-12
OPS2	A, B and C	1-12
OPS3	A and B	1-8

[00295] In the example where OPSs uses statuses, the output may be specified as shown in Table 20 below, and example syntax of how this can be signalled is shown in Table 21. Here OPS1 specifies that all portions having status A and C are decoded but only portions having status A are output.

TABLE 20

	Status	Output
OPS0	A	Status A
OPS1	A and C	Status A
OPS2	A, B and C	Status A,B,C

TABLE 21

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_groups_in_ops_minus1[i]	u(v)
for(j = 0; j <= num_groups_in_ops_minus1[i]; j++) {	
group_id [i][j]	u(v)
output_group_flag [i][j]	u(1)
}	
}	
}	

[00296] The “group_id [i][j]” specifies the group ID of the j-th active group (the group of portions that should be decoded) in the i-th OPS and the “output_group_flag [i][j]” specifies whether the portions of that group should be output or not, for example only decoded but not output.

[00297] In another example, based on the example in FIG. 4A, one OPS is specified for the vertical orientation and another OPS is specified for the horizontal orientation. Yet another OPS could comprise all subpictures.

[00298] The OPSs may be signalled in the bitstream or conveyed by external means. In one embodiment of the embodiment, the OPSs are signalled in the VPS as for the OLSs. In other embodiments of the embodiment, the OPSs are signalled in another parameter set, such as SPS or PPS, or in a header, such as a picture header or a slice header, or in an SEI message.

[00299] Indication of which OPS to apply

[00300] Similar to the OPI NAL unit type in VVC, the bitstream may contain an indication of which of the OPS the decoder should use (i.e., which ones are the “active” OPSs). This indication may be referred to as the target OPS indicator and may be said to indicate the OPS in scope. This indication may as in VVC be signalled in a separate NAL unit with its own NAL unit type. The indication may also be signalled in a parameter set, such as VPS, SPS or PPS, in a header, such as a picture header or a slice header or be conveyed by external means, e.g. set by user input, be hard coded in the decoder, be defined by a third party specification, etc.

[00301] In one embodiment, an active OPS is used to determine or specify the operating point at which a decoder shall operate (possibly in combination with other input, e.g. what layers and temporal sublayers to decode).

[00302] In one embodiment, an active OPS is used to determine or specify an operation point for a conforming decoder including which picture portions that shall be decoded and which picture portions that shall be output for the given operation point.

[00303] In one embodiment, the OPS is combined with the OLS such that the portions and layers that may be used for decoding and output are specified in the same structure. Similarly, the indication of which OPS to use could be signalled in the OPI together with the indication of which OLS to use.

[00304] Output Layout for an OPS

[00305] In this embodiment a layout is signalled for the OPS specifying the positioning of the decoded portions in the output picture. In a preferred variant of this embodiment the specified positioning of the decoded portions in the output picture is a post processing step which does not

affect what will be used as a reference picture for decoding other pictures later. An example syntax of how this can be signalled is shown in Table 22 below:

TABLE 22

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
portion_output_ctu_top_left_x [i][j]	u(v)
portion_output_ctu_top_left_y [i][j]	u(v)
}	
}	
}	

[00306] In the above example, portion_id[i][j] specifies which portions are in the i-th OPS and syntax elements portion_output_ctu_top_left_x[i][j] and portion_output_ctu_top_left_y[i][j] specify the location of the top left corner of the j-th portion of the i-th OPS in the output picture.

[00307] In a variant of this embodiment using the new layout for positioning of the decoded portions in the output picture is constrained. In one example, a layout1 is signalled for OPS1 and a layout2 is signalled for OPS2, but the decoder is conditioned to use the layout1 for the decoded portions in OPS1 only if the portions in OPS2 are not decoded by the decoder. In this example for instance the signalled layout1 and layout2 will not be used if portions in both OPS1 and OPS2 are decoded.

[00308] In another variant of this embodiment, the signalling of the locations for the portions of an OPS is conditioned to only those portions that are being output. In one example shown in the table below, the new location of a portion in an OPS is signalled only if the portion is being output. In this example, if value of the syntax element output_portion_flag [i][j] is equal to 1 then the portion is going to be output and so the location of this portion in the picture to be output is signalled with portion_output_ctu_top_left_x[i][j] and portion_output_ctu_top_left_y[i][j], and if syntax element output_portion_flag [i][j] is equal

to 0 then `portion_output_ctu_top_left_x[ i ][ j ]` and `portion_output_ctu_top_left_y[ i ][ j ]` will not be signalled .

Table 23

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>num_portions_minus1[i]</code>	<code>u(v)</code>
<code>for(j = 0; j <= num_portions_minus1[i]; j++) {</code>	
<code>portion_id [i][j]</code>	<code>u(v)</code>
<code>output_portion_flag [i][j]</code>	<code>u(1)</code>
<code>if output_portion_flag [i][j] {</code>	
<code>portion_output_ctu_top_left_x [i][j]</code>	<code>u(v)</code>
<code>portion_output_ctu_top_left_y [i][j]</code>	<code>u(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	
<code>}</code>	

[00309] In another embodiment one OPS, e.g. OPS2 is a subset of another OPS, e.g.

- 5 OPS1. In this embodiment the OPS2 may be constrained to only contain portions that are also contained in OPS1 (ok to not have all) and wherein the portions in OPS2 have the same layout as the corresponding portions of OPS1 in the output picture. In one example, shown in the table below, layout of the portions is signalled only for the superset OPS (OPS1 in the above example) but applied to both the superset and subset OPSs (OPS1 and OPS2 in the above example).

TABLE 24

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>subset_of_another_ops[i]</code>	<code>u(1)</code>
<code>if subset_of_another_ops[i]</code>	
<code>superset_ops_id[i]</code>	
<code>num_portions_minus1[i]</code>	<code>u(v)</code>

for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
if (!subset_of_another_ops[i]){	
portion_output_ctu_top_left_x [i][j]	u(v)
portion_output_ctu_top_left_y [i][j]	u(v)
}	
}	
}	
}	

[00310] In the above example, subset_of_another_ops[i] equal to 1 specifies that the (i+1)-th OPS is a subset of another OPS specified by superset_ops_id[i], and a new layout is not signalled for the (i+1)-th OPS, whereas subset_of_another_ops[i] equal to 0 specifies that the (i+1)-th OPS is not a subset of another OPS and a new layout is signalled for the (i+1)-th OPS using syntax elements portion_output_ctu_top_left_x[i][j] and portion_output_ctu_top_left_y[i][j].

[00311] Scaling Portions Specified by an OPS

[00312] In this embodiment, it is also signalled in the bitstream how the portions specified in an OPS are scaled after decoding by the decoder to be output as the output picture.

10 [00313] One example in the following table shows how this can be signalled.

Table 25

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1 [i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id [i][j]	u(v)
portion_output_ctu_top_left_x [i][j]	u(v)
portion_output_ctu_top_left_y [i][j]	u(v)
portion_output_scale_x [i][j]	u(v)
portion_output_scale_y [i][j]	u(v)
}	
}	

}	
---	--

[00314] In the above table, syntax elements `portion_output_scale_x[ i ][ j ]` and `portion_output_scale_y[ i ][ j ]` specify the scaling factor corresponding to the x and y dimensions of the j-th portion in the i-th OPS.

[00315] In a variant of this embodiment, signalling of the scale factors are conditioned. In one example the scale factor for a portion is conditioned only if the portion is going to be decoded and output.

[00316] Each Portion Belongs to Exactly one OPS

[00317] In this embodiment each portion belongs to exactly one OPS. In one example, the division of the portions into OPSs is signalled in a loop over the number of portions signalling an OPS id for each portion, as shown below:

Table 26 (list of OPS ids)

<code>seq_parameter_set_rbsp() {</code>	Descriptor
...	
num_portions_minus1	u(8)
<code>for(j = 0; j <= num_portions_minus1; j++) {</code>	
ops_id[j]	u(v)
<code>}</code>	
...	
<code>}</code>	

[00318] `ops_id[ j ]` is an OPS identifier for the j-th partition.

[00319] Sample values for portions which are not in an active OPS

[00320] This embodiment may be combined with any of the previous embodiments and in addition, for an OPS, it is signalled what sample value to output for the portions which are not decoded (the portions which are not in the set of active portions of the OPS).

[00321] In one example a portion P of the picture is not decoded at the operation point of OPS1, and a one-byte syntax element `filler_val` is signalled that specifies the sample value to fill the not decoded portions with, when the output picture includes not decoded portions such as P.

In this example `filler_val = 0` specifies that the sample values corresponding to the portion P in

the output picture are filled with white and `filler_val = 255` specifies that the sample values corresponding to the not decoded portion P in the output picture are filled with black and every other value of `filler_val` specifies a gray scale. This example is shown in Table 27.

Table 27

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>filler_val</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>num_portions_minus1[i]</code>	<code>u(v)</code>
<code>for(j = 0; j <= num_portions_minus1[i]; j++) {</code>	
<code>portion_id[i][j]</code>	<code>u(v)</code>
<code>}</code>	
<code>}</code>	
<code>}</code>	

5 [00322] In the above example one `filler_val` syntax element is specified once for all OPSs.

[00323] In a variant of this embodiment the specification of the output sample values for the portions which are not decoded is per OPS. Following is an example where `filler_val[ i ]` specifies the value to be used as the filler of not-decoded portions when having the *i*-th OPS as the active operation point:

10

Table 28

<code>output_portion_sets_structure() {</code>	Descriptor
<code>...</code>	
<code>num_opss_minus1</code>	<code>u(8)</code>
<code>for(i = 0; i <= num_opss_minus1; i++) {</code>	
<code>filler_val[i]</code>	<code>u(8)</code>
<code>num_portions_minus1[i]</code>	<code>u(v)</code>
<code>for(j = 0; j <= num_portions_minus1[i]; j++) {</code>	
<code>portion_id[i][j]</code>	<code>u(v)</code>
<code>}</code>	

}	
}	

[00324] In another variant of this embodiment the not-decoded sample values in the output picture are copied from another picture or decoded portions of the current picture. In one example of this variant shown in the table below, a syntax element `copy_interval = X` specifies that for all defined active operation points `OPS[ i ]` the not-decoded sample values in the output picture are copied from the collocated sample values of the $(X+1)$ -th former output picture in the output order, where `copy_interval = 0` specifies to copy from the previous output picture.

Table 29

output_portion_sets_structure() {	Descriptor
...	
num_opss_minus1	u(8)
copy_interval	u(v)
for(i = 0; i <= num_opss_minus1; i++) {	
num_portions_minus1[i]	u(v)
for(j = 0; j <= num_portions_minus1[i]; j++) {	
portion_id[i] [j]	u(v)
}	
}	
}	

[00325] OPS as a Post Processing SEI

[00326] In this embodiment OPS is a pure post-processing tool, so it does not affect the decoding of the pictures but signals processes to be applied to the decoded picture in order to create output pictures. In one example of this embodiment the OPS information is signalled in an SEI message and includes the information (such as a layout information) of what portions from the decoded picture to put together to create an output picture. In one example of this embodiment the OPS information includes POC information specifying portions of which decoded pictures to be used to create an output picture. The following table shows one example of an OPS SEI message:

Table 30

output_portion_set(payloadSize) {	Descriptor
...	
ops_id	ue(v)
ops_num_portions_minus1	u(v)
for(i = 0; i <= num_portions_minus1; i++) {	
ops_portion_id [i]	u(v)
ops_portion_output_ctu_top_left_x[i]	u(v)
ops_portion_output_ctu_top_left_y[i]	u(v)
}	
...	
}	

[00327] FIG. 12 is a flowchart illustrating a process 1200 for decoding a picture from a bitstream, the picture comprising a first portion and a second portion. Process 1200 may begin in step s1202. Step s1202 comprises obtaining, from the bitstream, first status information (e.g., a first indicator value) corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion. Step s1204 comprises determining whether or not to decode the first portion of the picture based on the first status information (e.g., decoding the first portion of the picture in response to determining that the first status information indicates that the first portion has a first status and refraining from decoding the first portion of the picture in response to determining that the first status information indicates that the first portion does not have the first status and zero or more other criteria are met).

[00328] FIG. 13 is a flowchart illustrating a process 1300 for producing a bitstream comprising a picture (i.e., an encoded picture), the picture comprising a first portion and a second portion. Process 1300 may begin in step s1302. Step s1302 comprises determining whether the first portion is required to be decoded by a decoder. Step s1304 comprises, as a result of determining that the first portion is required to be decoded by a decoder, producing encoded status information for the first portion and including the encoded status information in the bitstream, wherein the encoded status information for the first portion does not correspond to the second portion, and the encoded status information is for informing the decoder that the decoder should not refrain from decoding the first portion of the picture.

[00329] FIG. 14 is a flowchart illustrating a process 1400 performed by a network node (e.g., network node 1104). Process 1400 may begin in step s1402. Step s1402 comprises receiving i) a first unit (e.g., VCL NAL unit) comprising encoded picture sample values corresponding to a first portion of a picture and ii) first status information (e.g., encoded first status information) corresponding to the first portion of the picture, wherein the first status information does not correspond to a second portion of the picture. Step s1403 (optional) comprises receiving a second unit (e.g., VCL NAL unit) comprising encoded picture sample values corresponding to the second portion of the picture. Step s1404 comprises, based on the first status information, deciding whether or not to forward the first unit to a receiving device. In one embodiment, if the first status information indicates a first status the first unit, then the first unit will not be forwarded. Because the first status information does not correspond to the second portion of the picture, the decision of whether or not to forward the second unit corresponding to the second portion may be independent of the first status information. In one embodiment, the first status information is included in the first unit.

[00330] FIG. 16 is a block diagram of an apparatus 1600 for implementing encoder 102, decoder 104, and/or network node 1104, according to some embodiments. When apparatus 1600 implements encoder 102, apparatus 1600 may be referred to as an encoder apparatus, when apparatus 1600 implements decoder 104, apparatus 1600 may be referred to as a decoder apparatus, and when apparatus 1600 implements network node 1104, apparatus 1600 may be referred to as a network node. As shown in FIG. 16, apparatus 1600 may comprise: processing circuitry (PC) 1602, which may include one or more processors (P) 1655 (e.g., one or more general purpose microprocessors and/or one or more other processors, such as an application specific integrated circuit (ASIC), field-programmable gate arrays (FPGAs), and the like), which processors may be co-located in a single housing or in a single data center or may be geographically distributed (i.e., encoder apparatus 1600 may be a distributed computing apparatus); at least one network interface 1648 (e.g., a physical interface or air interface) comprising a transmitter (Tx) 1645 and a receiver (Rx) 1647 for enabling apparatus 1600 to transmit data to and receive data from other nodes connected to a network 110 (e.g., an Internet Protocol (IP) network) to which network interface 1648 is connected (physically or wirelessly) (e.g., network interface 1648 may be coupled to an antenna arrangement comprising one or more antennas for enabling encoder apparatus 1600 to wirelessly transmit/receive data); and a storage

unit (a.k.a., “data storage system”) 1608, which may include one or more non-volatile storage devices and/or one or more volatile storage devices. In embodiments where PC 1602 includes a programmable processor, a computer readable storage medium (CRSM) 1642 may be provided. CRSM 1642 may store a computer program (CP) 1643 comprising computer readable

- 5 instructions (CRI) 1644. CRSM 1642 may be a non-transitory computer readable medium, such as, magnetic media (e.g., a hard disk), optical media, memory devices (e.g., random access memory, flash memory), and the like. In some embodiments, the CRI 1644 of computer program 1643 is configured such that when executed by PC 1602, the CRI causes encoder apparatus 1600 to perform steps described herein (e.g., steps described herein with reference to the flow charts).
- 10 In other embodiments, encoder apparatus 1600 may be configured to perform steps described herein without the need for code. That is, for example, PC 1602 may consist merely of one or more ASICs. Hence, the features of the embodiments described herein may be implemented in hardware and/or software.

[00331] Summary of Various Embodiments

- 15 **[00332]** A1. A method 1200 for decoding a picture from a bitstream, the picture comprising a first portion and a second portion, the method comprising: obtaining, from the bitstream, first status information (e.g., a first indicator value) corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion; and determining whether or not to decode the first portion of the picture based on the first status
- 20 information.

[00333] A2. The method of embodiment A1, wherein the method further comprises obtaining second status information corresponding to the second portion and not corresponding to the first portion; and determining whether or not to decode the second portion of the picture based on the second status information.

- 25 **[00334]** A3. The method of embodiment A1 or A2, wherein the first portion of the picture is a subpicture, a slice, a tile, a CTU, a tile set, or a tile group.

[00335] A4. The method of any previous embodiment, wherein the first portion is independent from other portions in the picture (e.g., is independent from the second portion).

[00336] A5. The method of any previous embodiment, wherein the first portion comprises a set of one or more coded sample values, and the method further comprises, in response to determining that the first status information indicates that the first portion has a first status, decoding the set of coded sample values to produce a set of decoded sample values.

5 **[00337]** A6. The method of embodiment A5, wherein the method further comprises using the set of decoded sample values to generate a set of corresponding reconstructed sample values.

[00338] A7. The method of any previous embodiment, wherein the first status information is a status indicator value, and the method further comprises: determining a first status for the first portion based on the status indicator value; and decoding the first portion in response to the
10 first status being equal to a status A or refraining from decoding the first portion in response to the first status not being equal to status A.

[00339] A8. The method of any of the previous embodiment, wherein the first status information indicates that the first portion is master portion (e.g., the status information indicates that no other portion of the picture has a higher status (importance, priority, etc.) than the first
15 portion).

[00340] A9. The method of any previous embodiment, wherein the first status information is decoded from a parameter set, a header, or an SEI message.

[00341] A10. The method of any of the previous embodiment, wherein the first portion comprises two or more CTUs.

20 **[00342]** A11. The method of any previous embodiment, wherein the method further comprises: obtaining, from the bitstream, second status information (e.g., a second status indicator value) corresponding to the second portion of the picture (the second status indicator value may be equal to or not equal to the first status indicator value); and determining whether or not to decode the second portion of the picture based on the second status information.

25 **[00343]** A12. The method of any previous embodiment, wherein the method comprises determining not to decode the first portion, and the method further comprises generating reconstructed sample values corresponding to the first portion using: constant sample values, default sample values, previously reconstructed sample values, collocated sample values in one

or more previously decoded pictures, other sample values in one or more previously decoded pictures, other sample values in the picture, and/or a pre-determined picture or part of a picture.

[00344] A12b. The method of claim A12, wherein the bitstream comprises information for indicating that reconstructed sample values corresponding to the first portion should be generated using: constant sample values, default sample values, previously reconstructed sample values, collocated sample values in one or more previously decoded pictures, other sample values in one or more previously decoded pictures, other sample values in the picture, interpolated sample values, extrapolated sample values, and/or a pre-determined picture or part of a picture.

[00345] A13. The method of any previous embodiment, wherein the method comprises determining not to decode the first portion, and the method further comprises generating reconstructed sample values corresponding to the first portion using generated, copied, interpolated, and/or extrapolated sample values.

[00346] A14. The method of any previous embodiment, wherein the method comprises determining not to decode the first portion, and the method further comprises: rearranging at least a second portion of the picture (e.g., such as subpicture, slice, tile in the decoded picture); rescaling at least the second portion of the picture (scales the second portion to cover an area of picture corresponding to the first portion); and/or changing the output resolution of the decoded picture.

[00347] A15. The method of any previous embodiment, wherein the first status information is decoded from the bitstream in response to a third indicator being decoded from the bitstream and the value of the third indicator being equal to a third value (e.g., a third value indicating that the first value is encoded in the bitstream).

[00348] A16. The method of embodiment A15, wherein the third indicator is decoded from a parameter set (e.g., SPS or PPS) or a header (e.g., slice header or picture header).

[00349] A17. The method of any previous embodiment, wherein the first status information indicates that the first portion has a particular status (e.g., master status), and the method further comprises, as a result of determining that the first status information indicates that the first portion has a particular status, determining that no other portion of the picture has said particular status.

[00350] A18. The method of any previous embodiment, wherein obtaining the first status information from the bitstream comprises decoding the first status information from a first index syntax element in the bitstream.

[00351] A19. The method of embodiment A18, wherein the first status information indicates that the first portion has a particular status (e.g., master status), and the method further comprises: decoding the first portion of the picture in response to determining that the first portion has said particular status; and refraining from decoding all other portions of the picture.

[00352] A20. The method of any previous embodiment, wherein the first portion is a slice of the picture, and the status indication is signalled in a slice header for the slice.

[00353] A21. The method of embodiment A20, wherein the slice header is contained in a VCL NAL unit, and the VCL NAL unit does not contain any coded sample values (i.e., does not contain any data that represents picture sample values).

[00354] A22. The method of any previous embodiment, wherein the method further comprises obtaining, from the bitstream, second status information corresponding to the second portion of the picture, the first status information indicates that the first portion has a first particular status, status A (e.g., master status), the second status information indicates that the second portion has a second particular status, status B (e.g., non-master status), and status A has a higher priority than status B.

[00355] A23. The method of embodiment A22, wherein status A having a higher priority than status B indicates that the first portion with status A is not decoded only if the second portion with status B is not decoded.

[00356] A24. The method of any previous embodiment, wherein the method further comprises decoding, from the bitstream, a gating value, the determination as to whether or not to decode the first portion is based not only on the first status information, but also the gating value.

[00357] A25. The method of embodiment A24, wherein if the first status information has a first particular value, then the first portion is decoded only if the gating value satisfies a condition.

[00358] A26. The method of embodiments A24 and A25, wherein the gating value specifies that only picture portions having a particular status should be decoded.

[00359] A27. The method of any previous embodiment, wherein the bitstream comprises a set of parameter sets, the set of parameter sets comprising a first parameter set and a second parameters set, the method further comprises selecting, based on the first status information, a parameter set from the set of parameter sets, and the method further comprises using the selected
5 parameter in a decoding process for decoding the picture.

[00360] A28. The method of embodiment A27, wherein the bitstream comprises a picture parameter set (PPS) for the picture, the first parameter set is a first sequence parameter set (SPS) having a first SPS identifier (ID), the second parameter set is a second SPS having the first SPS ID, and obtaining the first status information from the bitstream comprises decoding the first
10 status information from a syntax element in the PPS.

[00361] A29. The method of embodiment A28, wherein the method comprises selecting the first SPS as a result of determining that the first status information indicates that the first portion has a particular status or selecting the first SPS as a result of determining that no portion of the picture has the particular status.

15 [00362] A30. The method of any previous embodiment, wherein the method is performed by a decoder of a device, and the determination as to whether or not to decode the first portion is based not only on the first status information, but also information regarding the device.

[00363] A31. The method of embodiment A30, wherein the information regarding the device comprises: screen size information, screen resolution information, screen orientation
20 information, screen ratio information, memory load information, processing load information, battery level information, and/or power consumption information.

[00364] A32. The method of any previous embodiment, wherein only picture portions with a particular status are guaranteed a bit exact decode.

[00365] A33. The method of any previous embodiment, wherein the first status
25 information indicates an output portion set, OPS, to which the first portion belongs.

[00366] B1. A method 1300 (see FIG. 13) for producing a bitstream comprising a picture (i.e., an encoded picture), the picture comprising a first portion and a second portion, the method comprising: determining whether the first portion is required to be decoded by a decoder; and as a result of determining that the first portion is required to be decoded by a decoder, producing

encoded status information for the first portion and including the encoded status information in the bitstream, wherein the encoded status information for the first portion does not correspond to the second portion of the picture, and the encoded status information is for informing the decoder that the decoder should not refrain from decoding the first portion of the picture.

5 **[00367]** B2. The method of embodiment B1, wherein determining whether the first portion is required to be decoded by a decoder comprises receiving input from a pre-processor, wherein the input is used to determine whether the first portion is required to be decoded by a decoder.

[00368] B3. The method of embodiment B2, wherein the input comprises picture partitioning information that identifies the portions of the picture and, for each identified portion,
10 a status for the portion.

[00369] B4. The method of embodiment B3, wherein the picture partitioning information identifies a region of interest (RoI) within the picture, and the status for the RoI is a first status (e.g., master status).

[00370] C1. A method 1400 (see FIG. 14) performed by a network node, the method
15 comprising: receiving a first unit (e.g., VCL NAL unit) comprising encoded picture sample values corresponding to a first portion of a picture and first status information (e.g., encoded first status information) corresponding to the first portion of the picture; and based on the first status information, deciding whether or not to forward the first unit to a receiving device.

[00371] C2. The method of embodiment C1, wherein the first status information
20 comprises a first status indicator value, and deciding whether or not to forward the first unit to a receiving device based on the status information comprises determining whether the first status indicator value satisfies a criterion (e.g., determining whether the first status indicator value is less than a threshold or is greater than a threshold)

[00372] C3. The method of embodiment C2, wherein the method further comprises:
25 receiving second status information corresponding to the second portion of the picture; and based on the second status information, deciding whether or not to forward the second unit to the receiving device, wherein the second status information comprises a second status indicator value that is different than the first status indicator value, and deciding whether or not to forward

the second unit to a receiving device based on the second status information comprises determining whether the second status indicator value satisfies said criterion.

[00373] D1. A computer program comprising instructions which when executed by processing circuitry of an apparatus causes the apparatus to perform the method of any one of the
5 above embodiments.

[00374] D2. A carrier containing the computer program of embodiment D1, wherein the carrier is one of an electronic signal, an optical signal, a radio signal, and a computer readable storage medium.

[00375] E1. An apparatus configured to perform any of the method of any one of the
10 above method embodiments.

[00376] While the terminology in this disclosure is described in terms of VVC, the embodiments of this disclosure also apply to any existing or future codec, which may use a different, but equivalent terminology.

[00377] Also, while various embodiments are described herein, it should be understood
15 that they have been presented by way of example only, and not limitation. Thus, the breadth and scope of this disclosure should not be limited by any of the above-described exemplary embodiments. Moreover, any combination of the above-described elements in all possible variations thereof is encompassed by the disclosure unless otherwise indicated herein or otherwise clearly contradicted by context. As used herein “a” means “at least one” or “one or
20 more.”

[00378] Additionally, while the processes described above and illustrated in the drawings are shown as a sequence of steps, this was done solely for the sake of illustration. Accordingly, it is contemplated that some steps may be added, some steps may be omitted, the order of the steps may be re-arranged, and some steps may be performed in parallel.

CLAIMS

1. A method (1200) for decoding a picture from a bitstream, the picture comprising a first portion and a second portion, the method comprising:

5 obtaining (s1202), from the bitstream, first status information corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion; and

determining (s1204) whether or not to decode the first portion of the picture based on the first status information.

10 2. The method of claim 1, wherein the method further comprises obtaining second status information corresponding to the second portion and not corresponding to the first portion; and

determining whether or not to decode the second portion of the picture based on the second status information.

3. The method of claim 1 or 2, wherein the first portion of the picture is a subpicture, a slice, a tile, a CTU, a tile set, or a tile group.

20 4. The method of any previous claim, wherein the first portion is independent from other portions in the picture.

5. The method of any previous claim, wherein the first portion comprises a set of one or more coded sample values, and
25 the method further comprises, in response to determining that the first status information indicates that the first portion has a first status, decoding the set of coded sample values to produce a set of decoded sample values.

6. The method of claim 5, wherein the method further comprises using the set of decoded
30 sample values to generate a set of corresponding reconstructed sample values.

7. The method of any previous claim, wherein
the first status information is a status indicator value, and
the method further comprises:

determining a first status for the first portion based on the status indicator value; and

5 decoding the first portion in response to the first status being equal to a status A or
refraining from decoding the first portion in response to the first status not being equal to status
A.

8. The method of any of the previous claim, wherein the first status information indicates
10 that the first portion is master portion.

9. The method of any previous claim, wherein the first status information is decoded from
a parameter set, a header, or an SEI message.

10. The method of any of the previous claim, wherein the first portion comprises two or
15 more CTUs.

11. The method of any previous claim, wherein the method further comprises:
obtaining, from the bitstream, second status information corresponding to the second
20 portion of the picture; and
determining whether or not to decode the second portion of the picture based on the
second status information.

12. The method of any previous claim, wherein
25 the method comprises determining not to decode the first portion,
the method further comprises generating reconstructed sample values corresponding to
the first portion using: constant sample values, default sample values, previously reconstructed
sample values, collocated sample values in one or more previously decoded pictures, other
sample values in one or more previously decoded pictures, other sample values in the picture,
30 and/or a pre-determined picture or part of a picture.

13. The method of any previous claim, wherein
the method comprises determining not to decode the first portion, and
the method further comprises generating reconstructed sample values corresponding to
the first portion using generated, copied, interpolated, and/or extrapolated sample values.

5

14. The method of any previous claim, wherein
the method comprises determining not to decode the first portion, and
the method further comprises:

10

rearranging at least a second portion of the picture;
rescaling at least the second portion of the picture; and/or
changing the output resolution of the decoded picture.

15

15. The method of any previous claim, wherein the first status information is decoded
from the bitstream in response to a third indicator being decoded from the bitstream and the
value of the third indicator being equal to a third value.

16. The method of claim 15, wherein the third indicator is decoded from a parameter set
or a header.

20

17. The method of any previous claim, wherein
the first status information indicates that the first portion has a particular status, and
the method further comprises, as a result of determining that the first status information
indicates that the first portion has a particular status, determining that no other portion of the
picture has said particular status.

25

18. The method of any previous claim, wherein
obtaining the first status information from the bitstream comprises decoding the first
status information from a first index syntax element in the bitstream.

30

19. The method of claim 18, wherein
the first status information indicates that the first portion has a particular status, and

the method further comprises:

decoding the first portion of the picture in response to determining that the first portion has said particular status; and
refraining from decoding all other portions of the picture.

5

20. The method of any previous claim, wherein
the first portion is a slice of the picture, and
the status indication is signalled in a slice header for the slice.

10

21. The method of claim 20, wherein
the slice header is contained in a VCL NAL unit, and
the VCL NAL unit does not contain any coded sample values.

15

22. The method of any previous claim, wherein
the method further comprises obtaining, from the bitstream, second status information corresponding to the second portion of the picture,
the first status information indicates that the first portion has a first particular status, status A,
the second status information indicates that the second portion has a second particular status, status B,
status A has a higher priority than status B.

20

25

23. The method of any previous claim, wherein
the method further comprises decoding, from the bitstream, a gating value, and
the determination as to whether or not to decode the first portion is based not only on the first status information, but also the gating value.

30

24. The method of any previous claim, wherein
the bitstream comprises a set of parameter sets, the set of parameter sets comprising a first parameter set and a second parameters set,

the method further comprises selecting, based on the first status information, a parameter set from the set of parameter sets, and

the method further comprises using the selected parameter in a decoding process for decoding the picture.

5

25. The method of claim 24, wherein
the bitstream comprises a picture parameter set (PPS) for the picture,
the first parameter set is a first sequence parameter set (SPS) having a first SPS identifier (ID),

10

the second parameter set is a second SPS having the first SPS ID, and
obtaining the first status information from the bitstream comprises decoding the first status information from a syntax element in the PPS.

15

26. The method of any previous claim, wherein
the method is performed by a decoder of a device, and
the determination as to whether or not to decode the first portion is based not only on the first status information, but also information regarding the device.

20

27. The method of any previous claim, wherein only picture portions with a particular status are guaranteed a bit exact decode.

28. The method of any previous claim, wherein the first status information indicates an output portion set, OPS, to which the first portion belongs.

25

29. A method (1300) for producing a bitstream comprising a picture, the picture comprising a first portion and a second portion, the method comprising:

determining (s1302) whether the first portion is required to be decoded by a decoder; and
as a result of determining that the first portion is required to be decoded by a decoder, producing (s1304) encoded status information for the first portion and including the encoded status information in the bitstream, wherein

30

the encoded status information for the first portion does not correspond to the second portion of the picture, and

the encoded status information is for informing the decoder that the decoder should not refrain from decoding the first portion of the picture.

5

30. The method of claim 29, wherein
determining whether the first portion is required to be decoded by a decoder comprises receiving input from a pre-processor, wherein the input is used to determine whether the first portion is required to be decoded by a decoder.

10

31. The method of claim 30, wherein
the input comprises picture partitioning information that identifies the portions of the picture and, for each identified portion, a status for the portion, and
the picture partitioning information identifies a region of interest (RoI) within the picture.

15

32. A method (1400) performed by a network node (1104), the method comprising:
receiving (s1402) a first unit comprising encoded picture sample values corresponding to a first portion of a picture and first status information corresponding to the first portion of the picture; and

20

based on the first status information, deciding (s1402) whether or not to forward the first unit to a receiving device.

33. The method of claim 32, wherein
the first status information comprises a first status indicator value, and
deciding whether or not to forward the first unit to a receiving device based on the status information comprises determining whether the first status indicator value satisfies a criterion.

25

34. The method of claim 33, wherein the method further comprises:
receiving second status information corresponding to the second portion of the picture;

30 and

based on the second status information, deciding whether or not to forward the second unit to the receiving device, wherein

the second status information comprises a second status indicator value that is different than the first status indicator value, and

5 deciding whether or not to forward the second unit to a receiving device based on the second status information comprises determining whether the second status indicator value satisfies said criterion.

10 35. A computer program (1643) comprising instructions (1644) which when executed by processing circuitry (1602) of an apparatus (1600) causes the apparatus to perform the method of any one of claims 1-34.

15 36. A carrier containing the computer program of claim 35, wherein the carrier is one of an electronic signal, an optical signal, a radio signal, and a computer readable storage medium (1642).

37. An apparatus (1600) for decoding a picture from a bitstream, the picture comprising a first portion and a second portion, the apparatus being configured to:

20 obtain (s1202), from the bitstream, first status information corresponding to the first portion of the picture, wherein the first status information does not correspond to the second portion; and

determine (s1204) whether or not to decode the first portion of the picture based on the first status information.

25 38. The apparatus of claim 37, wherein the apparatus is further configured to perform the method of any one of claims 2-28.

39. An apparatus (1600) for producing a bitstream comprising a picture, the picture comprising a first portion and a second portion, the apparatus being configured to:

30 determine (s1302) whether the first portion is required to be decoded by a decoder; and

as a result of determining that the first portion is required to be decoded by a decoder, produce (s1304) encoded status information for the first portion and including the encoded status information in the bitstream, wherein

the encoded status information for the first portion does not correspond to the second
5 portion of the picture, and

the encoded status information is for informing the decoder that the decoder should not refrain from decoding the first portion of the picture.

10 40. The apparatus of claim 39, wherein the apparatus is further configured to perform the method of any one of claims 30-31.

41. A network node (1104), the network node being configured to:

receive (s1402) a first unit comprising encoded picture sample values corresponding to a
15 first portion of a picture and first status information corresponding to the first portion of the picture; and

based on the first status information, decide (s1402) whether or not to forward the first unit to a receiving device.

20 42. The network node of claim 41, wherein the network node is further configured to perform the method of any one of claims 33-34.

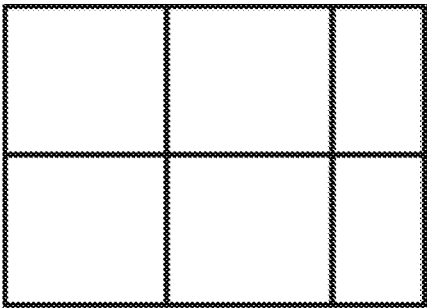


FIG. 1A

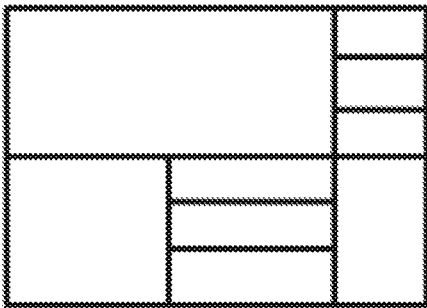


FIG. 1B

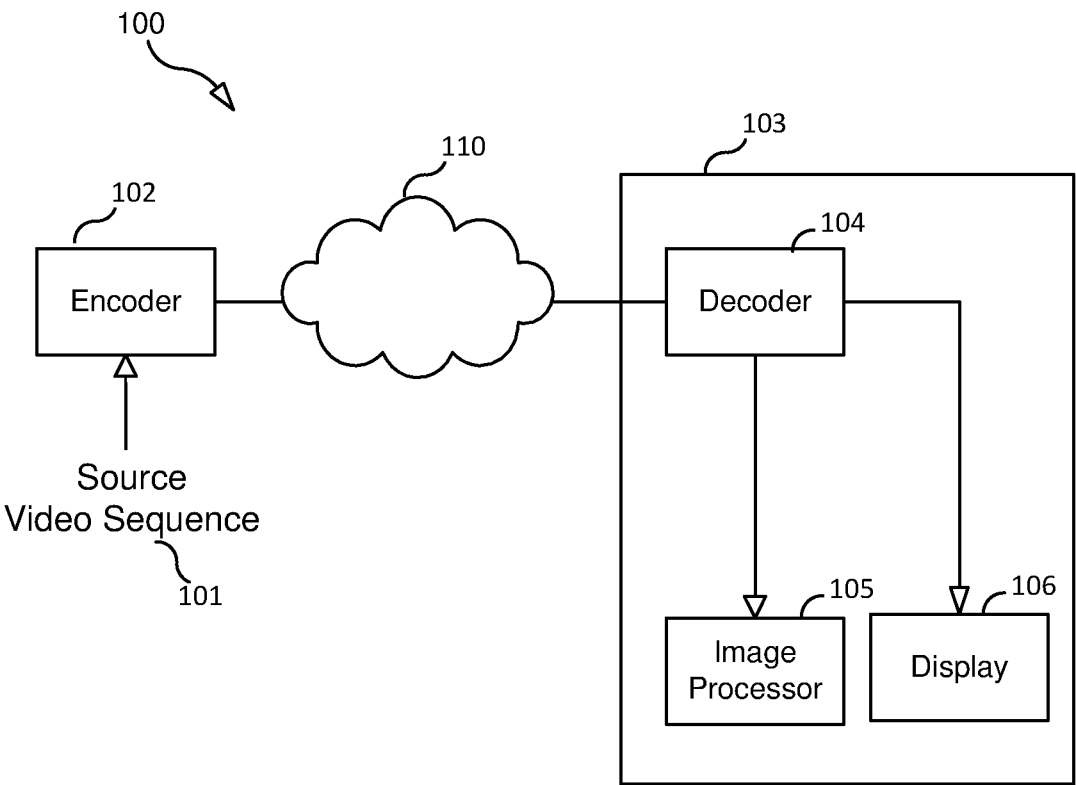


FIG. 1C

2/14

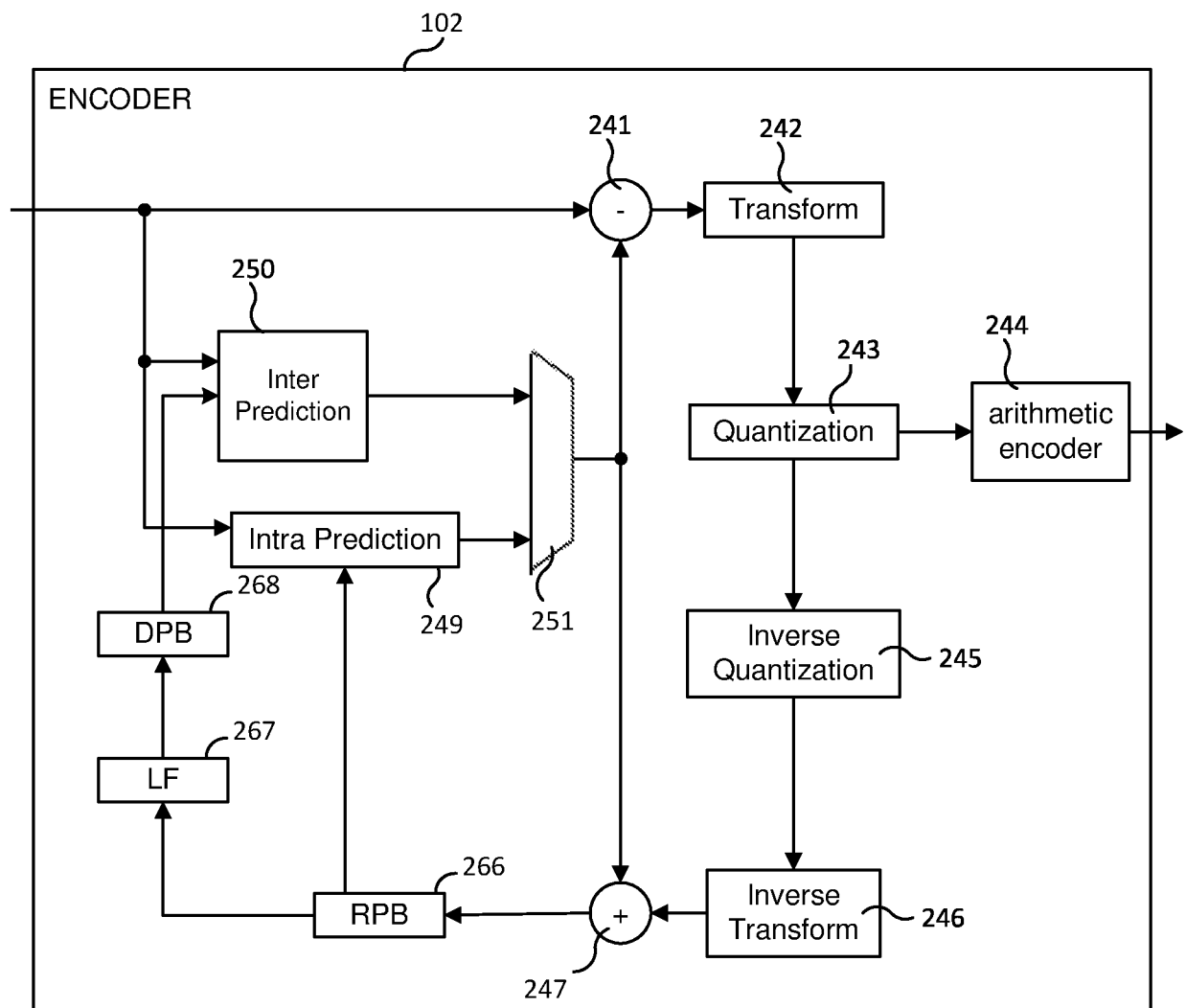


FIG. 2

3/14

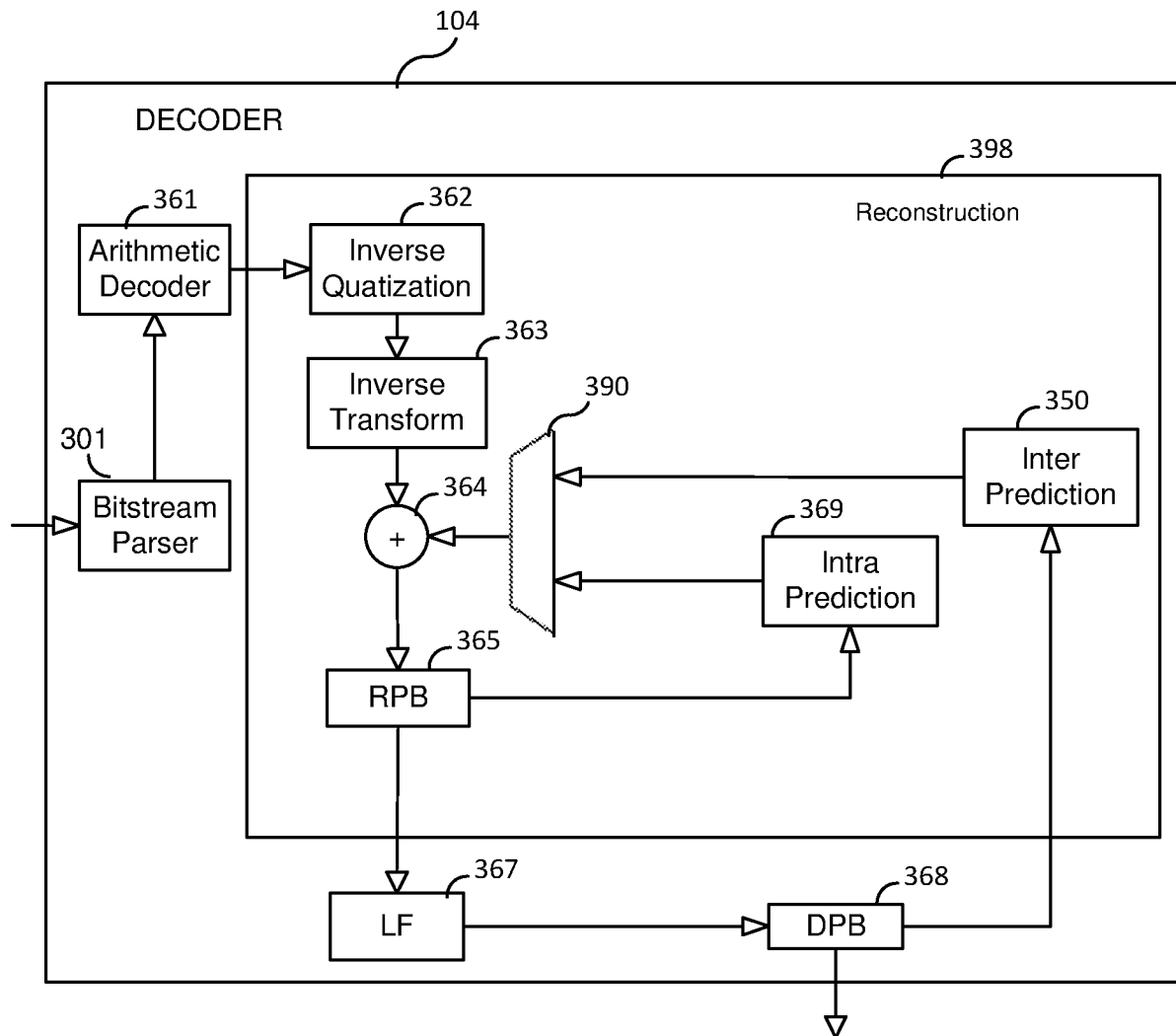
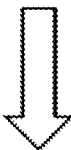
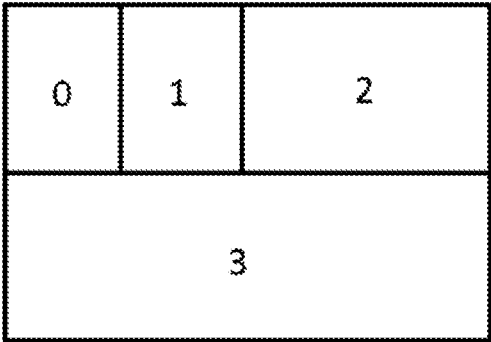


FIG. 3

0	1	2
3	4	5
6	7	8

FIG. 4A

A first layout specifies
subpicture partitioning



A second layout specifies
subpicture 0 and subpicture 1
are master subpictures

Portion of the picture
with master status

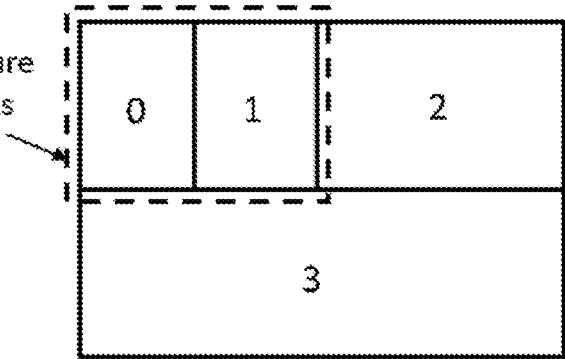


FIG. 4B

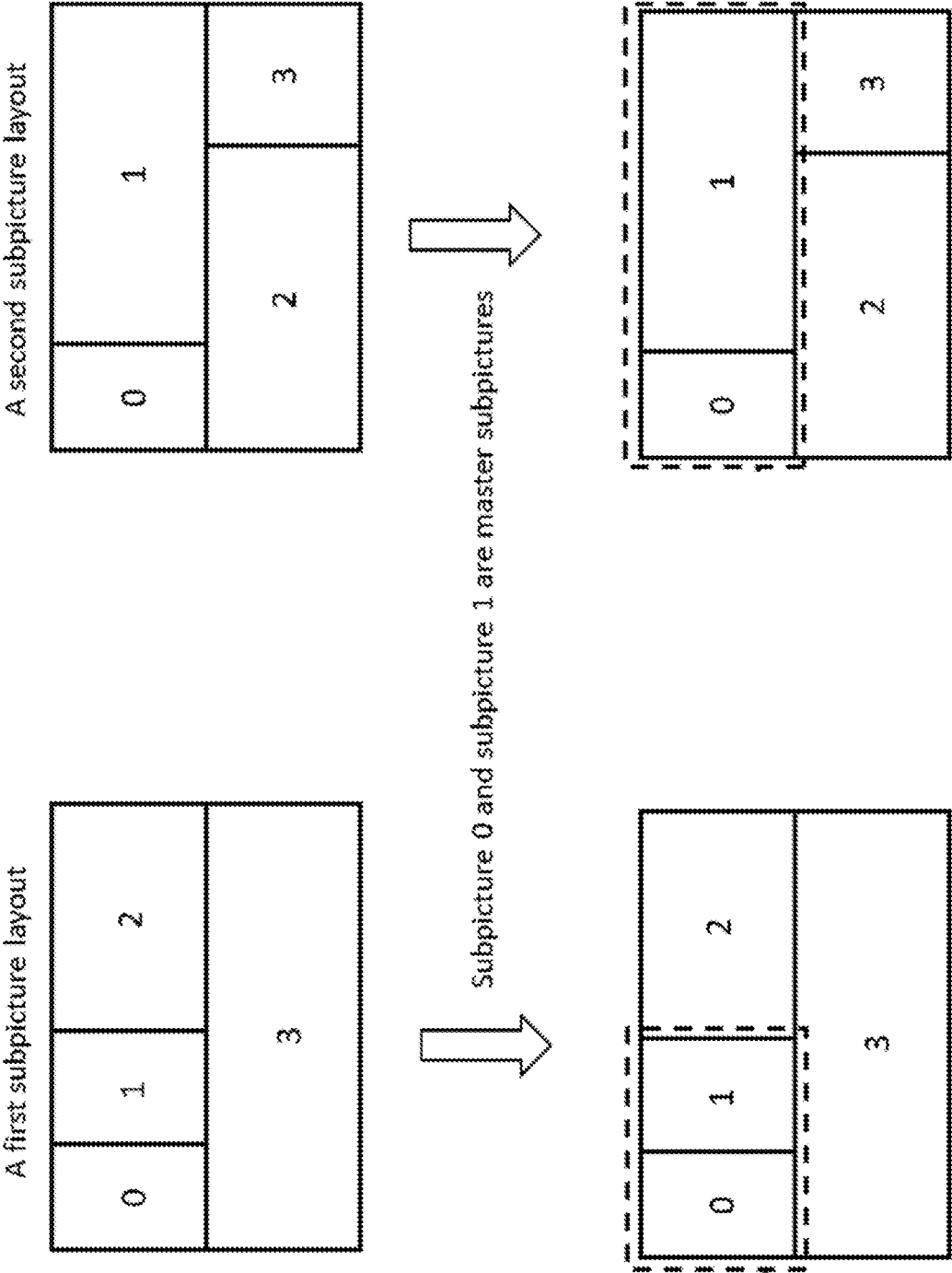


FIG. 5

6/14

A first subpicture layout

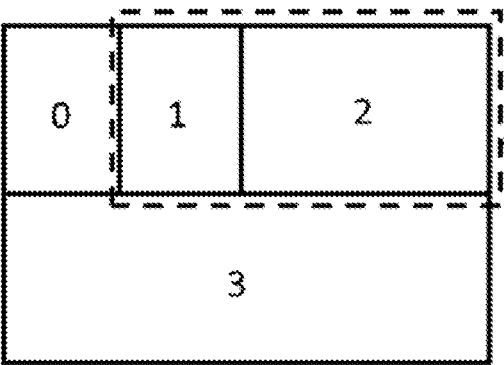
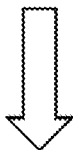
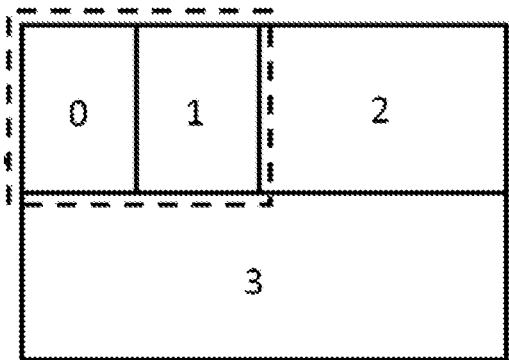
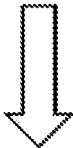
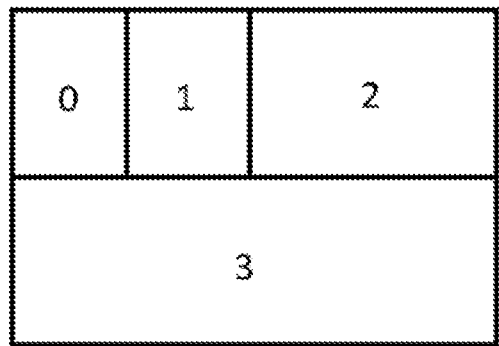


FIG. 6

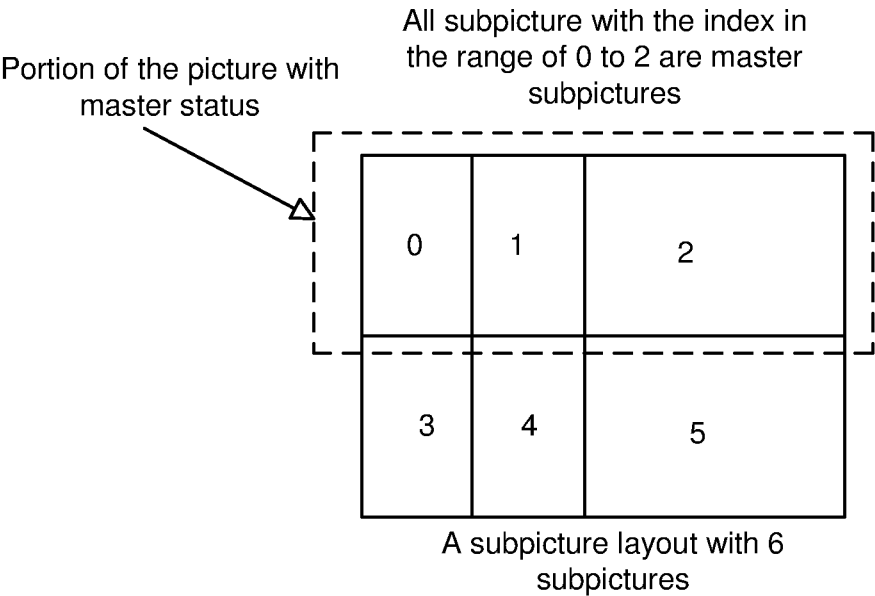


FIG. 7

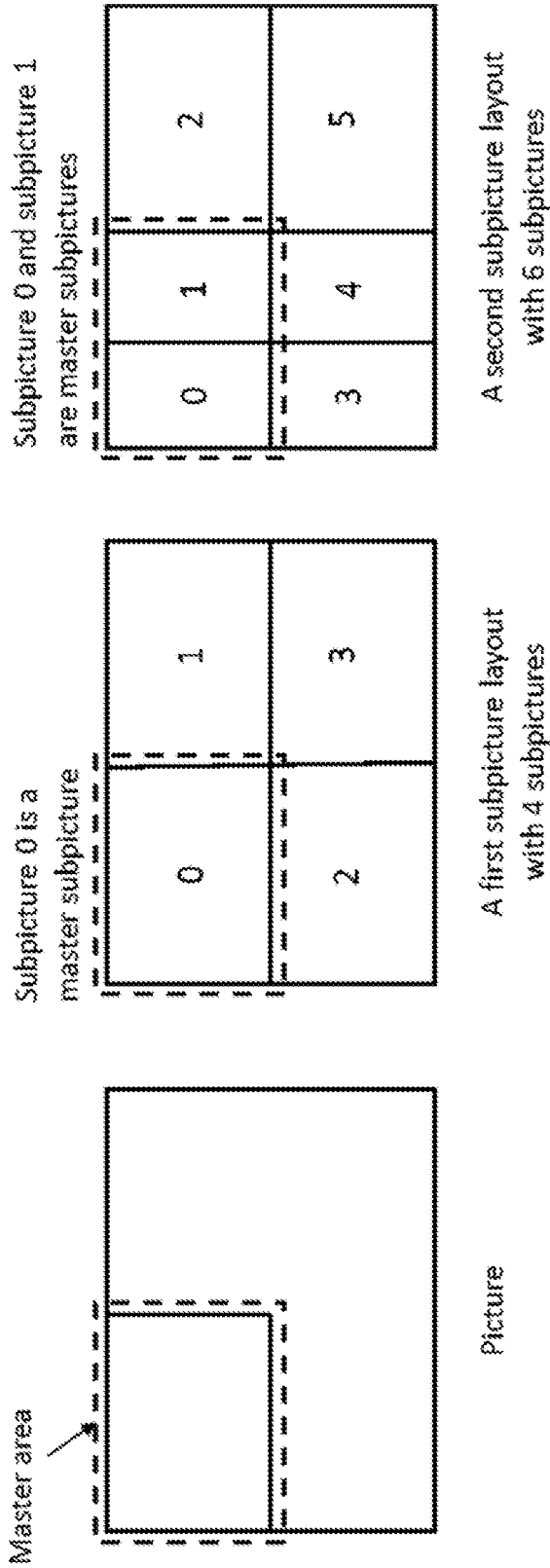


FIG. 8

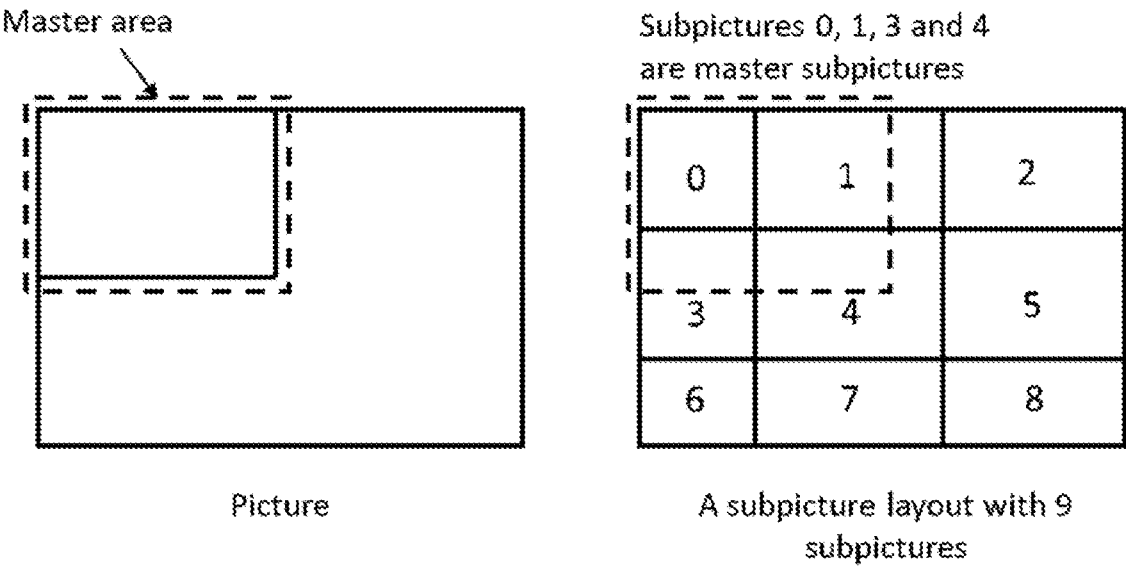


FIG. 9

10/14

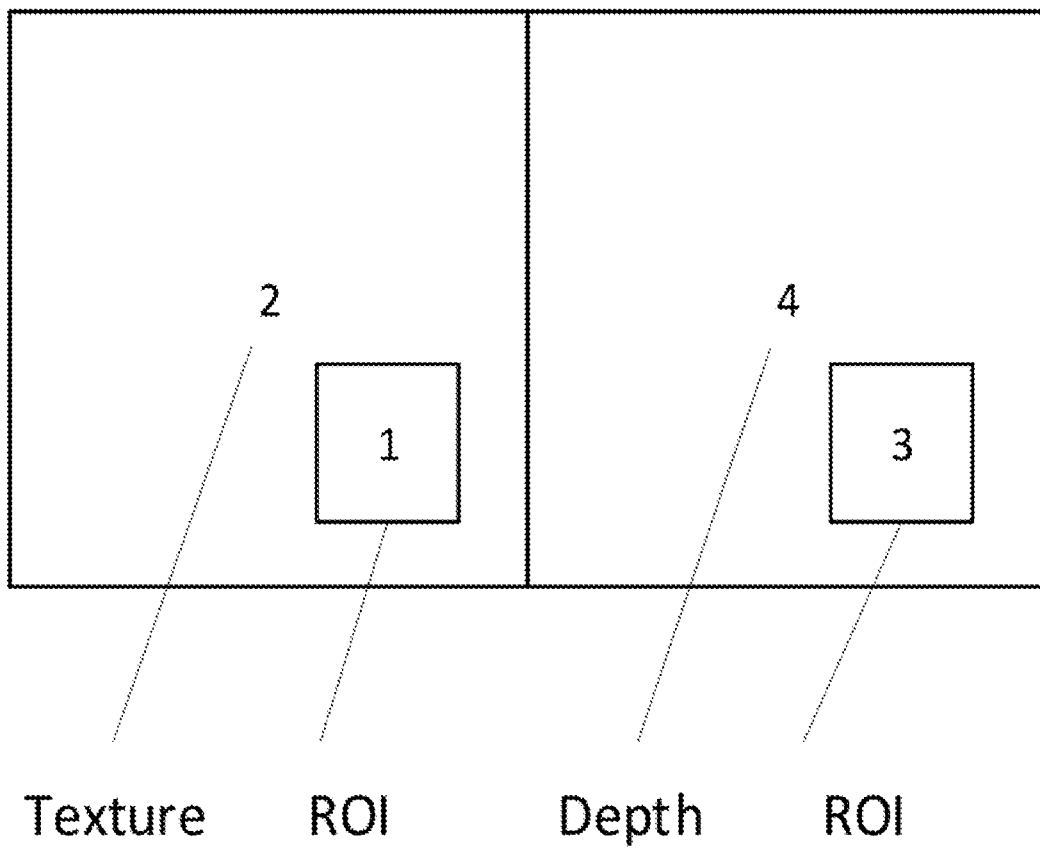


FIG. 10

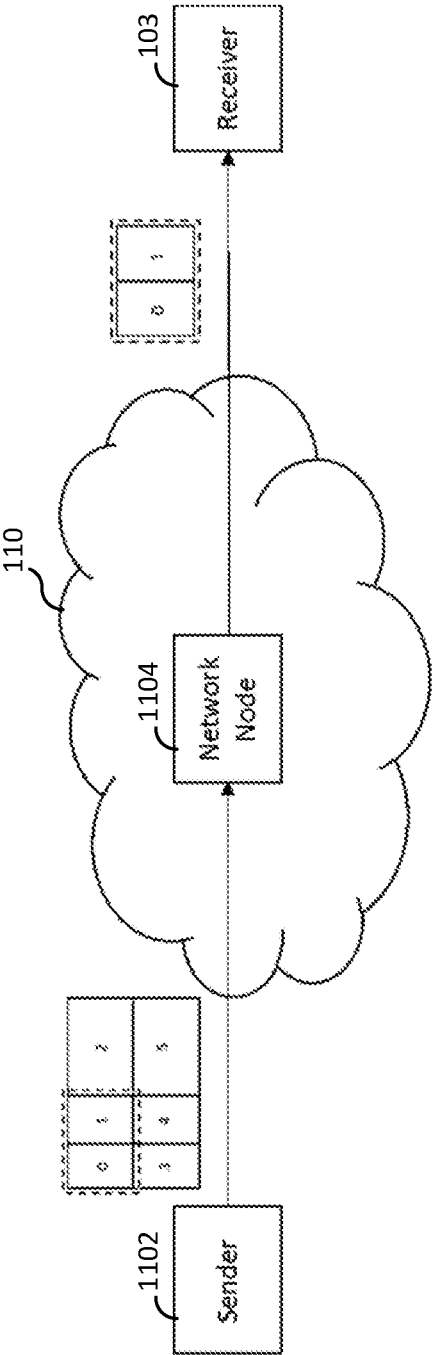
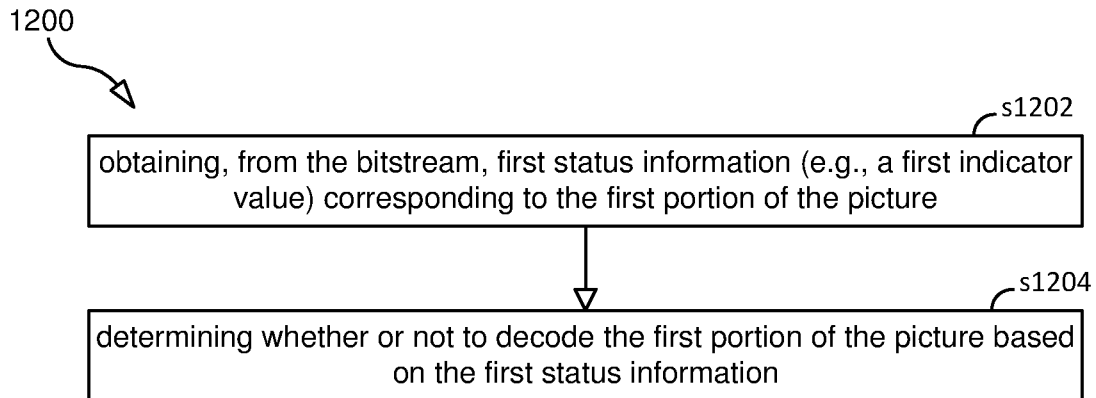
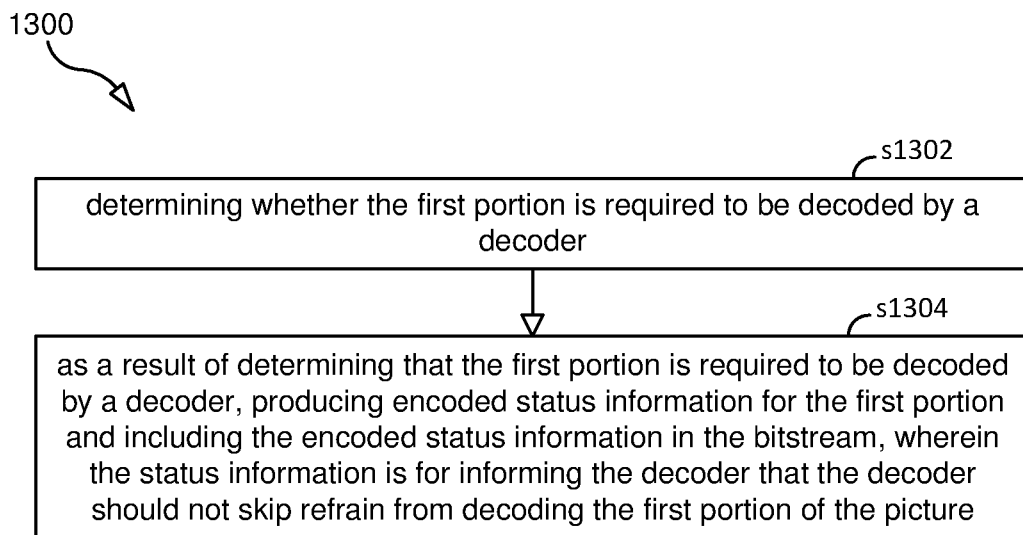


FIG. 11

12/14

**FIG. 12****FIG. 13**

13/14

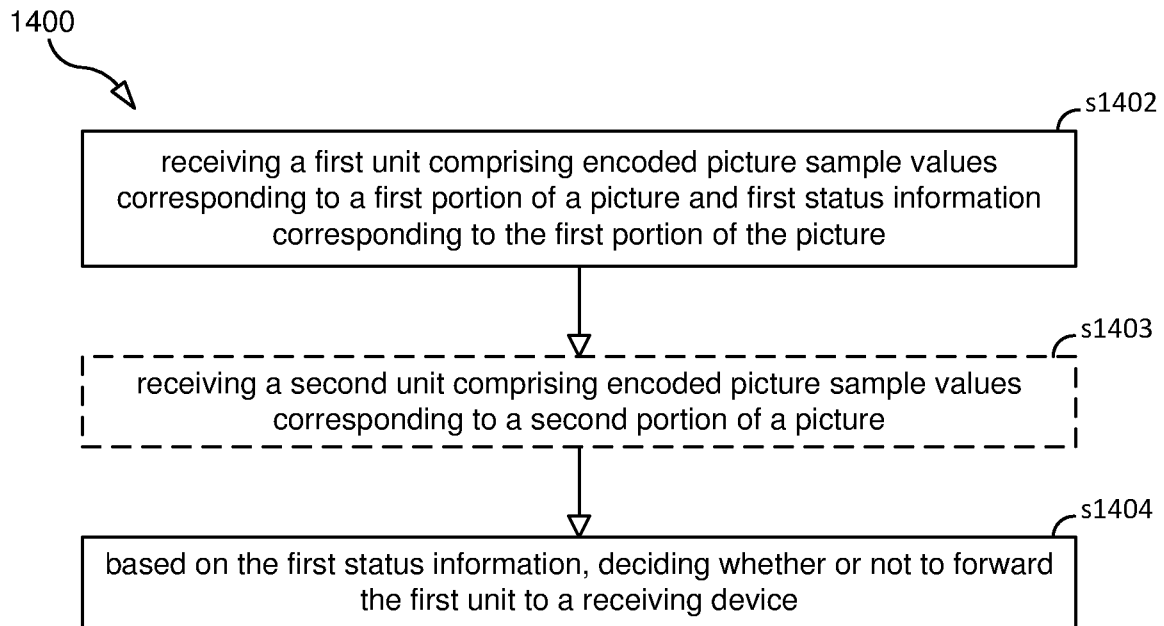


FIG. 14

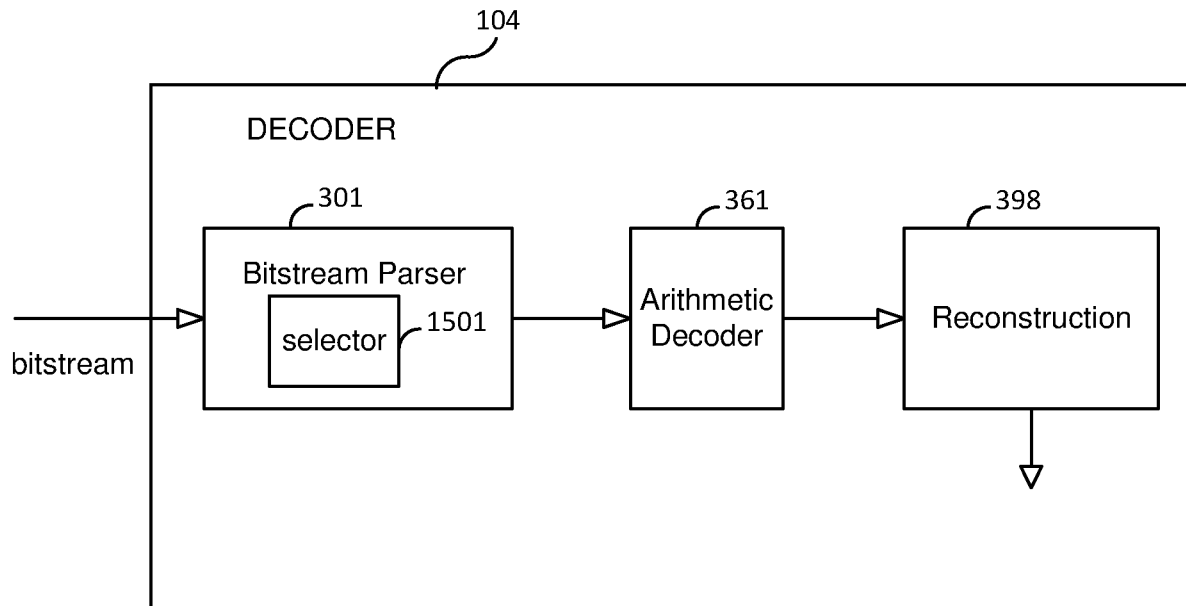


FIG. 15

14/14

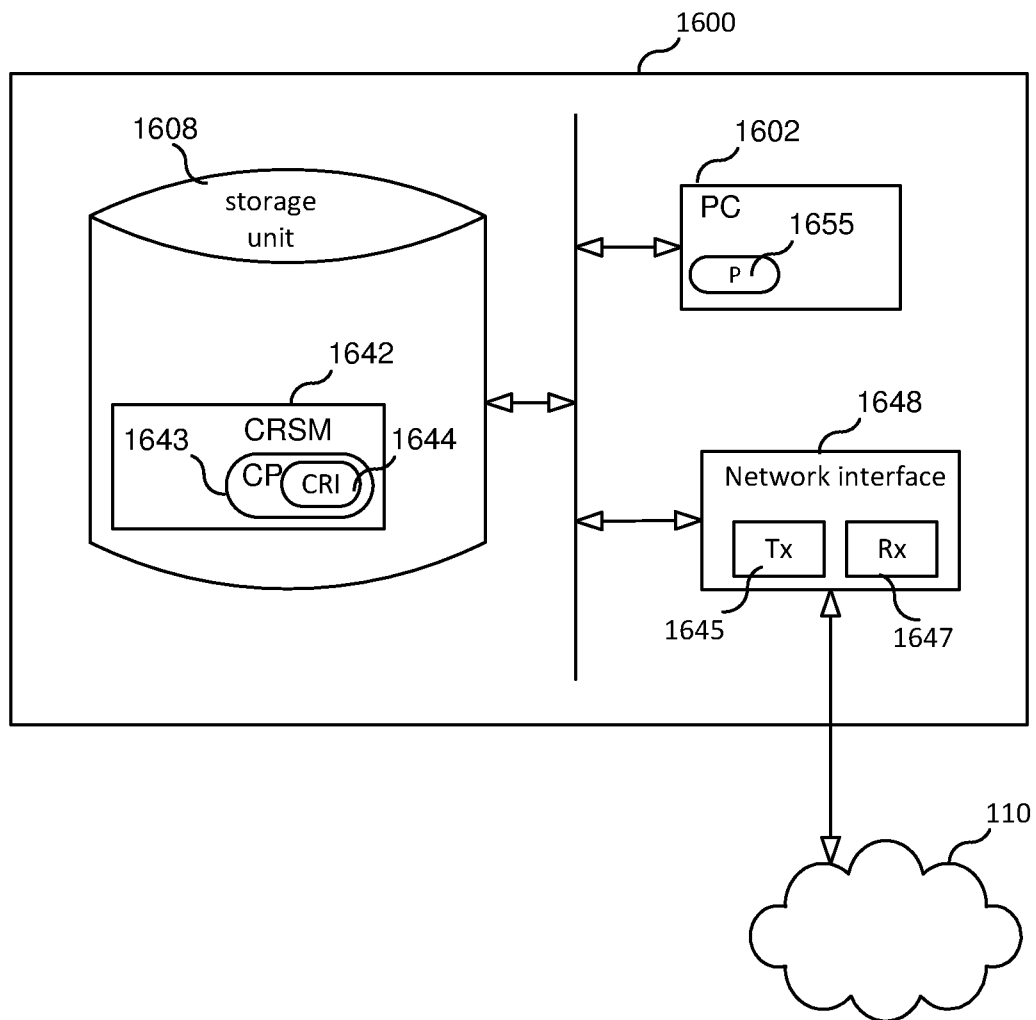


FIG. 16

INTERNATIONAL SEARCH REPORT

 International application No.
 PCT/SE2024/050961

A. CLASSIFICATION OF SUBJECT MATTER

IPC: see extra sheet

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

IPC: H04N

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

SE, DK, FI, NO classes as above

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, PAJ, WPI data, COMPENDEX, INSPEC, KIME

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 20210400295 A1 (LI MING ET AL), 23 December 2021 (2021-12-23); paragraphs [0031], [0037]-[0038], [0041]-[0050], [0057], [0060], [0066]; figures 1A,1B,6,7A,7B --	1-42
A	US 20220417544 A1 (CHOI BYEONGDOO ET AL), 29 December 2022 (2022-12-29); paragraphs [0121]-[0130], [0187]; figures 10,13 --	1-42
A	WANG YE-KUI et al. "The High-Level Syntax of the Versatile Video Coding (VVC) Standard", IEEE Transactions on Circuits and Systems for Video Technology, 2021-04-05, IEEE, USA, Vol. 31, nr. 10, p. 3779-3800, ISSN 1051-8215; whole document -- -----	1-42

☐ Further documents are listed in the continuation of Box C.
 ☒ See patent family annex.

* Special categories of cited documents:	"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"A" document defining the general state of the art which is not considered to be of particular relevance	"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"D" document cited by the applicant in the international application	"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"E" earlier application or patent but published on or after the international filing date	"&" document member of the same patent family
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)	
"O" document referring to an oral disclosure, use, exhibition or other means	
"P" document published prior to the international filing date but later than the priority date claimed	

 Date of the actual completion of the international search
 17-01-2025

 Date of mailing of the international search report
 20-01-2025

 Name and mailing address of the ISA/SE
 Patent- och registreringsverket
 Box 5055
 S-102 42 STOCKHOLM
 Facsimile No. + 46 8 666 02 86

 Authorized officer
 Alexander Lakic
 Telephone No. + 46 8 782 28 00

Continuation of: second sheet

International Patent Classification (IPC)

H04N 19/17 (2014.01)

H04N 19/119 (2014.01)

H04N 19/167 (2014.01)

H04N 19/44 (2014.01)

H04N 19/70 (2014.01)

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No.

PCT/SE2024/050961

US	20210400295 A1	23/12/2021	CN	113545060 A	22/10/2021
			EP	3935843 A4	05/10/2022
			JP	7416820 B2	17/01/2024
			JP	2022523440 A	22/04/2022
			KR	20210129210 A	27/10/2021
			WO	2020181435 A1	17/09/2020
US	20220417544 A1	29/12/2022	CN	116264863 B	15/11/2024
			EP	4169255 A1	26/04/2023
			JP	7540828 B2	27/08/2024
			JP	2023543590 A	17/10/2023
			KR	20230157482 A	16/11/2023
			US	11785239 B2	10/10/2023
			WO	2023279026 A1	05/01/2023