(12) INTERNATIONAL APPLICATION PUBLISHED UNDER THE PATENT COOPERATION TREATY (PCT)

(19) **World Intellectual Property Organization**
International Bureau

(43) **International Publication Date**
28 February 2019 (28.02.2019)

WIPO | PCT

(10) **International Publication Number**
WO 2019/038473 A1

(51) **International Patent Classification:**
H04N 21/4728 (2011.01) *H04N 19/167* (2014.01)
H04N 21/218 (2011.01) *H04N 19/85* (2014.01)
H04N 21/2343 (2011.01) *H04N 19/119* (2014.01)

(21) **International Application Number:**
PCT/FI2018/050587

(22) **International Filing Date:**
17 August 2018 (17.08.2018)

(25) **Filing Language:** English

(26) **Publication Language:** English

(30) **Priority Data:**
20175757 24 August 2017 (24.08.2017) FI

(71) **Applicant: NOKIA TECHNOLOGIES OY** [FI/FI]; Karaportti 3, 02610 Espoo (FI).

(72) **Inventor: HANNUKSELA, Miska**; Rusthollinrinne 2, 33610 Tampere (FI).

(74) **Agent: NOKIA TECHNOLOGIES OY** et al.; Ari Aarnio, IPR Department, Karakaari 7, 02610 Espoo (FI).

(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, CZ, DE, DJ, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IR, IS, JO, JP, KE, KG, KH, KN, KP, KR, KW, KZ, LA, LC, LK, LR, LS, LU, LY, MA, MD, ME, MG, MK, MN, 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, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

(84) **Designated States** *(unless otherwise indicated, for every kind of regional protection available)*: ARIPO (BW, GH, GM, KE, LR, LS, MW, MZ, NA, RW, 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, 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))*

(54) **Title:** AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR OMNIDIRECTIONAL VIDEO

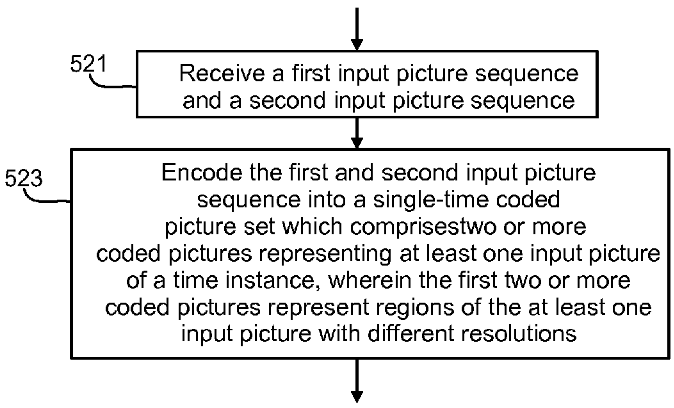


Fig. 10

(57) **Abstract:** There are disclosed various methods, apparatuses and computer program products for video encoding and decoding. In some embodiments a single-time coded picture set is generated comprising first two or more coded pictures representing at least one input picture of a first time instance. The first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

WO 2019/038473 A1

AN APPARATUS, A METHOD AND A COMPUTER PROGRAM FOR OMNIDIRECTIONAL VIDEO

TECHNICAL FIELD

[0001] The present invention relates to an apparatus, a method and a computer program for resolution-adaptive tile merging for viewport-adaptive streaming of omnidirectional video coding and decoding.

BACKGROUND

[0002] This section is intended to provide a background or context to the invention that is recited in the claims. The description herein may include concepts that could be pursued, but are not necessarily ones that have been previously conceived or pursued. Therefore, unless otherwise indicated herein, what is described in this section is not prior art to the description and claims in this application and is not admitted to be prior art by inclusion in this section.

[0003] A video coding system may comprise an encoder that transforms an input video into a compressed representation suited for storage/transmission and a decoder that can uncompress the compressed video representation back into a viewable form. The encoder may discard some information in the original video sequence in order to represent the video in a more compact form, for example, to enable the storage/transmission of the video information at a lower bitrate than otherwise might be needed.

[0004] Various technologies for providing three-dimensional (3D) video content are currently investigated and developed. Especially, intense studies have been focused on various multiview applications wherein a viewer is able to see only one pair of stereo video from a specific viewpoint and another pair of stereo video from a different viewpoint. One of the most feasible approaches for such multiview applications has turned out to be such wherein only a limited number of input views, e.g. a mono or a stereo video plus some supplementary data, is provided to a decoder side and all required views are then rendered (i.e. synthesized) locally by the decoder to be displayed on a display.

[0005] In the encoding of 3D video content, video compression systems, such as Advanced Video Coding standard (H.264/AVC), the Multiview Video Coding (MVC) extension of H.264/AVC or scalable extensions of HEVC (High Efficiency Video Coding) can be used.

SUMMARY

[0006] Some embodiments provide a method for encoding and decoding video information. In some embodiments of the present invention there is provided a method, apparatus and computer program product for video coding as well as decoding.

[0007] Various aspects of examples of the invention are provided in the detailed description.

[0008] According to a first aspect, there is provided a method comprising:

generating a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

[0009] An apparatus according to a second aspect comprises at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform at least:

generate a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

[0010] A computer readable storage medium according to a third aspect comprises code for use by an apparatus, which when executed by a processor, causes the apparatus to perform:

generate a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

[0011] An apparatus according to a fourth aspect comprises:

means for generating a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different

resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

[0012] A method according to a fifth aspect comprises:

obtaining a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decoding, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decoding a coded picture of the single-time coded picture set into a decoded picture;

using the separate region-wise packing information for the decoded picture in displaying the decoded picture.

[0013] An apparatus according to a sixth aspect comprises at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform at least:

obtain a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decode, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decode a coded picture of the single-time coded picture set into a decoded picture;

use the separate region-wise packing information for the decoded picture in displaying the decoded picture.

[0014] A computer readable storage medium according to a seventh aspect comprises code for use by an apparatus, which when executed by a processor, causes the apparatus to perform:

obtain a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decode, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decode a coded picture of the single-time coded picture set into a decoded picture;

use the separate region-wise packing information for the decoded picture in displaying the decoded picture.

[0015] An apparatus according to an eight aspect comprises:

means for obtaining a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

means for decoding, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

means for decoding a coded picture of the single-time coded picture set into a decoded picture;

means for using the separate region-wise packing information for the decoded picture in displaying the decoded picture.

[0016] Further aspects include at least apparatuses and computer program products/code stored on a non-transitory memory medium arranged to carry out the above methods.

BRIEF DESCRIPTION OF THE DRAWINGS

[0017] For a more complete understanding of example embodiments of the present invention, reference is now made to the following descriptions taken in connection with the accompanying drawings in which:

[0018] Figure 1a shows an example of a multi-camera system as a simplified block diagram, in accordance with an embodiment;

[0019] Figure 1b shows a perspective view of a multi-camera system, in accordance with an embodiment;

[0020] Figure 2a illustrates image stitching, projection, and mapping processes, in accordance with an embodiment;

[0021] Figure 2b illustrates a process of forming a monoscopic equirectangular panorama picture, in accordance with an embodiment;

[0022] Figure 3 shows an example of mapping a higher resolution sampled front face of a cube map on the same packed virtual reality frame as other cube faces, in accordance with an embodiment;

[0023] Figure 4 shows an example of merging coded rectangle sequences into a bitstream, in accordance with an embodiment;

[0024] Figure 5 shows an example how extractor tracks can be used for tile-based omnidirectional video streaming, in accordance with an embodiment;

[0025] Figure 6a illustrates an example of combining tiles from bitstreams of different resolution and using a single decoder for decoding the resulting the bitstream, in accordance with an embodiment;

[0026] Figure 6b illustrates an example of two views of an input picture of a cube map projection format at two different resolutions, in accordance with an embodiment;

[0027] Figure 6c illustrates an example of a high-resolution temporally frame packet bitstream and low-resolution bitstreams of a first view and a second view, in accordance with an embodiment;

[0028] Figure 6d illustrates how an extractor track may be generated and/or tiles may be extracted from encoded bitstreams to generate a spatiotemporal frame-packed bitstream;

[0029] Figure 6e illustrates how a single-time coded picture set can be obtained from a stereoscopic omnidirectional input picture sequence of an equirectangular projection format, in accordance with an embodiment;

[0030] Figure 6f illustrates how a single-time coded picture set can be obtained from a stereoscopic omnidirectional input picture sequence of an equirectangular projection format, in accordance with another embodiment;

[0031] Figure 6g illustrates how a single-time coded picture set can be obtained from a monoscopic omnidirectional input picture sequence of an equirectangular projection format, in accordance with an embodiment;

[0032] Figure 7 shows an example of a hierarchical data model used in DASH;

[0033] Figure 8a shows a schematic diagram of an encoder suitable for implementing embodiments of the invention;

[0034] Figure 8b shows a schematic diagram of a decoder suitable for implementing embodiments of the invention;

[0035] Figure 9a shows some elements of a video encoding section, in accordance with an embodiment;

[0036] Figure 9b shows an extractor of a video decoding section, in accordance with an embodiment;

[0037] Figure 10 shows a flow chart of an encoding method, in accordance with an embodiment;

[0038] Figure 11 shows a schematic diagram of an example multimedia communication system within which various embodiments may be implemented;

[0039] Figure 12 shows schematically an electronic device employing embodiments of the invention;

[0040] Figure 13 shows schematically a user equipment suitable for employing embodiments of the invention;

[0041] Figure 14 further shows schematically electronic devices employing embodiments of the invention connected using wireless and wired network connections.

DETAILED DESCRIPTON OF SOME EXAMPLE EMBODIMENTS

[0042] In the following, several embodiments of the invention will be described in the context of one video coding arrangement. It is to be noted, however, that the invention is not limited to this particular arrangement. In fact, the different embodiments have applications widely in any environment where improvement of coding when switching between coded fields and frames is desired. For example, the invention may be applicable to video coding systems like streaming systems, DVD players, digital television receivers, personal video recorders, systems and computer programs on personal computers, handheld computers and communication devices, as well as network elements such as transcoders and cloud computing arrangements where video data is handled.

[0043] In the following, several embodiments are described using the convention of referring to (de)coding, which indicates that the embodiments may apply to decoding and/or encoding.

[0044] The Advanced Video Coding standard (which may be abbreviated AVC or H.264/AVC) was developed by the Joint Video Team (JVT) of the Video Coding Experts Group (VCEG) of the Telecommunications Standardization Sector of International Telecommunication Union (ITU-T) and the Moving Picture Experts Group (MPEG) of International Organisation for Standardization (ISO) / International Electrotechnical Commission (IEC). The H.264/AVC standard is published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.264 and ISO/IEC International Standard 14496-10, also known as MPEG-4 Part 10 Advanced Video Coding (AVC). There have been multiple versions of the H.264/AVC standard, each integrating new extensions or features to the specification. These extensions include Scalable Video Coding (SVC) and Multiview Video Coding (MVC).

[0045] The High Efficiency Video Coding standard (which may be abbreviated HEVC or H.265/HEVC) was developed by the Joint Collaborative Team – Video Coding (JCT-VC) of VCEG and MPEG. The standard is published by both parent standardization organizations, and it is referred to as ITU-T Recommendation H.265 and ISO/IEC International Standard 23008-2, also known as MPEG-H Part 2 High Efficiency Video Coding (HEVC). Extensions to H.265/HEVC include scalable, multiview, three-dimensional, and fidelity range extensions, which may be referred to as SHVC, MV-HEVC, 3D-HEVC, and REXT, respectively. The references in this description to H.265/HEVC, SHVC, MV-HEVC, 3D-HEVC and REXT that have been made for the purpose of understanding definitions, structures or concepts of these standard specifications are to be understood to be references to the latest versions of these standards that were available before the date of this application, unless otherwise indicated.

[0046] Some key definitions, bitstream and coding structures, and concepts of H.264/AVC and HEVC and some of their extensions are described in this section as an example of a video encoder, decoder, encoding method, decoding method, and a bitstream structure, wherein the embodiments may be implemented. Some of the key definitions, bitstream and coding structures, and concepts of H.264/AVC are the same as in HEVC standard – hence, they are described below jointly. The aspects of the invention are not limited to H.264/AVC or HEVC or their extensions, but rather the description is given for one possible basis on top of which the invention may be partly or fully realized.

[0047] In the description of existing standards as well as in the description of example embodiments, a syntax element may be defined as an element of data represented in the bitstream. A syntax structure may be defined as zero or more syntax elements present together in the bitstream in a specified order.

[0048] Similarly to many earlier video coding standards, the bitstream syntax and semantics as well as the decoding process for error-free bitstreams are specified in H.264/AVC and HEVC. The encoding process is not specified, but encoders must generate conforming bitstreams. Bitstream and decoder conformance can be verified with the Hypothetical Reference Decoder (HRD). The standards contain coding tools that help in coping with transmission errors and losses, but the use of the tools in encoding is optional and no decoding process has been specified for erroneous bitstreams.

[0049] The elementary unit for the input to an H.264/AVC or HEVC encoder and the output of an H.264/AVC or HEVC decoder, respectively, is a picture. A picture given as an input to an encoder may also be referred to as a source picture, and a picture decoded by a decoder may be referred to as a decoded picture.

[0050] The source and decoded pictures may each be comprised of one or more sample arrays, such as one of the following sets of sample arrays:

- Luma (Y) only (monochrome).
- Luma and two chroma (YCbCr or YCgCo).
- Green, Blue and Red (GBR, also known as RGB).
- Arrays representing other unspecified monochrome or tri-stimulus color samplings (for example, YZX, also known as XYZ).

[0051] In the following, these arrays may be referred to as luma (or L or Y) and chroma, where the two chroma arrays may be referred to as Cb and Cr; regardless of the actual color representation method in use. The actual color representation method in use may be indicated e.g. in a coded bitstream e.g. using the Video Usability Information (VUI) syntax of H.264/AVC and/or HEVC. A component may be defined as an array or a single sample from one of the three sample arrays (luma and two chroma) or the array or a single sample of the array that compose a picture in monochrome format.

[0052] In H.264/AVC and HEVC, a picture may either be a frame or a field. A frame comprises a matrix of luma samples and possibly the corresponding chroma samples. A field is a set of alternate sample rows of a frame. Fields may be used as encoder input for example when the source signal is interlaced. Chroma sample arrays may be absent (and hence monochrome sampling may be in use) or may be subsampled when compared to luma sample arrays. Some chroma formats may be summarized as follows:

- In monochrome sampling there is only one sample array, which may be nominally considered the luma array.
- In 4:2:0 sampling, each of the two chroma arrays has half the height and half the width of the luma array.
- In 4:2:2 sampling, each of the two chroma arrays has the same height and half the width of the luma array.
- In 4:4:4 sampling when no separate color planes are in use, each of the two chroma arrays has the same height and width as the luma array.

[0053] In H.264/AVC and HEVC, it is possible to code sample arrays as separate color planes into the bitstream and respectively decode separately coded color planes from the bitstream. When separate color planes are in use, each one of them is separately processed (by the encoder and/or the decoder) as a picture with monochrome sampling.

[0054] When chroma subsampling is in use (e.g. 4:2:0 or 4:2:2 chroma sampling), the location of chroma samples with respect to luma samples may be determined in the encoder side (e.g. as pre-processing step or as part of encoding). The chroma sample positions with respect to luma sample positions may be pre-defined for example in a coding standard, such as H.264/AVC or HEVC, or may be indicated in the bitstream for example as part of VUI of H.264/AVC or HEVC.

[0055] Generally, the source video sequence(s) provided as input for encoding may either represent interlaced source content or progressive source content. Fields of opposite parity have been captured at different times for interlaced source content. Progressive source content contains captured frames. An encoder may encode fields of interlaced source content in two ways: a pair of interlaced fields may be coded into a coded frame or a field may be coded as a coded field. Likewise, an encoder may encode frames of progressive source content in two ways: a frame of progressive source content may be coded into a coded frame or a pair of coded fields. A field pair or a complementary field pair may be defined as two fields next to each other in decoding and/or output order, having opposite parity (i.e. one being a top field and another being a bottom field) and neither belonging to any other complementary field pair. Some video coding standards or schemes allow mixing of coded frames and coded fields in the same coded video sequence. Moreover, predicting a coded field from a field in a coded frame and/or predicting a coded frame for a complementary field pair (coded as fields) may be enabled in encoding and/or decoding.

[0056] A partitioning may be defined as a division of a set into subsets such that each element of the set is in exactly one of the subsets. A picture partitioning may be defined as a division of a picture into smaller non-overlapping units. A block partitioning may be defined as a division of a block into smaller non-overlapping units, such as sub-blocks. In some cases term block partitioning may be considered to cover multiple levels of partitioning, for example partitioning of a picture into slices, and partitioning of each slice into smaller units, such as macroblocks of H.264/AVC. It is noted that the same unit, such as a picture, may have more than one partitioning. For example, a coding unit of HEVC may be partitioned into prediction units and separately by another quadtree into transform units.

[0057] A coded picture is a coded representation of a picture.

[0058] Video coding standards and specifications may allow encoders to divide a coded picture to coded slices or alike. In-picture prediction is typically disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture to independently decodable pieces. In H.264/AVC and HEVC, in-picture prediction may be disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture into independently decodable pieces, and slices are therefore often regarded as elementary units for transmission. In many cases, encoders may indicate in the bitstream which types of in-picture prediction are turned off across slice boundaries, and the decoder operation takes this information into account for example when concluding which prediction sources are available. For example, samples from a neighbouring macroblock or CU may be regarded as unavailable for intra prediction, if the neighbouring macroblock or CU resides in a different slice.

[0059] In H.264/AVC, a macroblock is a 16x16 block of luma samples and the corresponding blocks of chroma samples. For example, in the 4:2:0 sampling pattern, a macroblock contains one 8x8 block of chroma samples per each chroma component. In H.264/AVC, a picture is partitioned to one or more slice groups, and a slice group contains one or more slices. In H.264/AVC, a slice consists of an integer number of macroblocks ordered consecutively in the raster scan within a particular slice group.

[0060] When describing the operation of HEVC, the following terms may be used. A coding block may be defined as an NxN block of samples for some value of N such that the division of a coding tree block into coding blocks is a partitioning. A coding tree block (CTB) may be defined as an NxN block of samples for some value of N such that the division of a component into coding tree blocks is a partitioning. A coding tree unit (CTU) may be defined as a coding tree block of luma samples, two corresponding coding tree blocks of chroma samples of a picture that has three sample arrays, or a coding tree block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples. A coding unit (CU) may be defined as a coding block of luma samples, two corresponding coding blocks of chroma samples of a picture that has three sample arrays, or a coding block of samples of a monochrome picture or a picture that is coded using three separate color planes and syntax structures used to code the samples.

[0061] In some video codecs, such as High Efficiency Video Coding (HEVC) codec, video pictures are divided into coding units (CU) covering the area of the picture. A CU consists of one or more prediction units (PU) defining the prediction process for the samples within the CU and one or more transform units (TU) defining the prediction error coding process for the samples in the said CU. Typically, a CU consists of a square block of samples with a size selectable from a predefined set of possible CU sizes. A CU with the maximum allowed size may be named as LCU (largest coding unit) or coding tree unit (CTU) and the video picture is divided into non-overlapping LCUs. An LCU can be further split into a combination of smaller CUs, e.g. by recursively splitting the LCU and resultant CUs. Each resulting CU typically has at least one PU and at least one TU associated with it. Each PU and TU can be further split into smaller PUs and TUs in order to increase granularity of the prediction and prediction error coding processes, respectively. Each PU has prediction information associated with it defining what kind of a prediction is to be applied for the pixels within that PU (e.g. motion vector information for inter predicted PUs and intra prediction directionality information for intra predicted PUs).

[0062] Each TU can be associated with information describing the prediction error decoding process for the samples within the said TU (including e.g. DCT coefficient information). It is typically signalled at CU level whether prediction error coding is applied or not for each CU. In the case there is no prediction error residual associated with the CU, it can be considered there are no TUs for the said CU. The division of the image into CUs, and division of CUs into PUs and TUs is typically signalled in the bitstream allowing the decoder to reproduce the intended structure of these units.

[0063] In the HEVC standard, a picture can be partitioned in tiles, which are rectangular and contain an integer number of CTUs. In the HEVC standard, the partitioning to tiles forms a grid that may be characterized by a list of tile column widths (in CTUs) and a list of tile row heights (in CTUs). Tiles are ordered in the bitstream consecutively in the raster scan order of the tile grid. A tile may contain an integer number of slices.

[0064] In the HEVC, a slice consists of an integer number of CTUs. The CTUs are scanned in the raster scan order of CTUs within tiles or within a picture, if tiles are not in use. A slice may contain an integer number of tiles or a slice can be contained in a tile. Within a CTU, the CUs have a specific scan order.

[0065] In HEVC, a slice may be defined as an integer number of coding tree units contained in one independent slice segment and all subsequent dependent slice segments (if any) that precede the next independent slice segment (if any) within the same access unit. An independent slice segment may be defined as a slice segment for which the values of the syntax elements of the slice segment header are not inferred from the values for a preceding slice segment. A dependent slice segment may be defined as a slice segment for which the values of some syntax elements of the slice segment header are inferred from the values for the preceding independent slice segment in decoding order. In other words, only the independent slice segment may have a “full” slice header. An independent slice segment may be conveyed in one NAL unit (without other slice segments in the same NAL unit) and likewise a dependent slice segment may be conveyed in one NAL unit (without other slice segments in the same NAL unit).

[0066] In HEVC, a coded slice segment may be considered to comprise a slice segment header and slice segment data. A slice segment header may be defined as part of a coded slice segment containing the data elements pertaining to the first or all coding tree units represented in the slice segment. A slice header may be defined as the slice segment header of the independent slice segment that is a current slice segment or the most recent independent slice segment that precedes a current dependent slice segment in decoding order. Slice segment data may comprise an integer number of coding tree unit syntax structures.

[0067] In H.264/AVC and HEVC, in-picture prediction may be disabled across slice boundaries. Thus, slices can be regarded as a way to split a coded picture into independently decodable pieces, and slices are therefore often regarded as elementary units for transmission. In many cases, encoders may indicate in the bitstream which types of in-picture prediction are turned off across slice boundaries, and the decoder operation takes this information into account for example when concluding which prediction sources are available. For example, samples from a neighboring macroblock or CU may be regarded as unavailable for intra prediction, if the neighboring macroblock or CU resides in a different slice.

[0068] The elementary unit for the output of an H.264/AVC or HEVC encoder and the input of an H.264/AVC or HEVC decoder, respectively, is a Network Abstraction Layer (NAL) unit. For transport over packet-oriented networks or storage into structured files, NAL units may be encapsulated into packets or similar structures.

[0069] A NAL unit may be defined as a syntax structure containing an indication of the type of data to follow and bytes containing that data in the form of an RBSP interspersed as necessary with emulation prevention bytes. A raw byte sequence payload (RBSP) may be defined as a syntax structure containing an integer number of bytes that is encapsulated in a NAL unit. An RBSP is either empty or has the form of a string of data bits containing syntax elements followed by an RBSP stop bit and followed by zero or more subsequent bits equal to 0.

[0070] NAL units consist of a header and payload. In H.264/AVC, the NAL unit header indicates the type of the NAL unit and whether a coded slice contained in the NAL unit is a part of a reference picture or a non-reference picture. H.264/AVC includes a 2-bit nal_ref_idc syntax element, which when equal to 0 indicates that a coded slice contained in the NAL unit is a part of a non-reference picture and when greater than 0 indicates that a coded slice contained in the NAL unit is a part of a reference picture. The NAL unit header for SVC and MVC NAL units may additionally contain various indications related to the scalability and multiview hierarchy.

[0071] In HEVC, a two-byte NAL unit header is used for all specified NAL unit types. The NAL unit header contains one reserved bit, a six-bit NAL unit type indication (called nal_unit_type), a six-bit reserved field (called nuh_layer_id) and a three-bit temporal_id_plus1 indication for temporal level. The temporal_id_plus1 syntax element may be regarded as a temporal identifier for the NAL unit, and a zero-based TemporalId variable may be derived as follows: TemporalId = temporal_id_plus1 − 1. TemporalId equal to 0 corresponds to the lowest temporal level. The value of temporal_id_plus1 is required to be non-zero in order to avoid start code emulation involving the two NAL unit header bytes. The bitstream created by excluding all VCL NAL units having a TemporalId greater than or equal to a selected value and including all other VCL NAL units remains conforming. Consequently, a picture having TemporalId equal to TID does not use any picture having a TemporalId greater than TID as inter prediction reference. A sub-layer or a temporal sub-layer may be defined to be a temporal scalable layer of a temporal scalable bitstream, consisting of VCL NAL units with a particular value of the TemporalId variable and the associated non-VCL NAL units. Without loss of generality, in some example embodiments a variable LayerId is derived from the value of nuh_layer_id for example as follows: LayerId = nuh_layer_id. In the following, layer identifier, LayerId, nuh_layer_id and layer_id are used interchangeably unless otherwise indicated.

[0072] In HEVC extensions nuh_layer_id and/or similar syntax elements in NAL unit header carries scalability layer information. For example, the LayerId value nuh_layer_id and/or similar syntax elements may be mapped to values of variables or syntax elements describing different scalability dimensions.

[0073] NAL units can be categorized into Video Coding Layer (VCL) NAL units and non-VCL NAL units. VCL NAL units are typically coded slice NAL units. In H.264/AVC, coded slice NAL units contain syntax elements representing one or more coded macroblocks, each of which corresponds to a block of samples in the uncompressed picture. In HEVC, coded slice NAL units contain syntax elements representing one or more CU.

[0074] A non-VCL NAL unit may be for example one of the following types: a sequence parameter set, a picture parameter set, a supplemental enhancement information (SEI) NAL unit, an access unit delimiter, an end of sequence NAL unit, an end of bitstream NAL unit, or a filler data NAL unit. Parameter sets may be needed for the reconstruction of decoded pictures, whereas many of the other non-VCL NAL units are not necessary for the reconstruction of decoded sample values.

[0075] Parameters that remain unchanged through a coded video sequence may be included in a sequence parameter set. Examples of parameters that are required to be unchanged within a coded video sequence in many coding systems and hence included in a sequence parameter set are the width and height of the pictures included in the coded video sequence. In addition to the parameters that may be needed by the decoding process, the sequence parameter set may optionally contain video usability information (VUI), which includes parameters that may be important for buffering, picture output timing, rendering, and resource reservation. In HEVC a sequence parameter set RBSP includes parameters that can be referred to by one or more picture parameter set RBSPs or one or more SEI NAL units containing a buffering period SEI message. A picture parameter set contains such parameters that are likely to be unchanged in several coded pictures. A picture parameter set RBSP may include parameters that can be referred to by the coded slice NAL units of one or more coded pictures.

[0076] In HEVC, a video parameter set (VPS) may be defined as a syntax structure containing syntax elements that apply to zero or more entire coded video sequences as determined by the content of a syntax element found in the SPS referred to by a syntax element found in the PPS referred to by a syntax element found in each slice segment header. A video parameter set RBSP may include parameters that can be referred to by one or more sequence parameter set RBSPs.

[0077] The relationship and hierarchy between video parameter set (VPS), sequence parameter set (SPS), and picture parameter set (PPS) may be described as follows. VPS resides one level above SPS in the parameter set hierarchy and in the context of scalability and/or 3D video. VPS may include parameters that are common for all slices across all (scalability or view) layers in the entire coded video sequence. SPS includes the parameters that are common for all slices in a particular (scalability or view) layer in the entire coded video sequence, and may be shared by multiple (scalability or view) layers. PPS includes the parameters that are common for all slices in a particular layer representation (the representation of one scalability or view layer in one access unit) and are likely to be shared by all slices in multiple layer representations.

[0078] VPS may provide information about the dependency relationships of the layers in a bitstream, as well as many other information that are applicable to all slices across all (scalability or view) layers in the entire coded video sequence. VPS may be considered to comprise two parts, the base VPS and a VPS extension, where the VPS extension may be optionally present.

[0079] A SEI NAL unit may contain one or more SEI messages, which are not required for the decoding of output pictures but may assist in related processes, such as picture output timing, rendering, error detection, error concealment, and resource reservation. Several SEI messages are specified in H.264/AVC and HEVC, and the user data SEI messages enable organizations and companies to specify SEI messages for their own use. H.264/AVC and HEVC contain the syntax and semantics for the specified SEI messages but no process for handling the messages in the recipient is defined. Consequently, encoders are required to follow the H.264/AVC standard or the HEVC standard when they create SEI messages, and decoders conforming to the H.264/AVC standard or the HEVC standard, respectively, are not required to process SEI messages for output order conformance. One of the reasons to include the syntax and semantics of SEI messages in H.264/AVC and HEVC is to allow different system specifications to interpret the supplemental information identically and hence interoperate. It is intended that system specifications can require the use of particular SEI messages both in the encoding end and in the decoding end, and additionally the process for handling particular SEI messages in the recipient can be specified.

[0080] In HEVC, there are two types of SEI NAL units, namely the suffix SEI NAL unit and the prefix SEI NAL unit, having a different nal_unit_type value from each other. The SEI message(s) contained in a suffix SEI NAL unit are associated with the VCL NAL unit preceding, in decoding order, the suffix SEI NAL unit. The SEI message(s) contained in a prefix SEI NAL unit are associated with the VCL NAL unit following, in decoding order, the prefix SEI NAL unit.

[0081] In HEVC, a coded picture may be defined as a coded representation of a picture containing all coding tree units of the picture. In HEVC, an access unit (AU) may be defined as a set of NAL units that are associated with each other according to a specified classification rule, are consecutive in decoding order, and contain at most one picture with any specific value of nuh_layer_id. In addition to containing the VCL NAL units of the coded picture, an access unit may also contain non-VCL NAL units.

[0082] It may be required that coded pictures appear in certain order within an access unit. For example a coded picture with nuh_layer_id equal to nuhLayerIdA may be required to precede, in decoding order, all coded pictures with nuh_layer_id greater than nuhLayerIdA in the same access unit. An AU typically contains all the coded pictures that represent the same output time and/or capturing time.

[0083] A bitstream may be defined as a sequence of bits, in the form of a NAL unit stream or a byte stream, that forms the representation of coded pictures and associated data forming one or more coded video sequences. A first bitstream may be followed by a second bitstream in the same logical channel, such as in the same file or in the same connection of a communication protocol. An elementary stream (in the context of video coding) may be defined as a sequence of one or more bitstreams. The end of the first bitstream may be indicated by a specific NAL unit, which may be referred to as the end of bitstream (EOB) NAL unit and which is the last NAL unit of the bitstream.

[0084] A byte stream format has been specified in H.264/AVC and HEVC for transmission or storage environments that do not provide framing structures. The byte stream format separates NAL units from each other by attaching a start code in front of each NAL unit. To avoid false detection of NAL unit boundaries, encoders run a byte-oriented start code emulation prevention algorithm, which adds an emulation prevention byte to the NAL unit payload if a start code would have occurred otherwise. In order to, for example, enable straightforward gateway operation between packet- and stream-oriented systems, start code emulation prevention may always be performed regardless of whether the byte stream format is in use or not. The bit order for the byte stream format may be specified to start with the most significant bit (MSB) of the first byte, proceed to the least significant bit (LSB) of the first byte, followed by the MSB of the second byte, etc. The byte stream format may be considered to consist of a sequence of byte stream NAL unit syntax structures. Each byte stream NAL unit syntax structure may be considered to comprise one start code prefix followed by one NAL unit syntax structure, as well as trailing and/or heading padding bits and/or bytes.

[0085] A motion-constrained tile set (MCTS) is such a set of one or more tiles that the inter prediction process is constrained in encoding such that no sample value outside the motion-constrained tile set, and no sample value at a fractional sample position that is derived using one or more sample values outside the motion-constrained tile set, is used for inter prediction of any sample within the motion-constrained tile set. An MCTS may be required to be rectangular. Additionally, the encoding of an MCTS is constrained in a manner that motion vector candidates are not derived from blocks outside the MCTS. This may be enforced by turning off temporal motion vector prediction of HEVC, or by disallowing the encoder to use the TMVP candidate or any motion vector prediction candidate following the TMVP candidate in the merge or AMVP candidate list for PUs located directly left of the right tile boundary of the MCTS except the last one at the bottom right of the MCTS.

[0086] Note that sample locations used in inter prediction may be saturated so that a location that would be outside the picture otherwise is saturated to point to the corresponding boundary sample of the picture. Hence, if a tile boundary is also a picture boundary, motion vectors may effectively cross that boundary or a motion vector may effectively cause fractional sample interpolation that would refer to a location outside that boundary, since the sample locations are saturated onto the boundary. However, if tiles may be re-located in a tile merging operation (see e.g. embodiments of the present invention), encoders generating MCTSs may apply motion constraints to all tile boundaries of the MCTS, including picture boundaries.

[0087] The temporal motion-constrained tile sets SEI message of HEVC can be used to indicate the presence of motion-constrained tile sets in the bitstream.

[0088] A motion-constrained picture is such that the inter prediction process is constrained in encoding such that no sample value outside the picture, and no sample value at a fractional sample position that is derived using one or more sample values outside the picture, would be used for inter prediction of any sample within the picture and/or sample locations used for prediction need not be saturated to be within picture boundaries.

[0089] It may be considered that in stereoscopic or two-view video, one video sequence or view is presented for the left eye while a parallel view is presented for the right eye. More than two parallel views may be needed for applications which enable viewpoint switching or for autostereoscopic displays which may present a large number of views simultaneously and let the viewers to observe the content from different viewpoints.

[0090] A view may be defined as a sequence of pictures representing one camera or viewpoint. The pictures representing a view may also be called view components. In other words, a view component may be defined as a coded representation of a view in a single access unit. In multiview video coding, more than one view is coded in a bitstream. Since views are typically intended to be displayed on stereoscopic or multiview autostrereoscopic display or to be used for other 3D arrangements, they typically represent the same scene and are content-wise partly overlapping although representing different viewpoints to the content. Hence, inter-view prediction may be utilized in multiview video coding to take advantage of inter-view correlation and improve compression efficiency. One way to realize inter-view prediction is to include one or more decoded pictures of one or more other views in the reference picture list(s) of a picture being coded or decoded residing within a first view. View scalability may refer to such multiview video coding or multiview video bitstreams, which enable removal or omission of one or more coded views, while the resulting bitstream remains conforming and represents video with a smaller number of views than originally.

[0091] Frame packing may be defined to comprise arranging more than one input picture, which may be referred to as (input) constituent frames, into an output picture. In general, frame packing is not limited to any particular type of constituent frames or the constituent frames need not have a particular relation with each other. In many cases, frame packing is used for arranging constituent frames of a stereoscopic video clip into a single picture sequence, as explained in more details in the next paragraph. The arranging may include placing the input pictures in spatially non-overlapping areas within the output picture. For example, in a side-by-side arrangement, two input pictures are placed within an output picture horizontally adjacently to each other. The arranging may also include partitioning of one or more input pictures into two or more constituent frame partitions and placing the constituent frame partitions in spatially non-overlapping areas within the output picture. The output picture or a sequence of frame-packed output pictures may be encoded into a bitstream e.g. by a video encoder. The bitstream may be decoded e.g. by a video decoder. The decoder or a post-processing operation after decoding may extract the decoded constituent frames from the decoded picture(s) e.g. for displaying.

[0092] In frame-compatible stereoscopic video (a.k.a. frame packing of stereoscopic video), a spatial packing of a stereo pair into a single frame is performed at the encoder side as a pre-processing step for encoding and then the frame-packed frames are encoded with a conventional 2D video coding scheme. The output frames produced by the decoder contain constituent frames of a stereo pair.

[0093] In a typical operation mode, the spatial resolution of the original frames of each view and the packaged single frame have the same resolution. In this case the encoder downsamples the two views of the stereoscopic video before the packing operation. The spatial packing may use for example a side-by-side or top-bottom format, and the downsampling should be performed accordingly.

[0094] A uniform resource identifier (URI) may be defined as a string of characters used to identify a name of a resource. Such identification enables interaction with representations of the resource over a network, using specific protocols. A URI is defined through a scheme specifying a concrete syntax and associated protocol for the URI. The uniform resource locator (URL) and the uniform resource name (URN) are forms of URI. A URL may be defined as a URI that identifies a web resource and specifies the means of acting upon or obtaining the representation of the resource, specifying both its primary access mechanism and network location. A URN may be defined as a URI that identifies a resource by name in a particular namespace. A URN may be used for identifying a resource without implying its location or how to access it. The term requesting locator may be defined to an identifier that can be used to request a resource, such as a file or a segment. A requesting locator may, for example, be a URL or specifically an HTTP URL. A client may use a requesting locator with a communication protocol, such as HTTP, to request a resource from a server or a sender.

[0095] Available media file format standards include ISO base media file format (ISO/IEC 14496-12, which may be abbreviated ISOBMFF), MPEG-4 file format (ISO/IEC 14496-14, also known as the MP4 format), file format for NAL unit structured video (ISO/IEC 14496-15) and 3GPP file format (3GPP TS 26.244, also known as the 3GP format). ISOBMFF is the base for derivation of all the above mentioned file formats (excluding the ISOBMFF itself).

[0096] Some concepts, structures, and specifications of ISOBMFF are described below as an example of a container file format, based on which the embodiments may be implemented. The aspects of the invention are not limited to ISOBMFF, but rather the description is given for one possible basis on top of which the invention may be partly or fully realized.

[0097] A basic building block in the ISO base media file format is called a box. Each box has a header and a payload. The box header indicates the type of the box and the size of the box in terms of bytes. A box may enclose other boxes, and the ISO file format specifies which box types are allowed within a box of a certain type. Furthermore, the presence of some boxes may be mandatory in each file, while the presence of other boxes may be optional. Additionally, for some box types, it may be allowable to have more than one box present in a file. Thus, the ISO base media file format may be considered to specify a hierarchical structure of boxes.

[0098] According to the ISO family of file formats, a file includes media data and metadata that are encapsulated into boxes. Each box is identified by a four character code (4CC) and starts with a header which informs about the type and size of the box.

[0099] In files conforming to the ISO base media file format, the media data may be provided in a media data 'mdat' box and the movie 'moov' box may be used to enclose the metadata. In some cases, for a file to be operable, both of the 'mdat' and 'moov' boxes may be required to be present. The movie 'moov' box may include one or more tracks, and each track may reside in one corresponding track 'trak' box. A track may be one of the many types, including a media track that refers to samples formatted according to a media compression format (and its encapsulation to the ISO base media file format). A track may be regarded as a logical channel.

[0100] Movie fragments may be used e.g. when recording content to ISO files e.g. in order to avoid losing data if a recording application crashes, runs out of memory space, or some other incident occurs. Without movie fragments, data loss may occur because the file format may require that all metadata, e.g., the movie box, be written in one contiguous area of the file. Furthermore, when recording a file, there may not be sufficient amount of memory space (e.g., random access memory RAM) to buffer a movie box for the size of the storage available, and re-computing the contents of a movie box when the movie is closed may be too slow. Moreover, movie fragments may enable simultaneous recording and playback of a file using a regular ISO file parser. Furthermore, a smaller duration of initial buffering may be required for progressive downloading, e.g., simultaneous reception and playback of a file when movie fragments are used and the initial movie box is smaller compared to a file with the same media content but structured without movie fragments.

[0101] The movie fragment feature may enable splitting the metadata that otherwise might reside in the movie box into multiple pieces. Each piece may correspond to a certain period of time of a track. In other words, the movie fragment feature may enable interleaving file metadata and media data. Consequently, the size of the movie box may be limited and the use cases mentioned above be realized.

[0102] In some examples, the media samples for the movie fragments may reside in an mdat box. For the metadata of the movie fragments, however, a moof box may be provided. The moof box may include the information for a certain duration of playback time that would previously have been in the moov box. The moov box may still represent a valid movie on its own, but in addition, it may include an mvex box indicating that movie fragments will follow in the same file. The movie fragments may extend the presentation that is associated to the moov box in time.

[0103] Within the movie fragment there may be a set of track fragments, including anywhere from zero to a plurality per track. The track fragments may in turn include anywhere from zero to a plurality of track runs, each of which document is a contiguous run of samples for that track (and hence are similar to chunks). Within these structures, many fields are optional and can be defaulted. The metadata that may be included in the moof box may be limited to a subset of the metadata that may be included in a moov box and may be coded differently in some cases. Details regarding the boxes that can be included in a moof box may be found from the ISOBMFF specification. A self-contained movie fragment may be defined to consist of a moof box and an mdat box that are consecutive in the file order and where the mdat box contains the samples of the movie fragment (for which the moof box provides the metadata) and does not contain samples of any other movie fragment (i.e. any other moof box).

[0104] The track reference mechanism can be used to associate tracks with each other. The TrackReferenceBox includes box(es), each of which provides a reference from the containing track to a set of other tracks. These references are labeled through the box type (i.e. the four-character code of the box) of the contained box(es).

[0105] The ISO Base Media File Format contains three mechanisms for timed metadata that can be associated with particular samples: sample groups, timed metadata tracks, and sample auxiliary information. Derived specification may provide similar functionality with one or more of these three mechanisms.

[0106] A sample grouping in the ISO base media file format and its derivatives, such as the AVC file format and the SVC file format, may be defined as an assignment of each sample in a track to be a member of one sample group, based on a grouping criterion. A sample group in a sample grouping is not limited to being contiguous samples and may contain non-adjacent samples. As there may be more than one sample grouping for the samples in a track, each sample grouping may have a type field to indicate the type of grouping. Sample groupings may be represented by two linked data structures: (1) a SampleToGroupBox (sbgp box) represents the assignment of samples to sample groups; and (2) a SampleGroupDescriptionBox (sgpd box) contains a sample group entry for each sample group describing the properties of the group. There may be multiple instances of the SampleToGroupBox and SampleGroupDescriptionBox based on different grouping criteria. These may be distinguished by a type field used to indicate the type of grouping. SampleToGroupBox may comprise a grouping_type_parameter field that can be used e.g. to indicate a sub-type of the grouping.

[0107] The restricted video ('resv') sample entry and mechanism has been specified for the ISOBMFF in order to handle situations where the file author requires certain actions on the player or renderer after decoding of a visual track. Players not recognizing or not capable of processing the required actions are stopped from decoding or rendering the restricted video tracks. The 'resv' sample entry mechanism applies to any type of video codec. A RestrictedSchemeInfoBox is present in the sample entry of 'resv' tracks and comprises a OriginalFormatBox, SchemeTypeBox, and SchemeInformationBox. The original sample entry type that would have been unless the 'resv' sample entry type were used is contained in the OriginalFormatBox. The SchemeTypeBox provides an indication which type of processing is required in the player to process the video. The SchemeInformationBox comprises further information of the required processing. The scheme type may impose requirements on the contents of the SchemeInformationBox. For example, the stereo video scheme indicated in the SchemeTypeBox indicates that when decoded frames either contain a representation of two spatially packed constituent frames that form a stereo pair (frame packing) or only one view of a stereo pair (left and right views in different tracks). StereoVideoBox may be contained in SchemeInformationBox to provide further information e.g. on which type of frame packing arrangement has been used (e.g. side-by-side or top-bottom).

[0108] The Matroska file format is capable of (but not limited to) storing any of video, audio, picture, or subtitle tracks in one file. Matroska may be used as a basis format for derived file formats, such as WebM. Matroska uses Extensible Binary Meta Language (EBML) as basis. EBML specifies a binary and octet (byte) aligned format inspired by the principle of XML. EBML itself is a generalized description of the technique of binary markup. A Matroska file consists of Elements that make up an EBML "document." Elements incorporate an Element ID, a descriptor for the size of the element, and the binary data itself. Elements can be nested. A Segment Element of Matroska is a container for other top-level (level 1) elements. A Matroska file may comprise (but is not limited to be composed of) one

Segment. Multimedia data in Matroska files is organized in Clusters (or Cluster Elements), each containing typically a few seconds of multimedia data. A Cluster comprises BlockGroup elements, which in turn comprise Block Elements. A Cues Element comprises metadata which may assist in random access or seeking and may include file pointers or respective timestamps for seek points.

[0109] Several commercial solutions for adaptive streaming over HTTP, such as Microsoft® Smooth Streaming, Apple® Adaptive HTTP Live Streaming and Adobe® Dynamic Streaming, have been launched as well as standardization projects have been carried out. Adaptive HTTP streaming (AHS) was first standardized in Release 9 of 3rd Generation Partnership Project (3GPP) packet-switched streaming (PSS) service (3GPP TS 26.234 Release 9: "Transparent end-to-end packet-switched streaming service (PSS); protocols and codecs"). MPEG took 3GPP AHS Release 9 as a starting point for the MPEG DASH standard (ISO/IEC 23009-1: "Dynamic adaptive streaming over HTTP (DASH)-Part 1: Media presentation description and segment formats," International Standard, 2^{nd} Edition, 2014). MPEG DASH and 3GP-DASH are technically close to each other and may therefore be collectively referred to as DASH. Some concepts, formats, and operations of DASH are described below as an example of a video streaming system, wherein the embodiments may be implemented. The aspects of the invention are not limited to DASH, but rather the description is given for one possible basis on top of which the invention may be partly or fully realized.

[0110] In DASH, the multimedia content may be stored on an HTTP server and may be delivered using HTTP. The content may be stored on the server in two parts: Media Presentation Description (MPD), which describes a manifest of the available content, its various alternatives, their URL addresses, and other characteristics; and segments, which contain the actual multimedia bitstreams in the form of chunks, in a single or multiple files. The MDP provides the necessary information for clients to establish a dynamic adaptive streaming over HTTP. The MPD contains information describing media presentation, such as an HTTP- uniform resource locator (URL) of each Segment to make GET Segment request. To play the content, the DASH client may obtain the MPD e.g. by using HTTP, email, thumb drive, broadcast, or other transport methods. By parsing the MPD, the DASH client may become aware of the program timing, media-content availability, media types, resolutions, minimum and maximum bandwidths, and the existence of various encoded alternatives of multimedia components, accessibility features and required digital rights management (DRM), media-component locations on the network, and other content characteristics. Using this information, the DASH client may select the appropriate encoded alternative and start streaming the content by fetching the segments using e.g. HTTP GET requests. After appropriate buffering to allow for network throughput variations, the client may continue fetching the subsequent segments and also monitor the network bandwidth fluctuations. The client may decide how to adapt to the available bandwidth by fetching segments of different alternatives (with lower or higher bitrates) to maintain an adequate buffer.

[0111] In DASH, hierarchical data model is used to structure media presentation as shown in Figure 7. A media presentation consists of a sequence of one or more Periods, each Period contains one or more Groups, each Group contains one or more Adaptation Sets, each Adaptation Sets contains one or more Representations, each Representation consists of one or more Segments. A Representation is one of the alternative choices of the media content or a subset thereof typically differing by the encoding choice, e.g. by bitrate, resolution, language, codec, etc. The Segment contains certain duration of media data, and metadata to decode and present the included media content. A Segment is identified by a URI and can typically be requested by a HTTP GET request. A Segment may be defined as a unit of data associated with an HTTP-URL and optionally a byte range that are specified by an MPD.

[0112] The DASH MPD complies with Extensible Markup Language (XML) and is therefore specified through elements and attribute as defined in XML. The MPD may be specified using the following conventions: Elements in an XML document may be identified by an upper-case first letter and may appear in bold face as **Element**. To express that an element **Element1** is contained in another element **Element2**, one may write **Element2.Element1**. If an element's name consists of two or more combined words, camel-casing may be used, e.g. **ImportantElement**. Elements may be present either exactly once, or the minimum and maximum occurrence may be defined by <minOccurs> ... <maxOccurs>. Attributes in an XML document may be identified by a lower-case first letter as well as they may be preceded by a '@'-sign, e.g. @attribute. To point to a specific attribute @attribute contained in an element **Element**, one may write **Element**@attribute. If an attribute's name consists of two or more combined words, camel-casing may be used after the first word, e.g. @veryImportantAttribute. Attributes may have assigned a status in the XML as mandatory (M), optional (O), optional with default value (OD) and conditionally mandatory (CM).

[0113] In DASH, an independent representation may be defined as a representation that can be processed independently of any other representations. An independent representation may be understood to comprise an independent bitstream or an independent layer of a bitstream. A dependent representation may be defined as a representation for which Segments from its complementary representations are necessary for presentation and/or decoding of the contained media content components. A dependent representation may be understood to comprise e.g. a predicted layer of a scalable bitstream. A complementary representation may be defined as a representation which complements at least one dependent representation. A complementary representation may be an independent representation or a dependent representation. Dependent Representations may be described by a **Representation** element that contains a @dependencyId attribute. Dependent Representations can be regarded as regular Representations except that they depend on a set of complementary Representations for decoding and/or presentation. The @dependencyId contains the values of the @id attribute of all the complementary Representations, i.e. Representations that are necessary to present and/or decode the media content components contained in this dependent Representation.

[0114] In the context of DASH, the following definitions may be used: A media content component or a media component may be defined as one continuous component of the media content with an assigned media component type that can be encoded individually into a media stream. Media content may be defined as one media content period or a contiguous sequence of media content periods. Media content component type may be defined as a single type of media content such as audio, video, or text. A media stream may be defined as an encoded version of a media content component.

[0115] An Initialization Segment may be defined as a Segment containing metadata that is necessary to present the media streams encapsulated in Media Segments. In ISOBMFF based segment formats, an Initialization Segment may comprise the Movie Box ('moov') which might not include metadata for any samples, i.e. any metadata for samples is provided in 'moof' boxes.

[0116] A Media Segment contains certain duration of media data for playback at a normal speed, such duration is referred as Media Segment duration or Segment duration. The content producer or service provider may select the Segment duration according to the desired characteristics of the service. For example, a relatively short Segment duration may be used in a live service to achieve a short end-to-end latency. The reason is that Segment duration is typically a lower bound on the end-to-end latency perceived by a DASH client since a Segment is a discrete unit of generating media data for DASH. Content generation is typically done such a manner that a whole Segment of media data is made available for a server. Furthermore, many client implementations use a Segment as the unit for GET requests. Thus, in typical arrangements for live services a Segment can be requested by a DASH client only when the whole duration of Media Segment is available as well as encoded and encapsulated into a Segment. For on-demand service, different strategies of selecting Segment duration may be used.

[0117] DASH supports rate adaptation by dynamically requesting Media Segments from different Representations within an Adaptation Set to match varying network bandwidth. When a DASH client switches up/down Representation, coding dependencies within Representation have to be taken into account. A Representation switch may only happen at a random access point (RAP), which is typically used in video coding techniques such as H.264/AVC. In DASH, a more general concept named Stream Access Point (SAP) is introduced to provide a codec-independent solution for accessing a Representation and switching between Representations. In DASH, a SAP is specified as a position in a Representation that enables playback of a media stream to be started using only the information contained in Representation data starting from that position onwards (preceded by initialising data in the Initialisation Segment, if any). Hence, Representation switching can be performed in SAP.

[0118] Several types of SAP have been specified, including the following. SAP Type 1 corresponds to what is known in some coding schemes as a “Closed GOP random access point” (in which all pictures, in decoding order, can be correctly decoded, resulting in a continuous time sequence of correctly decoded pictures with no gaps) and in addition the first picture in decoding order is also the first picture in presentation order. SAP Type 2 corresponds to what is known in some coding schemes as a “Closed GOP random access point” (in which all pictures, in decoding order, can be correctly decoded, resulting in a continuous time sequence of correctly decoded pictures with no gaps), for which the first picture in decoding order may not be the first picture in presentation order. SAP Type 3 corresponds to what is known in some coding schemes as an “Open GOP random access point”, in which there may be some pictures in decoding order that cannot be correctly decoded and have presentation times less than intra-coded picture associated with the SAP.

[0119] A stream access point (SAP) sample group as specified in ISOBMFF identifies samples as being of the indicated SAP type. The grouping_type_parameter for the SAP sample group comprises the fields target_layers and layer_id_method_idc. target_layers specifies the target layers for the indicated SAPs. The semantics of target_layers may depend on the value of layer_id_method_idc, which specifies the semantics of target_layers. layer_id_method_idc equal to 0 specifies that the target layers consist of all the layers represented by the track. The sample group description entry for the SAP sample group comprises the fields dependent_flag and SAP_type. dependent_flag may be required to be 0 for non-layered media. dependent_flag equal to 1 specifies that the reference layers, if any, for predicting the target layers may have to be decoded for accessing a sample of this sample group. dependent_flag equal to 0 specifies that the reference layers, if any, for predicting the target layers need not be decoded for accessing any SAP of this sample group. sap_type values in the range of 1 to 6, inclusive, specify the SAP type, of the associated samples.

[0120] A sync sample may be defined as a sample in a track that is of a SAP of type 1 or 2. Sync samples may be indicated with SyncSampleBox or by sample_is_non_sync_sample equal to 0 in the signaling for track fragments.

[0121] A Segment may further be partitioned into Subsegments e.g. to enable downloading segments in multiple parts. Subsegments may be required to contain complete access units. Subsegments may be indexed by Segment Index box, which contains information to map presentation time range and byte range for each Subsegment. The Segment Index box may also describe subsegments and stream access points in the segment by signaling their durations and byte offsets. A DASH client may use the information obtained from Segment Index box(es) to make a HTTP GET request for a specific Subsegment using byte range HTTP request. If relatively long Segment duration is used, then Subsegments may be used to keep the size of HTTP responses reasonable and flexible for bitrate adaptation. The indexing information of a segment may be put in the single box at the beginning of that segment, or spread among many indexing boxes in the segment. Different methods of spreading are possible, such as hierarchical, daisy chain, and hybrid. This technique may avoid adding a large box at the beginning of the segment and therefore may prevent a possible initial download delay.

[0122] It may be required that for any dependent Representation X that depends on complementary Representation Y, the *m*-th Subsegment of X and the *n*-th Subsegment of Y shall be non-overlapping whenever *m* is not equal to *n*. It may be required that for dependent Representations the concatenation of the Initialization Segment with the sequence of Subsegments of the dependent Representations, each being preceded by the corresponding Subsegment of each of the complementary Representations in order as provided in the @dependencyId attribute shall represent a conforming Subsegment sequence conforming to the media format as specified in the @mimeType attribute for this dependent Representation.

[0123] MPEG-DASH defines segment-container formats for both ISOBMFF and MPEG-2 Transport Streams. Other specifications may specify segment formats based on other container formats. For example, a segment format based on Matroska container file format has been proposed and may be summarized as follows. When Matroska files are carried as DASH segments or alike, the association of DASH units and Matroska units may be specified as follows. A subsegment (of DASH) may be are defined as one or more consecutive Clusters of Matroska-encapsulated content. An Initialization Segment of DASH may be required to comprise the EBML header, Segment header (of Matroska), Segment Information (of Matroska) and Tracks, and may optionally comprise other level1 elements and padding. A Segment Index of DASH may comprise a Cues Element of Matroska.

[0124] Video codec may comprise an encoder that transforms the input video into a compressed representation suited for storage/transmission and a decoder that can uncompress the compressed video representation back into a viewable form. A video encoder and/or a video decoder may also be separate from each other, i.e. need not form a codec. Typically encoder discards some information in the original video sequence in order to represent the video in a more compact form (that is, at lower bitrate). A video encoder may be used to encode an image sequence, as defined subsequently, and a video decoder may be used to decode a coded image sequence. A video encoder or an intra coding part of a video encoder or an image encoder may be used to encode an image, and a video decoder or an inter decoding part of a video decoder or an image decoder may be used to decode a coded image.

[0125] Some hybrid video encoders, for example many encoder implementations of ITU-T H.263 and H.264, encode the video information in two phases. Firstly pixel values in a certain picture area (or "block") are predicted for example by motion compensation means (finding and indicating an area in one of the previously coded video frames that corresponds closely to the block being coded) or by spatial means (using the pixel values around the block to be coded in a specified manner). Secondly the prediction error, i.e. the difference between the predicted block of pixels and the original block of pixels, is coded. This is typically done by transforming the difference in pixel values using a specified transform (e.g. Discrete Cosine Transform (DCT) or a variant of it), quantizing the coefficients and entropy coding the quantized coefficients. By varying the fidelity of the quantization process, encoder can control the balance between the accuracy of the pixel representation (picture quality) and size of the resulting coded video representation (file size or transmission bitrate).

[0126] In temporal prediction, the sources of prediction are previously decoded pictures (a.k.a. reference pictures). In intra block copy (a.k.a. intra-block-copy prediction), prediction is applied similarly to temporal prediction but the reference picture is the current picture and only previously decoded samples can be referred in the prediction process. Inter-layer or inter-view prediction may be applied similarly to temporal prediction, but the reference picture is a decoded picture from another scalable layer or from another view, respectively. In some cases, inter prediction may refer to temporal prediction only, while in other cases inter prediction may refer collectively to temporal prediction and any of intra block copy, inter-layer prediction, and inter-view prediction provided that they are performed with the same or similar process than temporal prediction. Inter prediction or temporal prediction may sometimes be referred to as motion compensation or motion-compensated prediction.

[0127] Intra prediction utilizes the fact that adjacent pixels within the same picture are likely to be correlated. Intra prediction can be performed in spatial or transform domain, i.e., either sample values or transform coefficients can be predicted. Intra prediction is typically exploited in intra coding, where no inter prediction is applied.

[0128] There may be different types of intra prediction modes available in a coding scheme, out of which an encoder can select and indicate the used one, e.g. on block or coding unit basis. A decoder may decode the indicated intra prediction mode and reconstruct the prediction block accordingly. For example, several angular intra prediction modes, each for different angular direction, may be available. Angular intra prediction may be considered to extrapolate the border samples of adjacent blocks along a linear prediction direction. Additionally or alternatively, a planar prediction mode may be available. Planar prediction may be considered to essentially form a prediction block, in which each sample of a prediction block may be specified to be an average of vertically aligned sample in the adjacent sample column on the left of the current block and the horizontally aligned sample in the adjacent sample line above the current block. Additionally or alternatively, a DC prediction mode may be available, in which the prediction block is essentially an average sample value of a neighboring block or blocks.

[0129] One outcome of the coding procedure is a set of coding parameters, such as motion vectors and quantized transform coefficients. Many parameters can be entropy-coded more efficiently if they are predicted first from spatially or temporally neighbouring parameters. For example, a motion vector may be predicted from spatially adjacent motion vectors and only the difference relative to the motion vector predictor may be coded. Prediction of coding parameters and intra prediction may be collectively referred to as in-picture prediction.

[0130] Figure 8a shows a block diagram of a video encoder suitable for employing embodiments of the invention. Figure 8a presents an encoder for two layers, but it would be appreciated that presented encoder could be similarly simplified to encode only one layer or extended to encode more than two layers. Figure 8a illustrates an embodiment of a video encoder comprising a first encoder section 500 for a base layer and a second encoder section

502 for an enhancement layer. Each of the first encoder section 500 and the second encoder section 502 may comprise similar elements for encoding incoming pictures. The encoder sections 500, 502 may comprise a pixel predictor 302, 402, prediction error encoder 303, 403 and prediction error decoder 304, 404. Figure 8a also shows an embodiment of the pixel predictor 302, 402 as comprising an inter-predictor 306, 406, an intra-predictor 308, 408, a mode selector 310, 410, a filter 316, 416, and a reference frame memory 318, 418. The pixel predictor 302 of the first encoder section 500 receives 300 base layer images of a video stream to be encoded at both the inter-predictor 306 (which determines the difference between the image and a motion compensated reference frame 318) and the intra-predictor 308 (which determines a prediction for an image block based only on the already processed parts of current frame or picture). The output of both the inter-predictor and the intra-predictor are passed to the mode selector 310. The intra-predictor 308 may have more than one intra-prediction modes. Hence, each mode may perform the intra-prediction and provide the predicted signal to the mode selector 310. The mode selector 310 also receives a copy of the base layer picture 300. Correspondingly, the pixel predictor 402 of the second encoder section 502 receives 400 enhancement layer images of a video stream to be encoded at both the inter-predictor 406 (which determines the difference between the image and a motion compensated reference frame 418) and the intra-predictor 408 (which determines a prediction for an image block based only on the already processed parts of current frame or picture). The output of both the inter-predictor and the intra-predictor are passed to the mode selector 410. The intra-predictor 408 may have more than one intra-prediction modes. Hence, each mode may perform the intra-prediction and provide the predicted signal to the mode selector 410. The mode selector 410 also receives a copy of the enhancement layer picture 400.

[0131] Depending on which encoding mode is selected to encode the current block, the output of the inter-predictor 306, 406 or the output of one of the optional intra-predictor modes or the output of a surface encoder within the mode selector is passed to the output of the mode selector 310, 410. The output of the mode selector is passed to a first summing device 321, 421. The first summing device may subtract the output of the pixel predictor 302, 402 from the base layer picture 300/enhancement layer picture 400 to produce a first prediction error signal 320, 420 which is input to the prediction error encoder 303, 403.

[0132] The pixel predictor 302, 402 further receives from a preliminary reconstructor 339, 439 the combination of the prediction representation of the image block 312, 412 and the output 338, 438 of the prediction error decoder 304, 404. The preliminary reconstructed image 314, 414 may be passed to the intra-predictor 308, 408 and to a filter 316, 416. The filter 316, 416 receiving the preliminary representation may filter the preliminary representation and output a final reconstructed image 340, 440 which may be saved in a reference frame memory 318, 418. The reference frame memory 318 may be connected to the inter-predictor 306 to be used as the reference image against which a future base layer picture 300 is compared in inter-prediction operations. Subject to the base layer being selected and indicated to be source for inter-layer sample prediction and/or inter-layer motion information prediction of the enhancement layer according to some embodiments, the reference frame memory 318 may also be connected to the inter-predictor 406 to be used as the reference image against which a future enhancement layer pictures 400 is compared in inter-prediction operations. Moreover, the reference frame memory 418 may be connected to the inter-predictor 406 to be used as the reference image against which a future enhancement layer picture 400 is compared in inter-prediction operations.

[0133] Filtering parameters from the filter 316 of the first encoder section 500 may be provided to the second encoder section 502 subject to the base layer being selected and indicated to be source for predicting the filtering parameters of the enhancement layer according to some embodiments.

[0134] The prediction error encoder 303, 403 comprises a transform unit 342, 442 and a quantizer 344, 444. The transform unit 342, 442 transforms the first prediction error signal 320, 420 to a transform domain. The transform is, for example, the DCT transform. The quantizer 344, 444 quantizes the transform domain signal, e.g. the DCT coefficients, to form quantized coefficients.

[0135] The prediction error decoder 304, 404 receives the output from the prediction error encoder 303, 403 and performs the opposite processes of the prediction error encoder 303, 403 to produce a decoded prediction error signal 338, 438 which, when combined with the prediction representation of the image block 312, 412 at the second summing device 339, 439, produces the preliminary reconstructed image 314, 414. The prediction error decoder may be considered to comprise a dequantizer 361, 461, which dequantizes the quantized coefficient values, e.g. DCT coefficients, to reconstruct the transform signal and an inverse transformation unit 363, 463, which performs the inverse transformation to the reconstructed transform signal wherein the output of the inverse transformation unit 363, 463 contains reconstructed block(s). The prediction error decoder may also comprise a block filter which may filter the reconstructed block(s) according to further decoded information and filter parameters.

[0136] The entropy encoder 330, 430 receives the output of the prediction error encoder 303, 403 and may perform a suitable entropy encoding/variable length encoding on the signal to provide error detection and correction capability. The outputs of the entropy encoders 330, 430 may be inserted into a bitstream e.g. by a multiplexer 508.

[0137] Figure 8b shows a block diagram of a video decoder suitable for employing embodiments of the invention. Figure 8b depicts a structure of a two-layer decoder, but it would be appreciated that the decoding operations may similarly be employed in a single-layer decoder.

[0138] The video decoder 550 comprises a first decoder section 552 for base layer pictures and a second decoder section 554 for enhancement layer pictures. Block 556 illustrates a demultiplexer for delivering information regarding base layer pictures to the first decoder section 552 and for delivering information regarding enhancement layer pictures to the second decoder section 554. Reference P'n stands for a predicted representation of an image block. Reference D'n stands for a reconstructed prediction error signal. Blocks 704, 804 illustrate preliminary reconstructed images (I'n). Reference R'n stands for a final reconstructed image. Blocks 703, 803 illustrate inverse transform (T-1). Blocks 702, 802 illustrate inverse quantization (Q-1). Blocks 700, 800 illustrate entropy decoding (E-1). Blocks 706, 806 illustrate a reference frame memory (RFM). Blocks 707, 807 illustrate prediction (P) (either inter prediction or intra prediction). Blocks 708, 808 illustrate filtering (F). Blocks 709, 809 may be used to combine decoded prediction error information with predicted base or enhancement layer pictures to obtain the preliminary reconstructed images (I'n). Preliminary reconstructed and filtered base layer pictures may be output 710 from the first decoder section 552 and preliminary reconstructed and filtered enhancement layer pictures may be output 810 from the second decoder section 554.

[0139] Herein, the decoder could be interpreted to cover any operational unit capable to carry out the decoding operations, such as a player, a receiver, a gateway, a demultiplexer and/or a decoder.

[0140] The decoder reconstructs the output video by applying prediction means similar to the encoder to form a predicted representation of the pixel blocks (using the motion or spatial information created by the encoder and stored in the compressed representation) and prediction error decoding (inverse operation of the prediction error coding recovering the quantized prediction error signal in spatial pixel domain). After applying prediction and prediction error decoding means the decoder sums up the prediction and prediction error signals (pixel values) to form the output video frame. The decoder (and encoder) can also apply additional filtering means to improve the quality of the output video before passing it for display and/or storing it as prediction reference for the forthcoming frames in the video sequence.

[0141] In typical video codecs the motion information is indicated with motion vectors associated with each motion compensated image block, such as a prediction unit. Each of these motion vectors represents the displacement of the image block in the picture to be coded (in the encoder side) or decoded (in the decoder side) and the prediction source block in one of the previously coded or decoded pictures. In order to represent motion vectors efficiently those are typically coded differentially with respect to block specific predicted motion vectors. In typical video codecs the predicted motion vectors are created in a predefined way, for example calculating the median of the encoded or decoded motion vectors of the adjacent blocks. Another way to create motion vector predictions is to generate a list of candidate predictions from adjacent blocks and/or co-located blocks in temporal reference pictures and signalling the chosen candidate as the motion vector predictor. In addition to predicting the motion vector values, it can be predicted which reference picture(s) are used for motion-compensated prediction and this prediction information may be represented for example by a reference index of previously coded/decoded picture. The reference index is typically predicted from adjacent blocks and/or co-located blocks in temporal reference picture. Moreover, typical high efficiency video codecs employ an additional motion information coding/decoding mechanism, often called merging/merge mode, where all the motion field information, which includes motion vector and corresponding reference picture index for each available reference picture list, is predicted and used without any modification/correction. Similarly, predicting the motion field information is carried out using the motion field information of adjacent blocks and/or co-located blocks in temporal reference pictures and the used motion field information is signalled among a list of motion field candidate list filled with motion field information of available adjacent/co-located blocks.

[0142] Typical video codecs enable the use of uni-prediction, where a single prediction block is used for a block being (de)coded, and bi-prediction, where two prediction blocks are combined to form the prediction for a block being (de)coded. Some video codecs enable weighted prediction, where the sample values of the prediction blocks are weighted prior to adding residual information. For example, multiplicative weighting factor and an additive offset which can be applied. In explicit weighted prediction, enabled by some video codecs, a weighting factor and offset may be coded for example in the slice header for each allowable reference picture index. In implicit weighted prediction, enabled by some video codecs, the weighting factors and/or offsets are not coded but are derived e.g. based on the relative picture order count (POC) distances of the reference pictures.

[0143] In typical video codecs the prediction residual after motion compensation is first transformed with a transform kernel (like DCT) and then coded. The reason for this is that often there still exists some correlation among the residual and transform can in many cases help reduce this correlation and provide more efficient coding.

[0144] Typical video encoders utilize Lagrangian cost functions to find optimal coding modes, e.g. the desired Macroblock mode and associated motion vectors. This kind of cost function uses a weighting factor λ to tie together the (exact or estimated) image distortion due to lossy coding methods and the (exact or estimated) amount of information that is required to represent the pixel values in an image area:

[0145] $$C = D + \lambda R \quad (1)$$

[0146] where C is the Lagrangian cost to be minimized, D is the image distortion (e.g. Mean Squared Error) with the mode and motion vectors considered, and R the number of bits needed to represent the required data to reconstruct the image block in the decoder (including the amount of data to represent the candidate motion vectors).

[0147] H.264/AVC and HEVC include a concept of picture order count (POC). A value of POC is derived for each picture and is non-decreasing with increasing picture position in output order. POC therefore indicates the output order of pictures. POC may be used in the decoding process, for example, for implicit scaling of motion vectors in the temporal direct mode of bi-predictive slices, for implicitly derived weights in weighted prediction, and for reference picture list initialization. Furthermore, POC may be used in the verification of output order conformance.

[0148] Video encoders and/or decoders may be able to store multiple reference pictures in a decoded picture buffer (DPB) and use them adaptively for inter prediction. The reference picture management may be defined as a process to determine which reference pictures are maintained in the DPB. Examples of reference picture management are described in the following.

[0149] In HEVC, a reference picture set (RPS) syntax structure and decoding process are used. A reference picture set valid or active for a picture includes all the reference pictures used as reference for the picture and all the reference pictures that are kept marked as “used for reference” for any subsequent pictures in decoding order. There are six subsets of the reference picture set, which are referred to as namely RefPicSetStCurr0 (a.k.a.

RefPicSetStCurrBefore), RefPicSetStCurr1 (a.k.a. RefPicSetStCurrAfter), RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, and RefPicSetLtFoll. RefPicSetStFoll0 and RefPicSetStFoll1 may also be considered to form jointly one subset RefPicSetStFoll. The notation of the six subsets is as follows. “Curr” refers to reference pictures that are included in the reference picture lists of the current picture and hence may be used as inter prediction reference for the current picture. “Foll” refers to reference pictures that are not included in the reference picture lists of the current picture but may be used in subsequent pictures in decoding order as reference pictures. “St” refers to short-term reference pictures, which may generally be identified through a certain number of least significant bits of their POC value. “Lt” refers to long-term reference pictures, which are specifically identified and generally have a greater difference of POC values relative to the current picture than what can be represented by the mentioned certain number of least significant bits. “0” refers to those reference pictures that have a smaller POC value than that of the current picture. “1” refers to those reference pictures that have a greater POC value than that of the current picture. RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0 and RefPicSetStFoll1 are collectively referred to as the short-term subset of the reference picture set. RefPicSetLtCurr and RefPicSetLtFoll are collectively referred to as the long-term subset of the reference picture set.

[0150] In HEVC, a reference picture set may be specified in a sequence parameter set and taken into use in the slice header through an index to the reference picture set. A reference picture set may also be specified in a slice header. A reference picture set may be coded independently or may be predicted from another reference picture set (known as inter-RPS prediction). In both types of reference picture set coding, a flag (used_by_curr_pic_X_flag) is additionally sent for each reference picture indicating whether the reference picture is used for reference by the current picture (included in a *Curr list) or not (included in a *Foll list). Pictures that are included in the reference picture set used by the current slice are marked as “used for reference”, and pictures that are not in the reference picture set used by the current slice are marked as “unused for reference”. If the current picture is an IDR picture, RefPicSetStCurr0, RefPicSetStCurr1, RefPicSetStFoll0, RefPicSetStFoll1, RefPicSetLtCurr, and RefPicSetLtFoll are all set to empty.

[0151] In many coding modes of H.264/AVC and HEVC, the reference picture for inter prediction is indicated with an index to a reference picture list. The index may be coded with variable length coding, which usually causes a smaller index to have a shorter value for the corresponding syntax element. In H.264/AVC and HEVC, two reference picture lists (reference picture list 0 and reference picture list 1) are generated for each bi-predictive (B) slice, and one reference picture list (reference picture list 0) is formed for each inter-coded (P) slice.

[0152] A reference picture list, such as reference picture list 0 and reference picture list 1, is typically constructed in two steps: First, an initial reference picture list is generated. The initial reference picture list may be generated for example on the basis of POC, or information on the prediction hierarchy, or any combination thereof. Second, the initial reference picture list may be reordered by reference picture list reordering (RPLR) commands, also known as reference picture list modification syntax structure, which may be contained in slice headers. If reference picture sets are used, the reference picture list 0 may be initialized to contain RefPicSetStCurr0 first, followed by RefPicSetStCurr1, followed by RefPicSetLtCurr. Reference picture list 1 may be initialized to contain RefPicSetStCurr1 first, followed by RefPicSetStCurr0. In HEVC, the initial reference picture lists may be modified through the reference picture list modification syntax structure, where pictures in the initial reference picture lists may be identified through an entry index to the list. In other words, in HEVC, reference picture list modification is encoded into a syntax structure comprising a loop over each entry in the final reference picture list, where each loop entry is a fixed-length coded index to the initial reference picture list and indicates the picture in ascending position order in the final reference picture list.

[0153] Many coding standards, including H.264/AVC and HEVC, may have decoding process to derive a reference picture index to a reference picture list, which may be used to indicate which one of the multiple reference pictures is used for inter prediction for a particular block. A reference picture index may be coded by an encoder into the bitstream in some inter coding modes or it may be derived (by an encoder and a decoder) for example using neighboring blocks in some other inter coding modes.

[0154] Figures 1a and 1b illustrate an example of a camera having multiple lenses and imaging sensors but also other types of cameras may be used to capture wide view images and/or wide view video.

[0155] In the following, the terms wide view image and wide view video mean an image and a video, respectively, which comprise visual information having a relatively large viewing angle, larger than 100 degrees. Hence, a so called 360 panorama image/video as well as images/videos captured by using a fish eye lens may also be called as a wide view image/video in this specification. More generally, the wide view image/video may mean an image/video in which some kind of projection distortion may occur when a direction of view changes between successive images or frames of the video so that a transform may be needed to find out co-located pixels from a reference image or a reference frame. This will be described in more detail later in this specification.

[0156] The camera 100 of Figure 1a comprises two or more camera units 102 and is capable of capturing wide view images and/or wide view video. In this example the number of camera units 102 is eight, but may also be less than eight or more than eight. Each camera unit 102 is located at a different location in the multi-camera system and may have a different orientation with respect to other camera units 102. As an example, the camera units 102 may have an omnidirectional constellation so that it has a 360 viewing angle in a 3D-space. In other words, such camera 100 may be able to see each direction of a scene so that each spot of the scene around the camera 100 can be viewed by at least one camera unit 102.

[0157] The camera 100 of Figure 1a may also comprise a processor 104 for controlling the operations of the camera 100. There may also be a memory 106 for storing data and computer code to be executed by the processor 104, and a transceiver 108 for communicating with, for example, a communication network and/or other devices in a wireless and/or wired manner. The camera 100 may further comprise a user interface (UI) 110 for displaying information to the user, for generating audible signals and/or for receiving user input. However, the camera 100 need not comprise each feature mentioned above, or may comprise other features as well. For example, there may be electric and/or mechanical elements for adjusting and/or controlling optics of the camera units 102 (not shown).

[0158] Figure 1a also illustrates some operational elements which may be implemented, for example, as a computer code in the software of the processor, in a hardware, or both. A focus control element 114 may perform operations related to adjustment of the optical system of camera unit or units to obtain focus meeting target specifications or some other predetermined criteria. An optics adjustment element 116 may perform movements of the optical system or one or more parts of it according to instructions provided by the focus control element 114. It should be noted here that the actual adjustment of the optical system need not be performed by the apparatus but it may be performed manually, wherein the focus control element 114 may provide information for the user interface 110 to indicate a user of the device how to adjust the optical system.

[0159] Figure 1b shows as a perspective view the camera 100 of Figure 1a. In Figure 1b seven camera units 102a-102g can be seen, but the camera 100 may comprise even more camera units which are not visible from this perspective. Figure 1b also shows two microphones 112a, 112b, but the apparatus may also comprise one or more than two microphones.

[0160] It should be noted here that embodiments disclosed in this specification may also be implemented with apparatuses having only one camera unit 102 or less or more than eight camera units 102a-102g.

[0161] In accordance with an embodiment, the camera 100 may be controlled by another device (not shown), wherein the camera 100 and the other device may communicate with each other and a user may use a user interface of the other device for entering commands, parameters, etc. and the user may be provided information from the camera 100 via the user interface of the other device.

[0162] Terms 360-degree video or virtual reality (VR) video may be used interchangeably. They may generally refer to video content that provides such a large field of view that only a part of the video is displayed at a single point of time in typical displaying arrangements. For example, a virtual reality video may be viewed on a head-mounted display (HMD) that may be capable of displaying e.g. about 100-degree field of view (FOV). The spatial subset of the virtual reality video content to be displayed may be selected based on the orientation of the head-mounted display. In another example, a flat-panel viewing environment is assumed, wherein e.g. up to 40-degree field-of-view may be displayed. When displaying wide field of view content (e.g. fisheye) on such a display, it may be preferred to display a spatial subset rather than the entire picture.

[0163] 360-degree image or video content may be acquired and prepared for example as follows. Images or video can be captured by a set of cameras or a camera device with multiple lenses and imaging sensors. The acquisition results in a set of digital image/video signals. The cameras/lenses may cover all directions around the center point of the camera set or camera device. The images of the same time instance are stitched, projected, and mapped onto a packed virtual reality frame, which may alternatively be referred to as a packed picture. The mapping may alternatively be referred to as region-wise mapping or region-wise packing. The breakdown of image stitching, projection, and mapping processes are illustrated with Figure 2a and described as follows. Input images 201 are stitched and projected 202 onto a three-dimensional projection structure, such as a sphere or a cube. The projection structure may be considered to comprise one or more surfaces, such as plane(s) or part(s) thereof. A projection structure may be defined as a three-dimensional structure consisting of one or more surface(s) on which the captured virtual reality image/video content may be projected, and from which a respective projected frame can be formed. The image data on the projection structure is further arranged onto a two-dimensional projected frame 203. The term projection may be defined as a process by which a set of input images are projected onto a projected frame or a projected picture. There may be a pre-defined set of representation formats of the projected frame, including for example an equirectangular panorama and a cube map representation format.

[0164] Region-wise mapping 204 may be applied to map projected frames 203 onto one or more packed virtual reality frames 205. In some cases, the region-wise mapping may be understood to be equivalent to extracting two or more regions from the projected frame, optionally applying a geometric transformation (such as rotating, mirroring, and/or resampling) to the regions, and placing the transformed regions in spatially non-overlapping areas, a.k.a. constituent frame partitions, within the packed virtual reality frame. If the region-wise mapping is not applied, the packed virtual reality frame 205 may be identical to the projected frame 203. Otherwise, regions of the projected frame are mapped onto a packed virtual reality frame by indicating the location, shape, and size of each region in the packed virtual reality frame. The term mapping may be defined as a process by which a projected frame is mapped to a packed virtual reality frame. The term packed virtual reality frame may be defined as a frame that results from a mapping of a projected frame. In practice, the input images 201 may be converted to packed virtual reality frames 205 in one process without intermediate steps.

[0165] Packing information may be encoded as metadata in or along the bitstream. For example, the packing information may comprise a region-wise mapping from a pre-defined or indicated source format to the packed frame format, e.g. from a projected frame to a packed VR frame, as described earlier. The region-wise mapping information may for example comprise for each mapped region a source rectangle in the projected frame and a destination rectangle in the packed VR frame, where samples within the source rectangle are mapped to the destination rectangle and rectangles may for example be indicated by the locations of the top-left corner and the bottom-right corner. The mapping may comprise resampling. Additionally or alternatively, the packing information may comprise one or more of the following: the orientation of the three-dimensional projection structure relative to a coordinate system, indication which omnidirectional projection format is used, region-wise quality ranking indicating the picture quality ranking between regions and/or first and second spatial region sequences, one or more transformation operations, such as rotation by 90, 180, or 270 degrees, horizontal mirroring, and vertical mirroring. The semantics of packing information may be specified in a manner that they are indicative for each sample location within packed regions of a decoded picture which is the respective spherical coordinate location.

[0166] In 360-degree systems, a coordinate system may be defined through orthogonal coordinate axes, such as X (lateral), Y (vertical, pointing upwards), and Z (back-to-front axis, pointing outwards). Rotations around the axes may be defined and may be referred to as yaw, pitch, and roll. Yaw may be defined to rotate around the Y axis, pitch around the X axis, and roll around the Z axis. Rotations may be defined to be extrinsic, i.e., around the X, Y, and Z fixed reference axes. The angles may be defined to increase clockwise when looking from the origin towards the positive end of an axis. The coordinate system specified can be used for defining the sphere coordinates, which may be referred to azimuth (ϕ) and elevation (θ).

[0167] Global coordinate axes may be defined as coordinate axes, e.g. according to the coordinate system as discussed above, that are associated with audio, video, and images representing the same acquisition position and intended to be rendered together. The origin of the global coordinate axes is usually the same as the center point of a device or rig used for omnidirectional audio/video acquisition as well as the position of the observer's head in the three-dimensional space in which the audio and video tracks are located. In the absence of the initial viewpoint metadata, the playback may be recommended to be started using the orientation (0, 0) in (azimuth, elevation) relative to the global coordinate axes.

[0168] As mentioned above, the projection structure may be rotated relative to the global coordinate axes. The rotation may be performed for example to achieve better compression performance based on the spatial and temporal activity of the content at certain spherical parts. Alternatively or additionally, the rotation may be performed to adjust the rendering orientation for already encoded content. For example, if the horizon of the encoded content is not horizontal, it may be adjusted afterwards by indicating that the projection structure is rotated relative to the global coordinate axes. The projection orientation may be indicated as yaw, pitch, and roll angles that define the orientation of the projection structure relative to the global coordinate axes. The projection orientation may be included e.g. in a box in a sample entry of an ISOBMFF track for omnidirectional video.

[0169] 360-degree panoramic content (i.e., images and video) cover horizontally the full 360-degree field-of-view around the capturing position of an imaging device. The vertical field-of-view may vary and can be e.g. 180 degrees. Panoramic image covering 360-degree field-of-view horizontally and 180-degree field-of-view vertically can be represented by a sphere that has been mapped to a two-dimensional image plane using equirectangular projection (ERP). In this case, the horizontal coordinate may be considered equivalent to a longitude, and the vertical coordinate may be considered equivalent to a latitude, with no transformation or scaling applied. In some cases panoramic content with 360-degree horizontal field-of-view but with less than 180-degree vertical field-of-view may be considered special cases of equirectangular projection, where the polar areas of the sphere have not been mapped onto the two-dimensional image plane. In some cases panoramic content may have less than 360-degree horizontal field-of-view and up to 180-degree vertical field-of-view, while otherwise have the characteristics of equirectangular projection format.

[0170] In cube map projection format, spherical video is projected onto the six faces (a.k.a. sides) of a cube. The cube map may be generated e.g. by first rendering the spherical scene six times from a viewpoint, with the views defined by an 90 degree view frustum representing each cube face. The cube sides may be frame-packed into the same frame or each cube side may be treated individually (e.g. in encoding). There are many possible orders of locating cube sides onto a frame and/or cube sides may be rotated or mirrored. The frame width and height for frame-packing may be selected to fit the cube sides "tightly" e.g. at 3x2 cube side grid, or may include unused constituent frames e.g. at 4x3 cube side grid.

[0171] A cube map can be stereoscopic. A stereoscopic cube map can e.g. be reached by re-projecting each view of a stereoscopic panorama to the cube map format.

[0172] The process of forming a monoscopic equirectangular panorama picture is illustrated in Figure 2b, in accordance with an embodiment. A set of input images 211, such as fisheye images of a camera array or a camera device 100 with multiple lenses and sensors 102, is stitched 212 onto a spherical image 213. The spherical image 213 is further projected 214 onto a cylinder 215 (without the top and bottom faces). The cylinder 215 is unfolded 216 to form a two-dimensional projected frame 217. In practice one or more of the presented steps may be merged; for example, the input images 213 may be directly projected onto a cylinder 217 without an intermediate projection onto the sphere 213 and/or to the cylinder 215. The projection structure for equirectangular panorama may be considered to be a cylinder that comprises a single surface.

[0173] The equirectangular projection may be defined as a process that converts any sample location within the projected picture (of the equirectangular projection format) to sphere coordinates of a coordinate system. The sample location within the projected picture may be defined relative to pictureWidth and pictureHeight, which are the width and height, respectively, of the equirectangular panorama picture in samples. In the following, let the center point of a sample location along horizontal and vertical axes be denoted as i and j, respectively. The sphere coordinates (ϕ, θ) for the sample location, in degrees, are given by the following equirectangular mapping equations: ϕ = (0.5 − i ÷ pictureWidth) * 360, θ = (0.5 − j ÷ pictureHeight) * 180. It is noted that depending on the direction of axes for (ϕ, θ) different conversion formulas may be derived.

[0174] In general, 360-degree content can be mapped onto different types of solid geometrical structures, such as polyhedron (i.e. a three-dimensional solid object containing flat polygonal faces, straight edges and sharp corners or vertices, e.g., a cube or a pyramid), cylinder (by projecting a spherical image onto the cylinder, as described above with the equirectangular projection), cylinder (directly without projecting onto a sphere first), cone, etc. and then unwrapped to a two-dimensional image plane. The two-dimensional image plane can also be regarded as a geometrical structure. In other words, 360-degree content can be mapped onto a first geometrical structure and further unfolded to a second geometrical structure. However, it may be possible to directly obtain the transformation to the second geometrical structure from the original 360-degree content or from other wide view visual content. In general, an omnidirectional projection format may be defined as a format to represent (up to) 360-degree content on a two-dimensional image plane. Examples of omnidirectional projection formats include the equirectangular projection format and the cubemap projection format.

[0175] In some cases panoramic content with 360-degree horizontal field-of-view but with less than 180-degree vertical field-of-view may be considered special cases of equirectangular projection, where the polar areas of the sphere have not been mapped onto the two-dimensional image plane. In some cases a panoramic image may have less than 360-degree horizontal field-of-view and up to 180-degree vertical field-of-view, while otherwise has the characteristics of equirectangular projection format.

[0176] Human eyes are not capable of viewing the whole 360 degrees space, but are limited to a maximum horizontal and vertical field-of-views (HHFoV, HVFoV). Also, a HMD device has technical limitations that allow only viewing a subset of the whole 360 degrees space in horizontal and vertical directions (DHFoV, DVFoV)).

[0177] In many displaying situations only a partial picture is needed to be displayed while the remaining picture is required to be decoded but is not displayed. These displaying situations include:

- Typical head-mounted displays (HMDs) display ~100 degrees field of view, while often the input video for HMD consumption covers entire 360 degrees.

- Typical flat-panel viewing environments display up to 40-degree field-of-view. When displaying wide-FOV content (e.g. fisheye) on such a display, it may be preferred to display a spatial subset rather than the entire picture.

[0178] A viewport may be defined as the part of the spherical video that is currently displayed and hence is viewable by the user(s). At any point of time, a video rendered by an application on a HMD renders a portion of the 360-degrees video, which is referred to as a viewport. Likewise, when viewing a spatial part of the 360-degree content on a conventional display, the spatial part that is currently displayed is a viewport. A viewport is a window on the 360-degrees world represented in the omnidirectional video displayed via a rendering display. A viewport may be characterized by a horizontal field-of-view (VHFoV) and a vertical field-of-view (VVFoV). In the following, the horizontal field-of-view of the viewport will be abbreviated with HFoV and, respectively, the vertical field-of-view of the viewport will be abbreviated with VFoV.

[0179] A tile track may be defined as a track that contains sequences of one or more motion-constrained tile sets of a coded bitstream. Decoding of a tile track without the other tile tracks of the bitstream may require a specialized decoder, which may be e.g. required to skip absent tiles in the decoding process. An HEVC tile track specified in ISO/IEC 14496-15 enables storage of one or more temporal motion-constrained tile sets as a track. When a tile track contains tiles of an HEVC base layer, the sample entry type 'hvt1' is used. When a tile track contains tiles of a non-base layer, the sample entry type 'lht1' is used. A sample of a tile track consists of one or more complete tiles in one or more complete slice segments. A tile track is independent from any other tile track that includes VCL NAL units of the same layer as this tile track. A tile track has a 'tbas' track reference to a tile base track. The tile base track does not include VCL NAL units. A tile base track indicates the tile ordering using a 'sabt' track reference to the tile tracks. An HEVC coded picture corresponding to a sample in the tile base track can be reconstructed by collecting the coded data from the tile-aligned samples of the tracks indicated by the 'sabt' track reference in the order of the track references.

[0180] A constructed tile set track is a tile set track, e.g. a track according to ISOBMFF, containing constructors that, when executed, result into a tile set bitstream.

[0181] A constructor is a set of instructions that, when executed, results into a valid piece of sample data according to the underlying sample format.

[0182] An extractor is a constructor that, when executed, copies the sample data of an indicated byte range of an indicated sample of an indicated track. Inclusion by reference may be defined as an extractor or alike that, when executed, copies the sample data of an indicated byte range of an indicated sample of an indicated track.

[0183] A full-picture-compliant tile set {track | bitstream} is a tile set {track | bitstream} that conforms to the full-picture {track | bitstream} format. Here, the notation {optionA | optionB} illustrates alternatives, i.e. either optionA or optionB, which is selected consistently in all selections. A full-picture-compliant tile set track can be played as with any full-picture track using the parsing and decoding process of full-picture tracks. A full-picture-compliant bitstream can be decoded as with any full-picture bitstream using the decoding process of full-picture bitstreams. A full-picture track is a track representing an original bitstream (including all its tiles). A tile set bitstream is a bitstream that contains a tile set of an original bitstream but not representing the entire original bitstream. A tile set track is a track representing a tile set of an original bitstream but not representing the entire original bitstream.

[0184] A full-picture-compliant tile set track may comprise extractors as defined for HEVC. An extractor may, for example, be an in-line constructor including a slice segment header and a sample constructor extracting coded video data for a tile set from a referenced full-picture track.

[0185] An in-line constructor is a constructor that, when executed, returns the sample data that it contains. For example, an in-line constructor may comprise a set of instructions for rewriting a new slice header. The phrase in-line may be used to indicate coded data that is included in the sample of a track.

[0186] A full-picture track is a track representing an original bitstream (including all its tiles).

[0187] A NAL-unit-like structure refers to a structure with the properties of a NAL unit except that start code emulation prevention is not performed.

[0188] A pre-constructed tile set track is a tile set track containing the sample data in-line.

[0189] A tile set bitstream is a bitstream that contains a tile set of an original bitstream but not representing the entire original bitstream.

[0190] A tile set track is a track representing a tile set of an original bitstream but not representing the entire original bitstream.

[0191] A recent trend in streaming in order to reduce the streaming bitrate of virtual reality video may be known as a viewport dependent delivery and can be explained as follows: a subset of 360-degree video content covering a primary viewport (i.e., the current view orientation) is transmitted at the best quality/resolution, while the remaining of 360-degree video is transmitted at a lower quality/resolution. There are generally two approaches for viewport-adaptive streaming:

[0192] The first approach is viewport-specific encoding and streaming, a.k.a. viewport-dependent encoding and streaming, a.k.a. asymmetric projection. In this approach, 360-degree image content is packed into the same frame with an emphasis (e.g. greater spatial area) on the primary viewport. The packed VR frames are encoded into a single bitstream. For example, the front face of a cube map may be sampled with a higher resolution compared to other cube faces and the cube faces may be mapped to the same packed VR frame as shown in Figure 3, where the front cube face is sampled with twice the resolution compared to the other cube faces.

[0193] The second approach is tile-based encoding and streaming. In this approach, 360-degree content is encoded and made available in a manner that enables selective streaming of viewports from different encodings.

[0194] An approach of tile-based encoding and streaming, which may be referred to as tile rectangle based encoding and streaming or sub-picture based encoding and streaming, may be used with any video codec, even if tiles similar to HEVC were not available in the codec or even if motion-constrained tile sets or alike were not implemented in an encoder. In tile rectangle based encoding, the source content may be split into tile rectangle sequences (a.k.a. sub-picture sequences) before encoding. Each tile rectangle sequence covers a subset of the spatial area of the source content, such as full panorama content, which may e.g. be of equirectangular projection format. Each tile rectangle sequence may then be encoded independently from each other as a single-layer bitstream, such as HEVC Main profile bitstream. Several bitstreams may be encoded from the same tile rectangle sequence, e.g. for different bitrates. Each tile rectangle bitstream may be encapsulated in a file as its own track (or alike) and made available for streaming. At the receiver side the tracks to be streamed may be selected based on the viewing orientation. The client may receive tracks covering the entire omnidirectional content. Better quality or higher resolution tracks may be received for the current viewport compared to the quality or resolution covering the remaining, currently non-visible viewports. In an example, each track may be decoded with a separate decoder instance.

[0195] In an example of tile rectangle based encoding and streaming, each cube face may be separately encoded and encapsulated in its own track (and Representation). More than one encoded bitstream for each cube face may be provided, e.g. each with different spatial resolution. Players can choose tracks (or Representations) to be decoded and played based on the current viewing orientation. High-resolution tracks (or Representations) may be selected for the cube faces used for rendering for the present viewing orientation, while the remaining cube faces may be obtained from their low-resolution tracks (or Representations).

[0196] In an approach of tile-based encoding and streaming, encoding is performed in a manner that the resulting bitstream comprises motion-constrained tile sets. Several bitstreams of the same source content are encoded using motion-constrained tile sets.

[0197] In an approach, one or more motion-constrained tile set sequences are extracted from a bitstream, and each extracted motion-constrained tile set sequence is stored as a tile set track (e.g. an HEVC tile track or a full-picture-compliant tile set track) or a sub-picture track in a file. A tile base track (e.g. an HEVC tile base track or a full picture track comprising extractors to extract data from the tile set tracks) may be generated and stored in a file. The tile base track represents the bitstream by implicitly collecting motion-constrained tile sets from the tile set tracks or by explicitly extracting (e.g. by HEVC extractors) motion-constrained tile sets from the tile set tracks. Tile set tracks and the tile base track of each bitstream may be encapsulated in an own file, and the same track identifiers may be used in all files. At the receiver side the tile set tracks to be streamed may be selected based on the viewing orientation. The client may receive tile set tracks covering the entire omnidirectional content. Better quality or higher resolution tile set tracks may be received for the current viewport compared to the quality or resolution covering the remaining, currently non-visible viewports.

[0198] In an example, equirectangular panorama content is encoded using motion-constrained tile sets. More than one encoded bitstream may be provided, e.g. with different spatial resolution and/or picture quality. Each motion-constrained tile set is made available in its own track (and Representation). Players can choose tracks (or Representations) to be decoded and played based on the current viewing orientation. High-resolution or high-quality tracks (or Representations) may be selected for tile sets covering the present primary viewport, while the remaining area of the 360-degree content may be obtained from low-resolution or low-quality tracks (or Representations).

[0199] In an approach, each received tile set track is decoded with a separate decoder or decoder instance.

[0200] In another approach, a tile base track is utilized in decoding as follows. If all the received tile tracks originate from bitstreams of the same resolution (or more generally if the tile base tracks of the bitstreams are identical or equivalent, or if the initialization segments or other initialization data, such as parameter sets, of all the bitstreams is the same), a tile base track may be received and used to construct a bitstream. The constructed bitstream may be decoded with a single decoder.

[0201] In yet another approach, a first set of tile rectangle tracks and/or tile set tracks may be merged into a first full-picture-compliant bitstream, and a second set of tile rectangle tracks and/or tile set tracks may be merged into a second full-picture-compliant bitstream. The first full-picture-compliant bitstream may be decoded with a first decoder or decoder instance, and the second full-picture-compliant bitstream may be decoded with a second decoder or decoder instance. In general, this approach is not limited to two sets of tile rectangle tracks and/or tile set tracks, two full-picture-compliant bitstreams, or two decoders or decoder instances, but applies to any number of them. With this approach, the client can control the number of parallel decoders or decoder instances. Moreover, clients that are not capable of decoding tile tracks (e.g. HEVC tile tracks) but only full-picture-compliant bitstreams can perform the merging in a manner that full-picture-compliant bitstreams are obtained. The merging may be solely performed in the client or full-picture-compliant tile set tracks may be generated to assist in the merging performed by the client.

[0202] A motion-constrained coded sub-picture sequence may be defined as a collective term of such a coded sub-picture sequence in which the coded pictures are motion-constrained pictures, as defined earlier, and an MCTS sequence. Depending on the context of using the term motion-constrained coded sub-picture sequence, it may be interpreted to mean either one or both of a coded sub-picture sequence in which the coded pictures are motion-constrained pictures, as defined earlier, and/or an MCTS sequence.

[0203] A collector track may be defined as a track that extracts implicitly or explicitly MCTSs or sub-pictures from other tracks. A collector track may be a full-picture-compliant track. A collector track may for example extract MCTSs or sub-pictures to form a coded picture sequence where MCTSs or sub-pictures are arranged to a grid. For example, when a collector track extracts two MCTSs or sub-pictures, they may be arranged into a 2x1 grid of MCTSs or sub-pictures. A tile base track may be regarded as a collector track, and an extractor track that extracts MCTSs or sub-pictures from other tracks may be regarded as a collector track. A collector track may also be referred to as a collection track. A track that is a source for extracting to a collector track may be referred to as a collection item track.

[0204] The term tile merging (in coded domain) may be defined as a process to merge coded sub-picture sequences and/or coded MCTS sequences, which may have been encapsulated as sub-picture tracks and tile tracks, respectively, into a full-picture-compliant bitstream. A creation of a collector track may be regarded as tile merging that is performed by the file creator. Resolving a collector track into a full-picture-compliant bitstream may be regarded as tile merging, which is assisted by the collector track.

[0205] It is also possible to combine the first approach (viewport-specific encoding and streaming) and the second approach (tile-based encoding and streaming) above.

[0206] It needs to be understood that tile-based encoding and streaming may be realized by splitting a source picture in sub-picture sequences that are partly overlapping. Alternatively or additionally, bitstreams with motion-constrained tile sets may be generated from the same source content with different tile grids or tile set grids. We could then imagine the 360 degrees space divided into a discrete set of viewports, each separate by a given distance (e.g., expressed in degrees), so that the omnidirectional space can be imagined as a map of overlapping viewports, and the primary viewport is switched discretely as the user changes his/her orientation while watching content with a head-mounted display. When the overlapping between viewports is reduced to zero, the viewports could be imagined as adjacent non-overlapping tiles within the 360 degrees space.

[0207] As explained above, in viewport-adaptive streaming the primary viewport (i.e., the current viewing orientation) is transmitted at the best quality/resolution, while the remaining of 360-degree video is transmitted at a lower quality/resolution. When the viewing orientation changes, e.g. when the user turns his/her head when viewing the content with a head-mounted display, another version of the content needs to be streamed, matching the new viewing orientation. In general, the new version can be requested starting from a stream access point (SAP), which are typically aligned with (sub)segments. In single-layer video bitstreams, SAPs are intra-coded and hence costly in terms of rate-distortion performance. Conventionally, relatively long SAP intervals and consequently relatively long (sub)segment durations in the order of seconds are hence used. Thus, the delay (here referred to as the viewport quality update delay) in upgrading the quality after a viewing orientation change (e.g. a head turn) is conventionally in the order of seconds and is therefore clearly noticeable and may be annoying.

[0208] Extractors specified in ISO/IEC 14496-15 for H.264/AVC and HEVC enable compact formation of tracks that extract NAL unit data by reference. An extractor is a NAL-unit-like structure. A NAL-unit-like structure may be specified to comprise a NAL unit header and NAL unit payload like any NAL units, but start code emulation prevention (that is required for a NAL unit) might not be followed in a NAL-unit-like structure. For HEVC, an extractor contains one or more constructors. A sample constructor extracts, by reference, NAL unit data from a sample of another track. An in-line constructor includes NAL unit data.

When an extractor is processed by a file reader that requires it, the extractor is logically replaced by the bytes resulting when resolving the contained constructors in their appearance order. Nested extraction may be disallowed, e.g. the bytes referred to by a sample constructor shall not contain extractors; an extractor shall not reference, directly or indirectly, another extractor. An extractor may contain one or more constructors for extracting data from the current track or from another track that is linked to the track in which the extractor resides by means of a track reference of type 'scal'. The bytes of a resolved extractor may represent one or more entire NAL units. A resolved extractor starts with a valid length field and a NAL unit header. The bytes of a sample constructor are copied only from the single identified sample in the track referenced through the indicated 'scal' track reference. The alignment is on decoding time, i.e. using the time-to-sample table only, followed by a counted offset in sample number. An extractor track may be defined as a track that contains one or more extractors.

[0209] Extractors are a media-level concept and hence apply to the destination track before any edit list is considered. However, one would normally expect that the edit lists in the two tracks would be identical.

[0210] The following syntax may be used:

```
class aligned(8) Extractor () {
        NALUnitHeader();
        do {
                unsigned int(8)       constructor_type;
                if( constructor_type == 0 )
                      SampleConstructor();
                else if( constructor_type == 2 )
                      InlineConstructor();
        } while( !EndOfNALUnit() )
}
```

[0211] The semantics may be defined as follows:

NALUnitHeader() is the first two bytes of HEVC NAL units. A particular nal_unit_type value indicates an extractor, e.g. nal_unit_type equal to 49.

constructor_type specifies the constructor being used.

EndOfNALUnit() is a function that returns 0 (false) when more data follows in this extractor; otherwise it returns 1 (true).

[0212] The sample constructor (SampleConstructor) may have the following syntax:

```
class aligned(8) SampleConstructor () {
        unsigned int(8) track_ref_index;
        signed   int(8) sample_offset;
        unsigned int((lengthSizeMinusOne+1)*8)
                data_offset;
        unsigned int((lengthSizeMinusOne+1)*8)
```

```
        data_length;
}
```

[0213] track_ref_index identifies the source track from which data is extracted. track_ref_index is the index of the track reference of type 'scal'. The first track reference has the index value 1; the value 0 is reserved.

[0214] The sample in that track from which data is extracted is temporally aligned or nearest preceeding in the media decoding timeline, i.e. using the time-to-sample table only, adjusted by an offset specified by sample_offset with the sample containing the extractor. sample_offset gives the relative index of the sample in the linked track that shall be used as the source of information. Sample 0 (zero) is the sample with the same, or the closest preceding decoding time compared to the decoding time of the sample containing the extractor; sample 1 (one) is the next sample, sample -1 (minus 1) is the previous sample, and so on.

[0215] data_offset is the offset of the first byte within the reference sample to copy. If the extraction starts with the first byte of data in that sample, the offset takes the value 0.

[0216] data_length is the number of bytes to copy.

[0217] The syntax of the in-line constructor may be specified as follows:

```
class aligned(8) InlineConstructor () {
        unsigned int(8) length;
        unsigned int(8) inline_data[length];
}
```

[0218] length is the number of bytes that belong to the InlineConstructor following this field, and inline_data is the data bytes to be returned when resolving the in-line constructor.

[0219] Coded data of several tile tracks may be merged to one e.g. as follows.

[0220] In an approach, the file/segment encapsulation generates pre-constructed tile tracks, which may be full-picture-compliant. Furthermore, the file/segment encapsulation generates constructed full-picture track(s) that use pre-constructed tile tracks as reference for construction. The instructions may be stored in the same file with the segment(s) or media file(s), or they may be stored in separate segment hint file(s). The format of the instructions may but need not comply with ISOBMFF (or more generally the format used for the segment(s) or media file(s)). For example, the instructions may form a track (which may be called e.g. MPEG-DASH segment hint track) according to ISOBMFF, and each sample of the track may provide instructions to construct a segment or subsegment.

[0221] Coded sub-picture sequences may be merged e.g. as follows and as depicted in Figure 4.

[0222] The source picture sequence 71 is split 72 into sub-picture sequences 73 before encoding. Each sub-picture sequence 73 is then encoded 74 independently.

[0223] Two or more coded sub-picture sequences 75 are merged 76 into a bitstream 77. The coded sub-picture sequences 75 may have different characteristics, such as picture quality, so as to be used for viewport-dependent delivery. The coded sub-pictures 75 of a time instance are merged vertically into a coded picture of the bitstream 77. Each coded sub-picture 75 in a coded picture forms a coded slice. Vertical arrangement of the coded sub-pictures 75 into a coded picture may bring at least the following benefits:

- Slices can be used as a unit to carry a coded sub-picture and no tile support is needed in the codec, hence the approach is suitable e.g. for H.264/AVC.
- No transcoding is needed for the vertical arrangement, as opposed to horizontal arrangement where transcoding would be needed as coded sub-pictures would be interleaved in the raster scan order (i.e., the decoding order) of blocks (e.g. macroblocks in H.264/AVC or coding tree units in HEVC).
- Motion vectors that require accessing sample locations horizontally outside the picture boundaries (in inter prediction) can be used in the encoding of sub-picture sequences. Hence, the compression efficiency benefit that comes from allowing motion vectors over horizontal picture boundaries is maintained (unlike e.g. when using motion-constrained tile sets).

[0224] The merged bitstream 77 is full-picture compliant. For example, if sub-picture sequences were coded with H.264/AVC, the merged bitstream is also compliant with H.264/AVC and can be decoded with a regular H.264/AVC decoder.

[0225] In resolution-adaptive MCTS-based viewport-adaptive streaming several HEVC bitstreams of the same omnidirectional source content are encoded at different resolutions using motion-constrained tile sets. When the bitstreams are encapsulated into file(s), tile tracks are formed from each motion-constrained tile set sequence. Clients that are capable of decoding HEVC tile streams can receive and decode tile tracks independently.

[0226] In addition to tile tracks, 'hvc2'/'hev2' tracks containing extractors (a.k.a. extractor tracks) can be formed for each expected viewing orientation. An extractor track corresponds to a dependent Representation in the DASH MPD, with @dependencyId including the Representation identifiers of the tile tracks from which the tile data is extracted. Clients that are not capable of decoding HEVC tile streams but only fully compliant HEVC bitstreams can receive and decode the extractor tracks.

[0227] Figure 5 presents an example how extractor tracks can be used for tile-based omnidirectional video streaming. A 4x2 tile grid has been used in forming of the motion-constrained tile sets 81a, 81b. In many viewing orientations 2x2 tiles out of the 4x2 tile grid are needed to cover a typical field of view of a head-mounted display. In the example, the presented extractor track for high-resolution motion-constrained tile sets 1, 2, 5 and 6 covers certain viewing orientations, while the extractor track for low-resolution motion-constrained tile sets 3, 4, 7, and 8 includes a region assumed to be non-visible for these viewing orientations. Two HEVC decoders are used in this example, one for the high-resolution extractor track and another for the low-resolution extractor track.

[0228] While the description above referred to tile tracks, it should be understood that sub-picture tracks can be similarly formed.

[0229] Tile merging in coded domain is needed or beneficial for the following purposes:

- Enable a number of tiles that is greater than the number of decoder instances, down to one decoder only
- Avoid synchronization challenges of multiple decoder instances
- Reach higher effective spatial and temporal resolutions, e.g. 6k@60fps with 4k@60fps decoding capacity
- Enable specifying interoperability points for standards as well as client APIs that require one decoder only

[0230] By selecting the vertical or horizontal tile grid to be aligned in bitstreams of different resolution, it is possible to combine tiles 82, 83 from bitstreams of different resolution and use a single decoder for decoding the resulting the bitstream. This is illustrated with Figure 6a, where constant boundaries indicate motion-constrained tile sets and dotted boundaries indicate tile boundaries without motion constraints.

[0231] In this example, four tiles 84 of the high-resolution version are selected. Four MCTSs of 4×2 MCTSs grid of a picture of the equirectangular projection format provides high-resolution viewport of 90° horizontal and vertical field-of-view in all viewing orientations (at 98% coverage of the viewport) and in a vast majority of viewing orientations (at 100% coverage). The created extractor track 86 may contain region-wise packing information and the selected four high-resolution MCTSs 84 and possibly also tiles selected from the low-resolution version. The low-resolution tiles 85 from the low-resolution bitstream 83 may be selected among the tiles which represent the non-visible areas. In Figure 6a, the hatched tiles illustrate an example of the selected high-resolution tiles and the selected low-resolution tiles suitable for a particular range of viewing orientations.

[0232] In accordance with an example, the size of the high-resolution picture is 5120x2560 pixels, the tile size of the high-resolution bitstream is 1280x1280 pixels, the size of the low-resolution picture is 2560x1280 pixels, and the tile size of the low-resolution bitstream is 640x640 pixels, but in some other embodiments the picture sizes and/or the tile sizes may be different from those.

[0233] The coding scenario above may be sub-optimal, since tile boundaries (regardless of whether motion constraints are applied) break in-picture prediction. For example, intra prediction and spatial motion prediction are not applied across tile boundaries, and entropy coding state is not carried over a tile boundary. Since the high-resolution bitstream has a tile grid that is twice as fine as the motion-constrained tile set grid, the rate-distortion performance of the high-resolution bitstream is compromised.

[0234] Many embodiments of the present invention avoid the above-mentioned shortcoming. In the present invention the tile grid used in the high-resolution bitstream does not need to be selected to be aligned horizontally or vertically with the tile grid used in the low-resolution bitstream. Consequently, the rate-distortion performance of the high-resolution bitstream is not compromised due to unnecessarily fine tile grid.

[0235] There may be picture size limitations e.g. imposed by coding standards. The picture size limitations may have been introduced for example to make memory allocation and management practical. For example, the maximum number of luma samples in the luma sample array in HEVC Level 5.1 is 8 912 896. However, the constraints of the levels of coding standards may be designed for relatively high picture rates, such as 60 Hz. For example, HEVC Level 5.1 allows resolution 4096x2160 at 60 Hz. Such picture size limitations may disallow spatial frame packing of stereoscopic video in practical codec levels. Thus, spatial frame packing of constituent pictures of 4096x2048 luma samples e.g. to a picture of 4096x4096 luma samples does not comply with the constraint of Level 5.1. Yet, the maximum luma sample rate Level 5.1 would be sufficient for decoding stereoscopic frame-packed content of resolution 4096x4096 at more than 30 frame-packed pictures per second. The present invention helps in avoiding such picture size limitations.

[0236] Only a viewport of 360-degree video is typically displayed at a particular time. It would be desirable to make the resolution of the viewport in the 360-degree video as high as allowed by the display device and to make the decoding resources required for non-displayed parts of 360-degree video as small as possible. In some cases, it would be desirable to make the resolution of the viewport higher than allowed by the picture size limitations discussed above. For example, the use of 8K equirectangular picture resolution (e.g. 7680x3840) in content production may be desirable, while the decoding capacity may be limited e.g. to that of HEVC Level 5.1 (i.e., 4096x2160 at 60 Hz). In practical encoding and streaming scenarios, such as tile-based viewport-adaptive encoding and streaming discussed above, the size of the high-resolution and/or high-quality portion is greater than the viewport size for example to compensate head motion in viewing using head-mounted displays. Thus, it may be desirable to provide a portion of the 360-video that is large enough to compensate viewport orientation changes at a resolution that is effectively greater than the decoding capacity.

[0237] In order to alleviate or overcome the problems and shortcomings above, a set of two or more coded pictures representing at least one input picture of the same time instance are generated in a manner that the two or more coded pictures represent regions of the at least one input picture with different resolutions. The two or more coded pictures may be referred to as a single-time coded picture set.

[0238] The single-time coded picture set may be generated by an encoder. Alternatively or additionally, the single-time coded picture set may be generated by a bitstream merger that merges two or more coded bitstreams. These two or more coded bitstreams may be referred to as source bitstreams for single-time coded picture set merging.

[0239] Instructions for merging source bitstreams to generate a single-time coded picture set may be generated by an encapsulator (e.g. a file generator as described in relation to Figure 11). Instructions may comprise extractors for extracting parts of source bitstreams for single-time coded picture set merging.

[0240] To sum up, at least the following possibilities exist when it comes to the entities performing parts of the operation:

- An encoder (e.g. block 1510 in Figure 11) encodes the single-time coded picture set. In other words, the encoder generates the spatiotemporal frame-packed bitstream.
- The bitstream merger as described above may operate as part of or operationally connected with different entities, including but not limited to the following:
    - A transcoding entity that e.g. prepares streamable versions of the content from the input bitstreams. Such a transcoding entity may, for example, operate as distributed cloud software.
    - A file generator (e.g. as described in relation to Figure 11), which may also be referred to as a file encapsulator,
    - A sender or a server (e.g. block 1530 in Figure 11).
    - A receiver (e.g. block 1550 in Figure 11).

- A file/segment decapsulator or parser that is included or operationally connected with the receiver, a player, and/or decoder(s).
- A decoder (e.g. block 1560 in Figure 11).

[0241] The bitstream merger may but need not operate by following the instructions described above.

[0242] A reason for generating the single-time coded picture set in a manner that it comprises regions of the at least one input picture with different resolutions is to cover a viewport with higher resolution region(s) than the resolutions(s) of other areas covered by the single-time coded picture set. Here, the phrasing higher resolution may be understood to mean a denser sampling grid in the projected picture.

[0243] In some embodiments, the client reports the viewport orientation to the sender, the bitstream merger, and/or the encoder. The reporting may happen periodically or may be triggered by viewport orientation changes. The reporting may also include the viewport size (e.g. width and height, or horizontal and vertical field of view). The encoder or the bitstream merger selects the region(s) having higher resolution so that they cover the viewport.

[0244] In some embodiments, several sets of instructions for merging source bitstreams to generate a single-time coded picture set (i.e. more generally a spatiotemporal frame-packed bitstream) are generated e.g. by an encapsulator and/or several spatiotemporal frame-packed bitstreams are generated by an encoder and/or a bitstream merger. Each set of instructions or respectively each spatiotemporal frame-packed bitstream may be intended for a particular viewport orientation and/or size. The client may select which set of instructions and/or which spatiotemporal frame-packed bitstream is received based on the viewport orientation and/or size, so that decoded spatiotemporal frame-packed bitstream covers the viewport at higher resolution(s) (than the resolution(s) for the other areas covered by the spatiotemporal frame-packed bitstream).

[0245] In an embodiment, a single-time coded picture set representing a first input picture of a first view and a second input picture of a second view is generated in a manner that at least one picture of the single-time coded picture set represents at least a part of both the first input picture and the second input picture. The first and second input picture represent the same time instance.

[0246] In an embodiment, the encoding of source bitstreams for single-time coded picture set merging is characterized by using the same coding structure in all the bitstreams, particularly when it comes to picture order counts and reference picture sets, and the encoding comprises generating one or more dummy pictures per a single-time coded picture set in at least one of the source bitstreams. The dummy pictures may be used as place-holders to insert coded data from another source bitstream as part of the merging. The picture content of the dummy pictures needs not be semantically relevant and can represent for example a plane of constant color. The dummy pictures may be marked as not to be output by a decoder; hence, decoded dummy pictures would not appear in a decoded output of a source bitstream.

[0247] In the following, an example embodiment is illustrated using the cube map projection. However, the cube map projection is not the only projection which may be used but also other types of projections may be utilized.

[0248] A stereoscopic input picture sequence of the cube map projection format is obtained. The cube faces can be arranged for example onto a 3x2 grid. Two resolutions of both views of the cube map are derived, the low-resolution version being for example a quarter of the size of the high-resolution version in pixels. In this example, each cube face in both resolutions is partitioned into four tiles, each coded as an MCTS. Figure 6b illustrates an example of this kind of partitioning. In Figure 6b, the top-most picture 87a illustrates a high-resolution version of a first view, e.g. a left view, the picture 87b below the top-most picture illustrates a low-resolution version of the first view, the next picture 88a illustrates a high-resolution version of a second view, e.g. a right view, and the lower-most picture 88b illustrates a low-resolution version of the second view.

[0249] An example of encoding tile sets of different resolution and different views is presented next with reference to Figure 6c, in accordance with an embodiment. The high-resolution views 87a, 88a are encoded in a temporally interleaved manner. In this embodiment, a dummy picture 89 is added after the two views so that every third picture in the high-resolution bitstream 90 is the dummy picture. The picture content of the dummy pictures 89 can be arbitrary, for example they may contain constant color. The dummy pictures 89 are merely used as place-holders to insert low-resolution tiles when merging the high- and low-resolution bitstreams and which may particularly make reference picture management easier in the merging process. The second view 88a (View 1) may be predicted from pictures of the first view 87a (View 0). If the coded pictures of the first view are not extracted from the bitstream and used as monoscopic version of the content, coded pictures of the first view may also be predicted from coded pictures of the second view.

[0250] Each of the low-resolution views 87b, 88b may be encoded as a separate bitstream 91a, 91b. In this embodiment, two out of three pictures in the low-resolution bitstreams are dummy pictures 92, which are merely used as place-holders for high-resolution tiles when merging the high- and low-resolution bitstreams.

[0251] The dummy pictures 89 of the high-resolution bitstream 90 maintain the corresponding reference pictures in the reference picture set as the respective (non-dummy) pictures in the low-resolution bitstreams 91a, 91b. Vice versa, the dummy pictures 92 of the low-resolution bitstreams 91a, 91b maintain the corresponding reference pictures in the reference picture set as the respective (non-dummy) pictures of the high-resolution bitstream. In accordance with an embodiment, the coding structure in all the bitstreams is identical, particularly when it comes to picture order counts and which pictures are kept as reference pictures e.g. by using reference picture sets. Consequently, dummy pictures can be replaced by tiles of respective pictures from other bitstream(s).

[0252] Figure 6d illustrates how an extractor track may be generated and/or tiles may be extracted from the encoded bitstreams 90, 91a, 91b to generate a spatiotemporal frame-packed bitstream 92, in accordance with an embodiment. Eight tiles in the above-described arrangement can be understood to cover one entire cube face and additionally two halves of other cube faces, and can be considered to roughly cover 135 degree field of view horizontally and vertically. For a particular viewing orientation, eight high-resolution tiles 93a, 93b are selected from both views encoded into the high-resolution bitstream 90. The respective samples of the extractor track comprise extractors each of which includes by reference the selected eight high-resolution tiles for the respective view. The 16 tiles that cover the remaining of the cube area are selected from both low-resolution bitstreams 91a, 91b. The low-resolution bitstreams 91a, 91b are spatially merged, i.e. tiles of the two respective pictures of the low-resolution bitstreams 91a, 91b are merged into one picture. The spatially merged low-resolution picture 93c is arranged in the place of the high-resolution dummy picture 89 in the spatiotemporal frame-packed bitstream 92, which is a fully compliant bitstream. In HEVC, the picture parameter set structure defines the tile grid, and hence the tile structure can be changed on picture basis, as may be done in the spatiotemporal frame-packed bitstream 92.

[0253] Using the principles presented above enables resolution-adaptive tile merging so that a single decoder can be used for improved effective decoding resolution or picture rate. For example, effective resolution of 4K stereoscopic content @60 Hz may be obtained with decoder capacity of monoscopic 4K @60 Hz. Furthermore, only one version per resolution needs to be encoded.

[0254] While the example above is illustrated with cube maps, the principles may suit all other projection formats used in 360-degree video. For example, when 12x8 tile grid is used for equirectangular projection, 32 tiles may cover a 90-degree viewport in almost all viewing orientations. Similarly, it may be possible to extract 32 tiles of view 0 of the high-resolution bitstream to a first picture, 32 tiles of view 1 of the high-resolution bitstream to a second picture, and 64+64 tiles of the low-resolution bitstreams to a third picture of the same time instant.

[0255] The same tile grid is used in both high- and low-resolution versions, and hence no overlaps of extracted tiles occur and the decoding capacity is not wasted.

[0256] Another example embodiment is illustrated with reference to Figure 6e using the equirectangular projection. It needs to be understood that the particular resolutions (width and height in pixels) are intended for providing concrete examples and the example embodiment can similarly be applied to other resolutions too.

[0257] A stereoscopic omnidirectional input picture sequence of the equirectangular projection format is obtained. Two resolutions of both views are derived:

- a high-resolution version 95a of view 0 (in the example figure the left view at resolution 5120x2560),
- a high-resolution version 95b of view 1 that has 4/5 of the width and height in terms of pixels compared to those of the high-resolution version of view 0 (in the example figure the right view at resolution 4096x2048),
- a low-resolution version 96a of view 0 that has 1/2 of the width and height in terms of pixels compared to those of the high-resolution version of view 1 (in the example figure the left view at resolution 2048x1024)
- a low-resolution version 96b of view 1 that has the same width and height in terms of pixels compared to those of the low-resolution version of view 0 (in the example figure the right view at resolution 2048x1024)

[0258] 4x2 tile grid and 4x2 MCTS grid are used in encoding the high-resolution version of view 0 and the low-resolution versions of both views. 8x2 tile grid and 4x2 MCTS grid is used in encoding the high-resolution version of view1, where tile boundaries that are not MCTS boundaries are illustrated by dashed lines. In all bitstreams, every second picture may be a dummy picture. In the high-resolution bitstream of view 0, the second picture of the bitstream may be the first dummy picture (and after that dummy pictures follow at every other coded picture). In the other bitstreams, the first picture of the bitstream may be the first dummy picture (and after that dummy pictures follow at every other coded picture).

[0259] MCTSs 97a, 97b from the high-resolution bitstreams may be selected for example to cover a viewport, and MCTSs 98a, 98b from the low-resolution bitstreams may be selected to cover the remaining area covered by the omnidirectional video.

[0260] A single-time coded picture set is formed from each pair of coded pictures in the four bitstreams, where, in each pair, one of the coded pictures is a dummy picture. The first picture in a single-time coded picture set is extracted from the selected MCTSs of the high-resolution bitstream of view 0. The first picture 99a has a tile grid of 2x2 and an MCTS grid of 2x2, and has the size of 2560x2560 pixels in this example. The second picture 99b in the single-time coded picture set is extracted from the selected MCTSs of the high-resolution bitstream of view 1 and of the two low-resolution bitstreams, as illustrated in Figure 6e. The second picture has a tile grid of 5x3. In the two top-most tile rows, the height of a tile in the second picture is double of the tile height in the low-resolution bitstreams. Hence, each tile originating from a low-resolution bitstream is encapsulated in its own slice in the two top-most tile rows. The tile at the bottom-right corner can contain any content.

[0261] It should be understood that there are other alternatives for encoding and merging than those illustrated in Figure 6e. For example, a tile grid of 8x4 could be used for the high-resolution bitstream of view 1, in which case the use of slices is not needed for the encapsulating any of the tiles of the low-resolution bitstream.

[0262] The presented example takes advantage of the human visual system that has been found to perceive picture quality closer to the higher fidelity view, i.e. view 0 in this example. The tiles of the low-resolution bitstreams in the merged bitstream can serve as a backup for quick and large changes in viewport orientation. The picture size in the presented example is 2560x2560, which complies with the constraints of HEVC Level 5.1. Consequently, HEVC Level 5.1 decoders are capable of decoding the merged bitstream and produce a stereoscopic output picture sequence at 30 Hz. Since the high-resolution bitstream of view 0 originates from a 5K (5120x2560) input picture sequence, the presented example can be considered to provide a 5K effective resolution with 4K decoding capability.

[0263] Another example embodiment is illustrated with reference to Figure 6f using the equirectangular projection. It needs to be understood that the particular resolutions (width and height in pixels) are intended for providing concrete examples and the example embodiment can similarly be applied to other resolutions too.

[0264] A stereoscopic omnidirectional input picture sequence of the equirectangular projection format is obtained. Two resolutions of both views are derived:

- a high-resolution version 95a of view 0 (in the example figure the left view at resolution 6144x3072),

- a high-resolution version 95b of view 1 that has 2/3 of the width and height in terms of pixels compared to those of the high-resolution version of view 0 (in the example figure the right view at resolution 4096x2048),
- a low-resolution version 96a of view 0 that has 1/2 of the width and height in terms of pixels compared to those of the high-resolution version of view 1 (in the example figure the left view at resolution 2048x1024)
- a low-resolution version 96b of view 1 that has the same width and height in terms of pixels compared to those of the low-resolution version of view 0 (in the example figure the right view at resolution 2048x1024)

[0265] 4x2 tile grid and 4x2 MCTS grid are used in encoding the high-resolution version of view 0 and the low-resolution versions of both views. 8x2 tile grid and 4x2 MCTS grid is used in encoding the high-resolution version of view1, where tile boundaries that are not MCTS boundaries are illustrated by dashed lines.

[0266] Two bitstreams of each combination of view and resolution may be encoded. In all bitstreams, every second picture may be a dummy picture. The first picture of a first bitstream of a particular combination of view and resolution may be the first dummy picture (and after that dummy pictures follow at every other coded picture), whereas the second picture of another bitstream of the same combination of view and resolution may be the first dummy picture (and after that dummy pictures follow at every other coded picture).

[0267] MCTSs 97a, 97b from the high-resolution bitstreams may be selected for example to cover a viewport, and MCTSs 98a, 98b from the low-resolution bitstreams may be selected to cover the remaining area covered by the omnidirectional video.

[0268] A single-time coded picture set may be formed from each pair of coded pictures in the eight bitstreams, where, in each pair, one of the coded pictures is a dummy picture. The first picture 99a in a single-time coded picture set may be extracted from the bitstreams where the second picture of the bitstream is a dummy picture, and the second picture 99b in a single-time coded picture set may be extracted from the bitstreams where the first picture of the bitstream is a dummy picture.

[0269] The pictures 99a, 99b of the single-time coded picture set have a tile grid of 3x2 (with tile column widths equal to 1536, 512, and 512 pixels from left to right and tile row heights equal to 1536 and 1536 pixels). Each tile originating from a low-resolution bitstream is encapsulated in its own slice.

[0270] It should be understood that there are other alternatives for encoding and merging than those illustrated in Figure 6f. For example, a single picture in the single-time coded picture set may comprise all selected low-resolution tiles of a view. In another example, the order of a slice encapsulating a high-resolution tile of view 1 and a slice encapsulating a low-resolution tile may be changed within a tile of a picture of the single-time coded picture set.

[0271] The presented example takes advantage of the human visual system that has been found to perceive picture quality closer to the higher fidelity view, i.e. view 0 in this example. The tiles of the low-resolution bitstreams in the merged bitstream can serve as a backup for quick and large changes in viewport orientation. The picture size in the presented example is 2560x3072, which complies with the constraints of HEVC Level 5.1. Consequently, HEVC Level 5.1 decoders are capable of decoding the merged bitstream and produce a stereoscopic output picture sequence at 30 Hz. Since the high-resolution bitstream of view 0 originates from a 6K (6144x3072) input picture sequence, the presented example can be considered to provide a 6K effective resolution with 4K decoding capability.

[0272] Another example embodiment is illustrated with reference to Figure 6g using the equirectangular projection. It needs to be understood that the particular resolutions (width and height in pixels) are intended for providing concrete examples and the example embodiment can similarly be applied to other resolutions too.

[0273] A monoscopic omnidirectional input picture sequence of the equirectangular projection format is obtained. Two resolutions are derived:

- a high-resolution version 95 (in the example figure at resolution 7680x3840),
- a low-resolution version 96 that has 2/5 of the width and height in terms of pixels compared to those of the high-resolution version (in the example figure at resolution 3072x1536).

[0274] 4x2 tile grid and 4x2 MCTS grid are used in encoding bitstreams of both resolutions. Two bitstreams of each resolution may be encoded. In all bitstreams, every second picture may be a dummy picture. The first picture of a first bitstream of a resolution may be the first dummy picture (and after that dummy pictures follow at every other coded picture), whereas the second picture of another bitstream of the same resolution may be the first dummy picture (and after that dummy pictures follow at every other coded picture).

[0275] MCTSs 97 from the high-resolution bitstreams may be selected for example to cover a viewport, and MCTSs 98 from the low-resolution bitstreams may be selected to cover the remaining area covered by the omnidirectional video.

[0276] A single-time coded picture set may be formed from each pair of coded pictures in the four bitstreams, where, in each pair, one of the coded pictures is a dummy picture. The first picture 99a in a single-time coded picture set may be extracted from the bitstreams where the second picture of the bitstream is a dummy picture, and the second picture 99b in a single-time coded picture set may be extracted from the bitstreams where the first picture of the bitstream is a dummy picture.

[0277] The pictures 99a, 99b of the single-time coded picture set have a tile grid of 3x1 (with tile column widths equal to 1920, 1920, and 768 pixels from left to right and tile row height equal to 1920 pixels). Each tile originating from a low-resolution bitstream is encapsulated in its own slice. The slice at the bottom-right corner can contain any content.

[0278] It should be understood that there are other alternatives for encoding and merging than those illustrated in Figure 6g. For example, the order of slices encapsulating in the third tile column may differ.

[0279] The picture size in the presented example is 4608x1920, which complies with the constraints of HEVC Level 5.1. Consequently, HEVC Level 5.1 decoders are capable of decoding the merged bitstream and produce a monoscopic output picture sequence at 30 Hz. The presented example can be considered to provide an 8K effective resolution at 30 Hz with the decoding capability of 4K picture resolution at 60 Hz.

[0280] In the following some signaling aspects will be discussed.

[0281] It is noted that bitstream and file format level signaling may need to be appended for the properties of the extracted spatiotemporal frame packed bitstream. The signaling may need to indicate on picture basis that which view does the picture represent or whether it spatially multiplexes two views, and that region-wise packing is applied for the picture.

[0282] In an embodiment, region-wise packing signaling, such as region-wise packing SEI message and/or region-wise packing box in file format metadata, is appended to include information that is region-wise indicative of the view. For example, a syntax element indicating whether a packed or projected region is from the left view or the right view may be included per each packed or projected region. With such amendment, two projected pictures may be generated for each time instant of stereoscopic video, one per each view. Thus, it might not be necessary to consider spatially frame-packed projected pictures or include the signaling of their spatial frame packing format in or along the bitstream.

[0283] For example, region-wise packing box may comprise syntax such as the following, where left_view_flag[i] equal to 0 specifies that the packed or projected region is from the right view and equal to 1 specifies that the packed or projected region is from the left view. num_regions indicates the number of packed regions or projected regions in a packed picture or projected picture, respectively. proj_picture_width and proj_picture_height indicate the width and height of the projected picture e.g. in relative projected picture sample units.

packed_picture_width and packed_picture_height indicate the width and height of the packed picture e.g. in relative packed picture sample units or in luma sample units of the decoded picture. RectRegionPacking(i) is a syntax structure specifying the mapping between a packed region and a projected region. GuardBandStruct(i) is a syntax structure specifying the guard bands for a packed region.

```
aligned(8) class RegionWisePackingStruct {
   unsigned int(8) num_regions;
   unsigned int(16) proj_picture_width;
   unsigned int(16) proj_picture_height;
   unsigned int(16) packed_picture_width;
   unsigned int(16) packed_picture_height;
   for (i = 0; i < num_regions; i++) {
      bit(2) reserved = 0;
      unsigned int(1) left_view_flag[i];
      unsigned int(1) guard_band_flag[i];
      unsigned int(4) packing_type[i];
      if (packing_type[i] == 0) {
         RectRegionPacking(i);
         if (guard_band_flag[i])
            GuardBandStruct(i);
      }
   }
}
```

[0284] In an embodiment, region-wise packing signaling, such as region-wise packing SEI message and/or region-wise packing box in file format metadata, is separately indicated for each picture in a single-time coded picture set. For example, region-wise packing box may comprise syntax such as the following, where num_pics specifies the number of pictures in a single-time coded picture set.

```
aligned(8) class SingleTimeCodedPictureSet {
   unsigned int(8) num_pics;
   for (k = 0; k < num_pics; k++)
      RegionWisePackingStruct();
}
```

[0285] In an embodiment, the same single-time coded picture set structure may be repetitively applied to associate pictures of the bitstream to each picture within the single-time coded picture set. For example, if the single-time coded picture set consists of two pictures, the first picture of the single-time coded picture set may be associated with the first, third, fifth, etc. picture of the bitstream, and the second picture of the single-timecoded picture set may be associated with the second, fourth, sixth, etc. picture of the bitstream. The association may be used to conclude which region-wise packing parameters apply for pictures of the bitstream. The repetitive association may be restarted according to pre-defined rules. For example, the repetitive association may be restarted as the start of each coded video sequence (e.g., at an IDR picture of an H.264/AVC bitstream) or at a sync sample and/or SAP sample of particular types (e.g. a SAP type in the range of 1 to 3, inclusive) as indicated by the file format metadata for a video track according to ISOBMFF.

[0286] A video bitstream may be encapsulated into and/or decapsulated from a track having a restricted video ('resv') sample entry of the ISO base media file format. The SchemeInformationBox in the sample entry may act as the container structure for indicating spatial arrangements. For example, the SchemeInformationBox may comprise one or more of a box for indicating the omnidirectional projection format (e.g. called ProjectionFormatBox) and a box for indicating the region-wise packing arrangement (e.g. RegionWisePackingBox).

[0287] The container structure for indicating spatial arrangements may be generated into and/or parsed from a streaming manifest, such as DASH MPD. The container structure may, for example, be an AdaptationSet, Representation, or SubRepresentation element, or may be a specific element within an AdaptationSet, Representation, or SubRepresentation element. For example, the containing element may comprise one or more of an essential property descriptor for indicating the omnidirectional projection format and an essential property descriptor for indicating the region-wise packing arrangement.

[0288] In an embodiment, region-wise packing is indicated as a sample group of ISOBMFF. A sample group description entry may comprise RegionWisePackingStruct as described above. SampleToGroupBox associates each sample with a sample group description entry and hence indicates which region-wise packing parameters apply to the sample.

[0289] Region-wise packing information included in any indication above may describe a region-wise process applied to a source frame (e.g. a projected frame) to obtain a destination frame (e.g. a packed frame) that has been encoded or that is generated by following instructions, such as extractors, included in a track. Alternatively, region-wise packing information included in any indication above may describe a region-wise process to be applied to a decoded frame (e.g. a packed frame) to obtain a rearranged frame (e.g. a projected picture). The latter region-wise process may also be referred to as region-wise back-mapping.

[0290] Region-wise back-mapping may be specified or implemented as a process that maps regions of a packed picture to a projected picture. Metadata may be included in or along the bitstream that describes the region-wise mapping of a projected picture to a packed picture or vice versa. For example, a mapping of a source rectangle of a projected picture to a destination rectangle in a packed picture may be included in such metadata. The width and height of the source rectangle in relation to the width and height of the destination rectangle, respectively, may indicate a horizontal and vertical resampling ratio, respectively. A back-mapping process maps samples of the destination rectangle (as indicated in the metadata) of the packed picture to the source rectangle (as indicated in the metadata) of an output projected picture. The back-mapping process may include resampling according to the width and height ratios of the source and destination rectangles.

[0291] According to an embodiment, a separate piece of spatial arrangement or region-wise packing signaling, such as region-wise packing SEI message and/or region-wise packing box in file format metadata, is parsed for each picture in a single-time coded picture set. The piece applicable for each decoded picture is concluded. When repetitive association as described above is used, it is concluded when the repetitive association is restarted according to pre-defined rules. For example, the repetitive association may be restarted at the start of each coded video sequence (e.g., at an IDR picture of an H.264/AVC bitstream) or at a sync sample and/or SAP sample of particular types (e.g. a SAP type in the range of 1 to 3, inclusive) as indicated by the file format metadata for a video track according to ISOBMFF. The piece of spatial arrangement or region-wise packing signaling is decoded from or along the bitstream. The spatial arrangement signaling may, for example, comprise one or more of omnidirectional projection and a region-wise packing arrangement. The decoded spatial arrangement may be processed to obtain reconstructed picture(s) of a particular format. For example, region-wise back-mapping may be applied to obtain a monoscopic or stereoscopic projected picture. The obtained reconstructed picture(s) of a particular format may further be processed, e.g. by extracting a viewport that is displayed, or using the reconstructed picture(s) to be overlaid on a triangular rendering mesh used in the displaying process.

[0292] A track comprising instructions to generate a spatiotemporal frame-packed bitstream may be indicated as a first Representation in a DASH MPD. The source bitstreams for generating the spatiotemporal frame-packed bitstreams may be indicated as other Representations in the DASH MPD. The first Representation may be indicated to depend on the other Representations using the @dependencyId attribute or using the DASH preselection feature, where the Adaptation Set containing the first Representation may be indicated to be the main Adaptation Set for the preselection and the Adaptation Sets containing the other Representations may be indicated to be the partial Adaptation Sets for the preselection.

[0293] A client may parse the availability of the first Representation (as described in the previous paragraph) and other Representations (e.g. for spatiotemporal bitstreams for other viewport orientations and/or sizes) from a DASH MPD. Other metadata, such as spatial arrangement as described above and/or region-wise quality ranking, may also be parsed for the Representation(s) from the DASH MPD. Based on the metadata and viewing orientation (and other viewing conditions) and network conditions (e.g. estimated received bitrate), a DASH client may determine which Representation(s) best suits its purpose and request the (Sub)segments of the best-suited Representation and any Representations that the best-suited Representation depends on.

[0294] Temporal motion-constrained tile set signaling may be indicated to apply to a sub-sequence of pictures. Several sub-sequences may be defined for the same spatiotemporal frame-packed bitstream. For example, a sub-sequence may be defined for a first view of the high-resolution bitstream, for the first views and the second view of the high-resolution bitstream, and for each of the low-resolution bitstream (or likewise the spatially merged low-resolution tiles).

[0295] Conventionally the above types of signaling apply in sequence-level, i.e., apply equally to each picture in the sequence.

[0296] In an embodiment, which may be applied together with or independently of other embodiments, an encoder or a bitstream merger generates information, such as an SEI message, into or along the bitstream, wherein the information indicates tiles and slices that form a motion-constrained area with similar constraints as explained earlier for MCTS. When a tile comprises multiple slices, the information may indicate which slices of the tile are part of the motion-constrained area. When a slice comprises multiple tiles that are all part of the same motion-constrained area, the information may indicate each of these tiles or may indicate the slice. A slice may be indicated, for example, with a slice address, which may be an enumerated position of the first block of the slice along a certain scan order. For example, the slice_segment_address of HEVC or similar may be used as an identification of a slice. In an embodiment, which may be applied together with or independently of other embodiments, a decoder or a file parser or alike decodes information, such as an SEI message, from or along the bitstream, wherein the information indicates tiles and slices that form a motion-constrained area with similar constraints as explained earlier for MCTS.

[0297] Embodiments described above for MCTSs may be similarly applied to motion-constrained areas that comprise tiles and slices.

[0298] In an embodiment, which may be applied together with or independently of other embodiments, an encoder or a bitstream merger generates information, such as an SEI message, into or along the bitstream, wherein the information indicates a partitioning grid, on a block grid (such as a CTU grid). The partitioning grid may but need not be the same as the tile grid. Thus, each cell in the partitioning grid includes a positive integer number of blocks (such as CTUs). Alternatively, the partitioning grid may be inferred to be same as the block grid. The information further indicates a motion-constrained area with similar constraints as explained earlier for MCTS. The motion-constrained area is indicated relative to the partitioning grid, for example by indicating a top-left position, width, and height on the partitioning grid, or by indicating a top-left position and a bottom-right position. The motion-constrained area may be constrained to be rectangular. In an embodiment, which may be applied together with or independently of other embodiments, a decoder or a file parser or alike decodes information, such as an SEI message, from or along the bitstream, wherein a partitioning grid indication is decoded from the information or inferred, and the information indicates a motion-constrained area relative to the partitioning grid with similar constraints as explained earlier for MCTS.

[0299] Embodiments described above for MCTSs may be similarly applied to motion-constrained areas that are indicated relative to a partitioning grid.

[0300] While embodiments have been described above with reference to motion-constrained tile sets or motion-constrained slices, it needs to be understood that embodiments can similarly be realized by splitting the input picture sequence to sub-picture sequences along the tile grid, and encoding each sub-picture sequence independently, wherein the encoding is constrained not to use motion vectors over picture boundaries.

[0301] While embodiments have been described above with reference to creating an extractor track that contains instructing for tile merging, it needs to be understood that embodiments can be similarly realized by executing the tile merging without extractor tracks.

[0302] While embodiments have been described above with reference to content authoring (including encoding and file encapsulation), it needs to be understood that similar embodiments can be authored for the client side.

[0303] The video encoding method according to an example embodiment will now be described with reference to the simplified block diagram of Figure 9a and the flow diagram of Figure 10. The elements of Figure 9a may, for example, be implemented by the first encoder section 500 of the encoder of Figure 8a, or they may be separate from the first encoder section 500.

[0304] The encoder section 500 receives 521 a first input picture sequence 511a and a second input picture sequence 511b. Pictures of the first and second picture sequence are in the same projection format which may be e.g. of a cube map projection format or equirectangular projection format. The first input picture sequence has a first resolution whereas the second input picture sequence 511b has a second resolution which is lower than the resolution of the first input picture sequence.

[0305] An encoding element 513 encodes 523 the first input picture sequence and the second input picture sequence into a single-time coded picture set. The single-time coded picture set comprises first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

[0306] The bitstream merging method according to an example embodiment will now be described with reference to the simplified block diagram of Figure 9b.

[0307] The encoded bitstreams such as 90, 91a, 91b, may be provided to an extracting element 516 (Figure 9b) for preparing a spatiotemporal frame-packed bitstream 94. The spatiotemporal frame-packed bitstream may contain the coded video data and/or it may contain instructions (e.g. extractors) to extract coded video data from other bitstreams or tracks. The extracting element 516 may receive, e.g. from a client device, information 517 on the location of the current viewport of the client device so that the extracting element 516 may deduce the location of the viewport with reference to the 360 degree image. Alternatively, the extracting element 516 may consider a multitude of possible viewport orientations, and operate separately for each one of them. The extracting element 516 may then determine which tiles of the first view and the second view cover (i.e. are located within) the viewport. Hence, the extracting element 516 may select a certain rectangular area of tiles for which the higher resolution bitstream (having the first spatial size) is utilized and a remaining area of the image for which the lower resolution bitstreams (having the second spatial size) are utilized. In accordance with an embodiment, the extracting element 516 generates two or more samples, such as 93a, 93b, 93c, for each time instance of the input video, such as original 360 degree panoramic video (Figure 6d). In an example related to Figure 6d, a first sample 93a is formed on the basis of the tiles of the first view at the location of the current viewport, and a second sample 93a is formed on the basis of the tiles of the second view at the location of the current viewport, and a third sample 93a is formed by merging the lower resolution tiles of both views at the location outside the current viewport to replace the dummy picture of the high-resolution bitstream 90. The outcome of the extraction process is the spatiotemporal frame-packed bitstream is the spatiotemporal frame-packed bitstream 92, such as 94 in Figure 6d , which may be transmitted to another entity, e.g. to a network server or to a consumer end device (the client device), and/or saved to a storage device such as a memory.

[0308] The process described above may be repeated to each successive picture in the original video. The process may be real-time or may be based on previously recorded video.

[0309] When the viewport changes the client device may indicate the new viewport to the encoder where the extracting element 516 may then change the area from which high-resolution tiles are used and, respectively, the low-resolution tiles.

[0310] Figure 11 is a graphical representation of an example multimedia communication system within which various embodiments may be implemented. A data source 1510 provides a source signal in an analog, uncompressed digital, or compressed digital format, or any combination of these formats. An encoder 1520 may include or be connected with a pre-processing, such as data format conversion and/or filtering of the source signal. The encoder 1520 encodes the source signal into a coded media bitstream. It should be noted that a bitstream to be decoded may be received directly or indirectly from a remote device located within virtually any type of network. Additionally, the bitstream may be received from local hardware or software. The encoder 1520 may be capable of encoding more than one media type, such as audio and video, or more than one encoder 1520 may be required to code different media types of the source signal. The encoder 1520 may also get synthetically produced input, such as graphics and text, or it may be capable of producing coded bitstreams of synthetic media. In the following, only processing of one coded media bitstream of one media type is considered to simplify the description. It should be noted, however, that typically real-time broadcast services comprise several streams (typically at least one audio, video and text sub-titling stream). It should also be noted that the system may include many encoders, but in the figure only one encoder 1520 is represented to simplify the description without a lack of generality. It should be further understood that, although text and examples contained herein may specifically describe an encoding process, one skilled in the art would understand that the same concepts and principles also apply to the corresponding decoding process and vice versa.

[0311] The coded media bitstream may be transferred to a storage 1530. The storage 1530 may comprise any type of mass memory to store the coded media bitstream. The format of the coded media bitstream in the storage 1530 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file, or the coded media bitstream may be encapsulated into a Segment format suitable for DASH (or a similar streaming system) and stored as a sequence of Segments. If one or more media bitstreams are encapsulated in a container file, a file generator (not shown in the figure) may be used to store the one more media bitstreams in the file and create file format metadata, which may also be stored in the file. The encoder 1520 or the storage 1530 may comprise the file generator, or the file generator is operationally attached to either the encoder 1520 or the storage 1530. Some systems operate “live”, i.e. omit storage and transfer coded media bitstream from the encoder 1520 directly to the sender 1540. The coded media bitstream may then be transferred to the sender 1540, also referred to as the server, on a need basis. The format used in the transmission may be an elementary self-contained bitstream format, a packet stream format, a Segment format suitable for DASH (or a similar streaming system), or one or more coded media bitstreams may be encapsulated into a container file. The encoder 1520, the storage 1530, and the server 1540 may reside in the same physical device or they may be included in separate devices. The encoder 1520 and server 1540 may operate with live real-time content, in which case the coded media bitstream is typically not stored permanently, but rather buffered for small periods of time in the content encoder 1520 and/or in the server 1540 to smooth out variations in processing delay, transfer delay, and coded media bitrate.

[0312] The server 1540 sends the coded media bitstream using a communication protocol stack. The stack may include but is not limited to one or more of Hypertext Transfer Protocol (HTTP), Transmission Control Protocol (TCP), and Internet Protocol (IP). When the communication protocol stack is packet-oriented, the server 1540 encapsulates the coded media bitstream into packets. It should be again noted that a system may contain more than one server 1540, but for the sake of simplicity, the following description only considers one server 1540.

[0313] If the media content is encapsulated in a container file for the storage 1530 or for inputting the data to the sender 1540, the sender 1540 may comprise or be operationally attached to a "sending file parser" (not shown in the figure). In particular, if the container file is not transmitted as such but at least one of the contained coded media bitstream is encapsulated for transport over a communication protocol, a sending file parser locates appropriate parts of the coded media bitstream to be conveyed over the communication protocol. The sending file parser may also help in creating the correct format for the communication protocol, such as packet headers and payloads. The multimedia container file may contain encapsulation instructions, such as hint tracks in the ISOBMFF, for encapsulation of the at least one of the contained media bitstream on the communication protocol.

[0314] The server 1540 may or may not be connected to a gateway 1550 through a communication network, which may e.g. be a combination of a CDN, the Internet and/or one or more access networks. The gateway may also or alternatively be referred to as a middle-box. For DASH, the gateway may be an edge server (of a CDN) or a web proxy. It is noted that the system may generally comprise any number gateways or alike, but for the sake of simplicity, the following description only considers one gateway 1550. The gateway 1550 may perform different types of functions, such as translation of a packet stream according to one communication protocol stack to another communication protocol stack, merging and forking of data streams, and manipulation of data stream according to the downlink and/or receiver capabilities, such as controlling the bit rate of the forwarded stream according to prevailing downlink network conditions.

[0315] The system includes one or more receivers 1560, typically capable of receiving, de-modulating, and de-capsulating the transmitted signal into a coded media bitstream. The coded media bitstream may be transferred to a recording storage 1570. The recording storage 1570 may comprise any type of mass memory to store the coded media bitstream. The recording storage 1570 may alternatively or additively comprise computation memory, such as random access memory. The format of the coded media bitstream in the recording storage 1570 may be an elementary self-contained bitstream format, or one or more coded media bitstreams may be encapsulated into a container file. If there are multiple coded media bitstreams, such as an audio stream and a video stream, associated with each other, a container file is typically used and the receiver 1560 comprises or is attached to a container file generator producing a container file from input streams. Some systems operate “live,” i.e. omit the recording storage 1570 and transfer coded media bitstream from the receiver 1560 directly to the decoder 1580. In some systems, only the most recent part of the recorded stream, e.g., the most recent 10-minute excerption of the recorded stream, is maintained in the recording storage 1570, while any earlier recorded data is discarded from the recording storage 1570.

[0316] The coded media bitstream may be transferred from the recording storage 1570 to the decoder 1580. If there are many coded media bitstreams, such as an audio stream and a video stream, associated with each other and encapsulated into a container file or a single media bitstream is encapsulated in a container file e.g. for easier access, a file parser (not shown in the figure) is used to decapsulate each coded media bitstream from the container file. The recording storage 1570 or a decoder 1580 may comprise the file parser, or the file parser is attached to either recording storage 1570 or the decoder 1580. It should also be noted that the system may include many decoders, but here only one decoder 1570 is discussed to simplify the description without a lack of generality

[0317] The coded media bitstream may be processed further by a decoder 1570, whose output is one or more uncompressed media streams. Finally, a renderer 1590 may reproduce the uncompressed media streams with a loudspeaker or a display, for example. The receiver 1560, recording storage 1570, decoder 1570, and renderer 1590 may reside in the same physical device or they may be included in separate devices.

[0318] A sender 1540 and/or a gateway 1550 may be configured to perform switching between different representations e.g. for view switching, bitrate adaptation and/or fast start-up, and/or a sender 1540 and/or a gateway 1550 may be configured to select the transmitted representation(s). Switching between different representations may take place for multiple reasons, such as to respond to requests of the receiver 1560 or prevailing conditions, such as throughput, of the network over which the bitstream is conveyed. A request from the receiver can be, e.g., a request for a Segment or a Subsegment from a different representation than earlier, a request for a change of transmitted scalability layers and/or sub-layers, or a change of a rendering device having different capabilities compared to the previous one. A request for a Segment may be an HTTP GET request. A request for a Subsegment may be an HTTP GET request with a byte range. Additionally or alternatively, bitrate adjustment or bitrate adaptation may be used for example for providing so-called fast start-up in streaming services, where the bitrate of the transmitted stream is lower than the channel bitrate after starting or random-accessing the streaming in order to start playback immediately and to achieve a buffer occupancy level that tolerates occasional packet delays and/or retransmissions. Bitrate adaptation may include multiple representation or layer up-switching and representation or layer down-switching operations taking place in various orders.

[0319] A decoder 1580 may be configured to perform switching between different representations e.g. for view switching, bitrate adaptation and/or fast start-up, and/or a decoder 1580 may be configured to select the transmitted representation(s). Switching between different representations may take place for multiple reasons, such as to achieve faster decoding operation or to adapt the transmitted bitstream, e.g. in terms of bitrate, to prevailing conditions, such as throughput, of the network over which the bitstream is conveyed. Faster decoding operation might be needed for example if the device including the decoder 580 is multi-tasking and uses computing resources for other purposes than decoding the scalable video bitstream. In another example, faster decoding operation might be needed when content is played back at a faster pace than the normal playback speed, e.g. twice or three times faster than conventional real-time playback rate. The speed of decoder operation may be changed during the decoding or playback for example as response to changing from a fast-forward play from normal playback rate or vice versa, and consequently multiple layer up-switching and layer down-switching operations may take place in various orders.

[0320] In the above, many embodiments have been described with reference to the equirectangular projection format. It needs to be understood that embodiments similarly apply to equirectangular pictures where the vertical coverage is less than 180 degrees. For example, the covered elevation range may be from −75° to 75°, or from −60° to 90° (i.e., covering one both not both poles). It also needs to be understood that embodiments similarly cover horizontally segmented equirectangular projection format, where a horizontal segment covers an azimuth range of 360 degrees and may have a resolution potentially differing from the resolution of other horizontal segments. Furthermore, it needs to be understood that embodiments similarly apply to omnidirectional picture formats, where a first sphere region of the content is represented by the equirectangular projection of limited elevation range and a second sphere region of the content is represented by another projection, such as cube map projection. For example, the elevation range −45° to 45° may be represented by a "middle" region of equirectangular projection, and the other sphere regions may be represented by a rectilinear projection, similar to cube faces of a cube map but where the corners overlapping with the middle region on the spherical domain are cut out. In such cases, embodiments can be applied to the middle region represented by the equirectangular projection.

[0321] In the above, some embodiments have been described with reference to terminology of particular codecs, most notably HEVC. It needs to be understood that embodiments can be similarly realized with respective terms of other codecs. For example, rather than tiles or tile sets, embodiments could be realized with rectangular slice groups of H.264/AVC.

[0322] The phrase along the bitstream (e.g. indicating along the bitstream) may be used in claims and described embodiments to refer to out-of-band transmission, signaling, or storage in a manner that the out-of-band data is associated with the bitstream. The phrase decoding along the bitstream or alike may refer to decoding the referred out-of-band data (which may be obtained from out-of-band transmission, signaling, or storage) that is associated with the bitstream.

[0323] The phrase along the track (e.g. including, along a track, a description of a motion-constrained coded sub-picture sequence) may be used in claims and described embodiments to refer to out-of-band transmission, signaling, or storage in a manner that the out-of-band data is associated with the track. In other words, the phrase "a description along the track" may be understood to mean that the description is not stored in the file or segments that carry the track, but within another resource, such as a media presentation description. For example, the description of the motion-constrained coded sub-picture sequence may be included in a media presentation description that includes information of a Representation conveying the track. The phrase decoding along the track or alike may refer to decoding the referred out-of-band data (which may be obtained from out-of-band transmission, signaling, or storage) that is associated with the track.

[0324] In the above, some embodiments have been described with reference to segments, e.g. as defined in MPEG-DASH. It needs to be understood that embodiments may be similarly realized with subsegments, e.g. as defined in MPEG-DASH.

[0325] In the above, some embodiments have been described in relation to DASH or MPEG-DASH. It needs to be understood that embodiments could be similarly realized with any other similar streaming system, and/or any similar protocols as those used in DASH, and/or any similar segment and/or manifest formats as those used in DASH, and/or any similar client operation as that of a DASH client. For example, some embodiments could be realized with the M3U manifest format.

[0326] In the above, some embodiments have been described in relation to ISOBMFF, e.g. when it comes to segment format. It needs to be understood that embodiments could be similarly realized with any other file format, such as Matroska, with similar capability and/or structures as those in ISOBMFF.

[0327] In the above, some embodiments have been described with reference to encoding or including indications or metadata in the bitstream and/or decoding indications or metadata from the bitstream. It needs to be understood that indications or metadata may additionally or alternatively be encoded or included along the bitstream and/or decoded along the bitstream. For example, indications or metadata may be included in or decoded from a container file that encapsulates the bitstream.

[0328] In the above, some embodiments have been described with reference to including metadata or indications in or along a container file and/or parsing or decoding metadata and/or indications from or along a container file. It needs to be understood that indications or metadata may additionally or alternatively be encoded or included in the video bitstream, for example as SEI message(s) or VUI, and/or decoded in the video bitstream, for example from SEI message(s) or VUI.

[0329] The following describes in further detail suitable apparatus and possible mechanisms for implementing the embodiments of the invention. In this regard reference is first made to Figure 12 which shows a schematic block diagram of an exemplary apparatus or electronic device 50 depicted in Figure 13, which may incorporate a transmitter according to an embodiment of the invention.

[0330] The electronic device 50 may for example be a mobile terminal or user equipment of a wireless communication system. However, it would be appreciated that embodiments of the invention may be implemented within any electronic device or apparatus which may require transmission of radio frequency signals.

[0331] The apparatus 50 may comprise a housing 30 for incorporating and protecting the device. The apparatus 50 further may comprise a display 32 in the form of a liquid crystal display. In other embodiments of the invention the display may be any suitable display technology suitable to display an image or video. The apparatus 50 may further comprise a keypad 34. In other embodiments of the invention any suitable data or user interface mechanism may be employed. For example the user interface may be implemented as a virtual keyboard or data entry system as part of a touch-sensitive display. The apparatus may comprise a microphone 36 or any suitable audio input which may be a digital or analogue signal input. The apparatus 50 may further comprise an audio output device which in embodiments of the invention may be any one of: an earpiece 38, speaker, or an analogue audio or digital audio output connection. The apparatus 50 may also comprise a battery 40 (or in other embodiments of the invention the device may be powered by any suitable mobile energy device such as solar cell, fuel cell or clockwork generator). The term battery discussed in connection with the embodiments may also be one of these mobile energy devices. Further, the apparatus 50 may comprise a combination of different kinds of energy devices, for example a rechargeable battery and a solar cell. The apparatus may further comprise an infrared port 41 for short range line of sight communication to other devices. In other embodiments the apparatus 50 may further comprise any suitable short range communication solution such as for example a Bluetooth wireless connection or a USB/FireWire wired connection.

[0332] The apparatus 50 may comprise a controller 56 or processor for controlling the apparatus 50. The controller 56 may be connected to memory 58 which in embodiments of the invention may store both data and/or may also store instructions for implementation on the controller 56. The controller 56 may further be connected to codec circuitry 54 suitable for carrying out coding and decoding of audio and/or video data or assisting in coding and decoding carried out by the controller 56.

[0333] The apparatus 50 may further comprise a card reader 48 and a smart card 46, for example a universal integrated circuit card (UICC) reader and a universal integrated circuit card for providing user information and being suitable for providing authentication information for authentication and authorization of the user at a network.

[0334] The apparatus 50 may comprise radio interface circuitry 52 connected to the controller and suitable for generating wireless communication signals for example for communication with a cellular communications network, a wireless communications system or a wireless local area network. The apparatus 50 may further comprise an antenna 60 connected to the radio interface circuitry 52 for transmitting radio frequency signals generated at the radio interface circuitry 52 to other apparatus(es) and for receiving radio frequency signals from other apparatus(es).

[0335] In some embodiments of the invention, the apparatus 50 comprises a camera 42 capable of recording or detecting imaging.

[0336] With respect to Figure 14, an example of a system within which embodiments of the present invention can be utilized is shown. The system 10 comprises multiple communication devices which can communicate through one or more networks. The system 10 may comprise any combination of wired and/or wireless networks including, but not limited to a wireless cellular telephone network (such as a global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), long term evolution (LTE) based network, code division multiple access (CDMA) network etc.), a wireless local area network (WLAN) such as defined by any of the IEEE 802.x standards, a Bluetooth personal area network, an Ethernet local area network, a token ring local area network, a wide area network, and the Internet.

[0337] For example, the system shown in Figure 14 shows a mobile telephone network 11 and a representation of the internet 28. Connectivity to the internet 28 may include, but is not limited to, long range wireless connections, short range wireless connections, and various wired connections including, but not limited to, telephone lines, cable lines, power lines, and similar communication pathways.

[0338] The example communication devices shown in the system 10 may include, but are not limited to, an electronic device or apparatus 50, a combination of a personal digital assistant (PDA) and a mobile telephone 14, a PDA 16, an integrated messaging device (IMD) 18, a desktop computer 20, a notebook computer 22, a tablet computer. The apparatus 50 may be stationary or mobile when carried by an individual who is moving. The apparatus 50 may also be located in a mode of transport including, but not limited to, a car, a truck, a taxi, a bus, a train, a boat, an airplane, a bicycle, a motorcycle or any similar suitable mode of transport.

[0339] Some or further apparatus may send and receive calls and messages and communicate with service providers through a wireless connection 25 to a base station 24. The base station 24 may be connected to a network server 26 that allows communication between the mobile telephone network 11 and the internet 28. The system may include additional communication devices and communication devices of various types.

[0340] The communication devices may communicate using various transmission technologies including, but not limited to, code division multiple access (CDMA), global systems for mobile communications (GSM), universal mobile telecommunications system (UMTS), time divisional multiple access (TDMA), frequency division multiple access (FDMA), transmission control protocol-internet protocol (TCP-IP), short messaging service (SMS), multimedia messaging service (MMS), email, instant messaging service (IMS), Bluetooth, IEEE 802.11, Long Term Evolution wireless communication technique (LTE) and any similar wireless communication technology. A communications device involved in implementing various embodiments of the present invention may communicate using various media including, but not limited to, radio, infrared, laser, cable connections, and any suitable connection.

[0341] Although the above examples describe embodiments of the invention operating within a wireless communication device, it would be appreciated that the invention as described above may be implemented as a part of any apparatus comprising a circuitry in which radio frequency signals are transmitted and received. Thus, for example, embodiments of the invention may be implemented in a mobile phone, in a base station, in a computer such as a desktop computer or a tablet computer comprising radio frequency communication means (e.g. wireless local area network, cellular radio, etc.).

[0342] In general, the various embodiments of the invention may be implemented in hardware or special purpose circuits or any combination thereof. While various aspects of the invention may be illustrated and described as block diagrams or using some other pictorial representation, it is well understood that these blocks, apparatus, systems, techniques or methods described herein may be implemented in, as non-limiting examples, hardware, software, firmware, special purpose circuits or logic, general purpose hardware or controller or other computing devices, or some combination thereof.

[0343] Embodiments of the inventions may be practiced in various components such as integrated circuit modules. The design of integrated circuits is by and large a highly automated process. Complex and powerful software tools are available for converting a logic level design into a semiconductor circuit design ready to be etched and formed on a semiconductor substrate.

[0344] Programs, such as those provided by Synopsys, Inc. of Mountain View, California and Cadence Design, of San Jose, California automatically route conductors and locate components on a semiconductor chip using well established rules of design as well as libraries of pre stored design modules. Once the design for a semiconductor circuit has been completed, the resultant design, in a standardized electronic format (e.g., Opus, GDSII, or the like) may be transmitted to a semiconductor fabrication facility or "fab" for fabrication.

[0345] The foregoing description has provided by way of exemplary and non-limiting examples a full and informative description of the exemplary embodiment of this invention. However, various modifications and adaptations may become apparent to those skilled in the relevant arts in view of the foregoing description, when read in conjunction with the accompanying drawings and the appended claims. However, all such and similar modifications of the teachings of this invention will still fall within the scope of this invention.

CLAIMS

1. A method comprising:

generating a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

2. The method of claim 1 further comprising:

obtaining two or more coded bitstreams, each comprising a set of two or more coded pictures out of which one picture represents actual picture content of the first time instance and the other pictures are dummy pictures that need not represent actual picture content,

merging the sets of two or more coded pictures into the single-time coded picture set in a manner that dummy pictures are omitted.

3. The method of claim 1 or 2 further comprising:

encoding, into or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format.

4. An apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform at least:

generate a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

5. The apparatus of claim 4, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to:

obtain two or more coded bitstreams, each comprising a set of two or more coded pictures out of which one picture represents actual picture content of the first time instance and the other pictures are dummy pictures that need not represent actual picture content; and

merge the sets of two or more coded pictures into the single-time coded picture set in a manner that dummy pictures are omitted.

6. The apparatus of claim 4 or 5, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to:

encode, into or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format.

7. A computer readable storage medium comprising code for use by an apparatus, which when executed by a processor, causes the apparatus to perform:

generate a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

8. An apparatus comprising:

means for generating a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that cover a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set.

9. An apparatus according to claim 8 comprising means for performing the method of any of the claims 2 or 3.

10. A method comprising:

obtaining a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decoding, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decoding a coded picture of the single-time coded picture set into a decoded picture;

using the separate region-wise packing information for the decoded picture in displaying the decoded picture.

11. The method of claim 10 further comprising:

decoding, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format.

12. An apparatus comprising at least one processor and at least one memory, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform at least:

obtain a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decode, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decode a coded picture of the single-time coded picture set into a decoded picture;

use the separate region-wise packing information for the decoded picture in displaying the decoded picture.

13. The apparatus of claim 12, said at least one memory stored with code thereon, which when executed by said at least one processor, causes the apparatus to perform the method of claim 11.

14. A computer readable storage medium comprising code for use by an apparatus, which when executed by a processor, causes the apparatus to perform:

obtain a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

decode, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

decode a coded picture of the single-time coded picture set into a decoded picture;

use the separate region-wise packing information for the decoded picture in displaying the decoded picture.

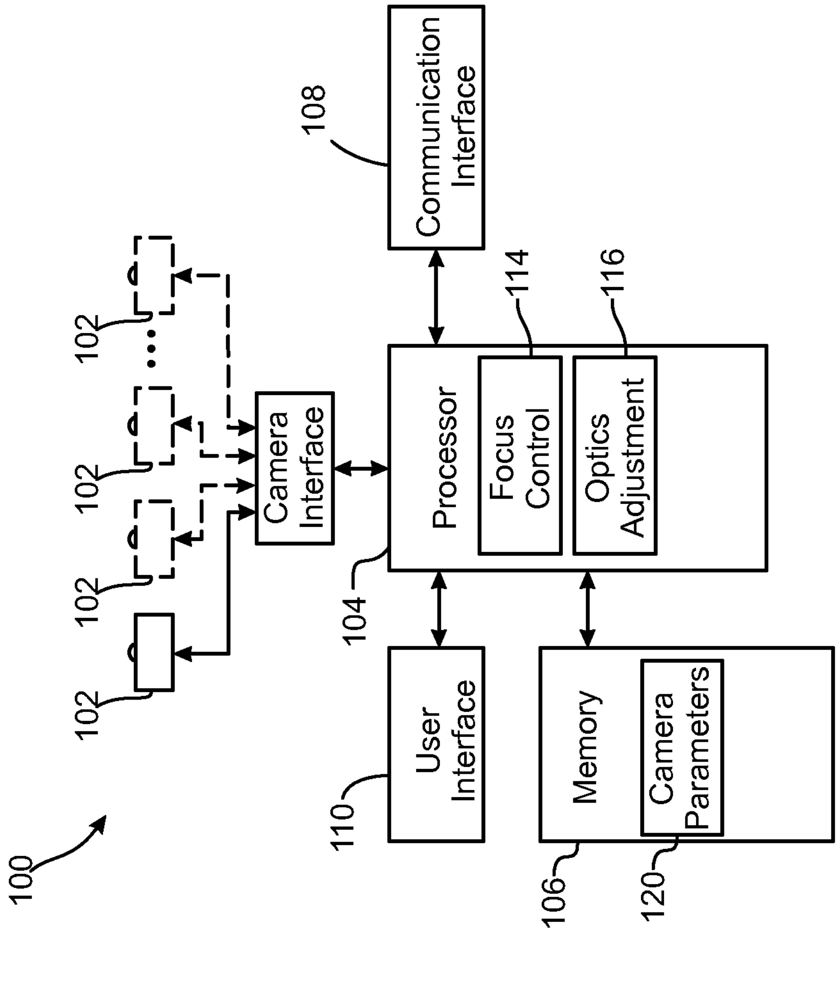
15. An apparatus comprising:

means for obtaining a single-time coded picture set comprising first two or more coded pictures representing at least one input picture of a first time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions so that a subset of the regions that covers a viewport has higher resolution, wherein the viewport represents a displayed portion of decoded regions of the single-time coded picture set;

means for decoding, from or along the single-time coded picture set, separate region-wise packing information for each picture of the single-time coded picture set, wherein the region-wise packing information indicates the sample locations within a projected picture for regions of a decoded picture, wherein the projected picture complies with a certain omnidirectional projection format;

means for decoding a coded picture of the single-time coded picture set into a decoded picture;

means for using the separate region-wise packing information for the decoded picture in displaying the decoded picture.


100
102
102
102
102
Camera Interface
Processor
Focus Control
Optics Adjustment
114
116
104
108
Communication Interface
110
User Interface
Memory
Camera Parameters
106
120

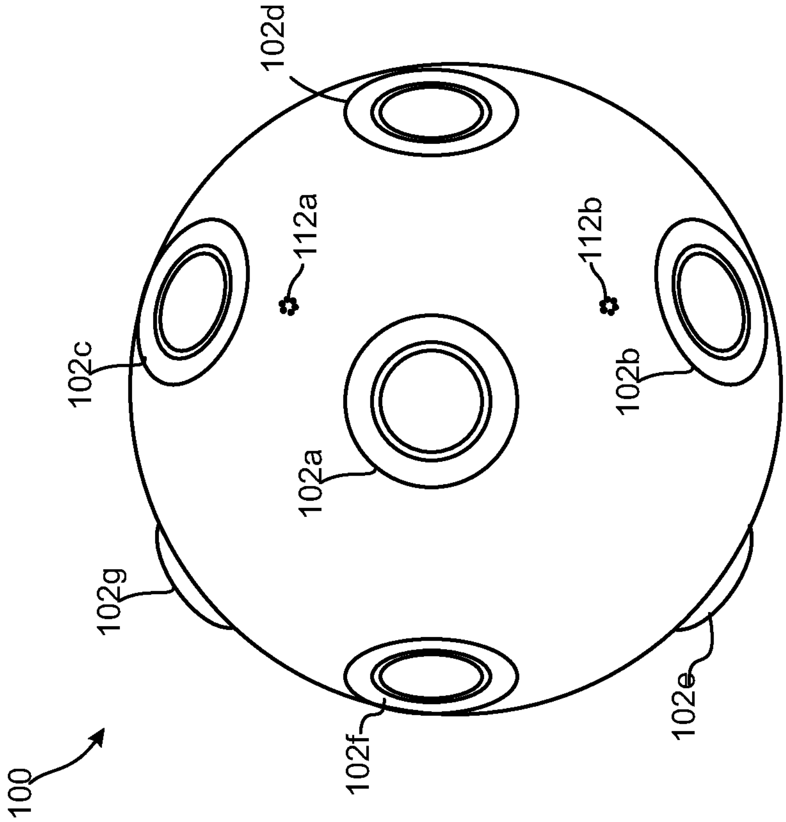

100
102a
102b
102c
102d
102e
102f
102g
112a
112b

Fig. 1aFig. 1b

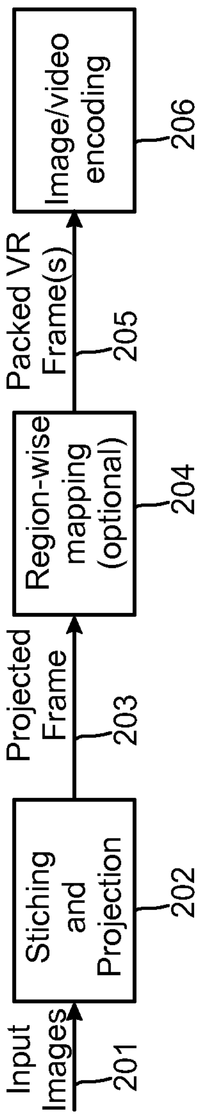

Input Images
201
Stiching and Projection
202
Projected Frame
203
Region-wise mapping (optional)
204
Packed VR Frame(s)
205
Image/video encoding
206

Fig. 2a

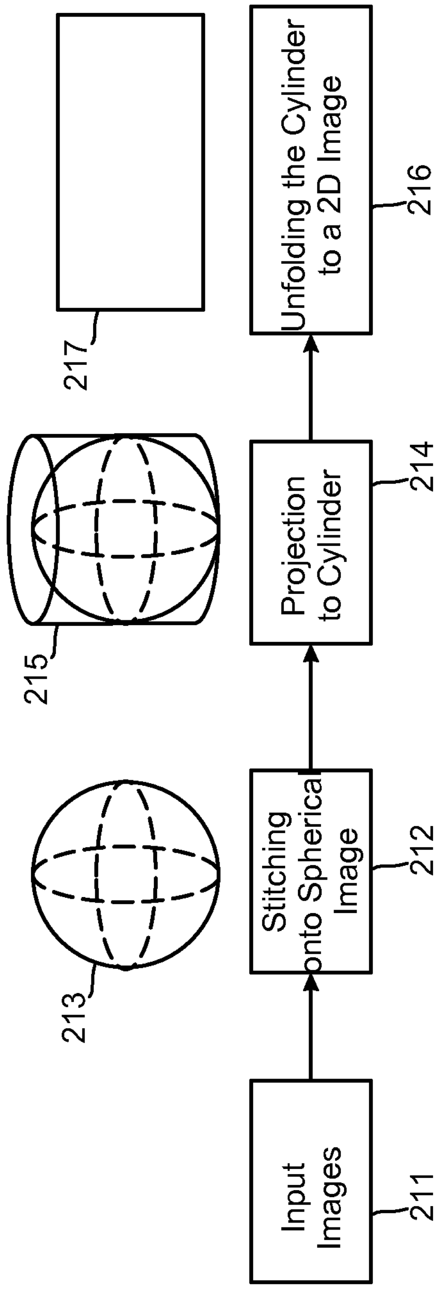

Input Images
211
Stitching onto Spherical Image
212
213
Projection to Cylinder
214
215
Unfolding the Cylinder to a 2D Image
216
217

Fig. 2b

Fig. 3

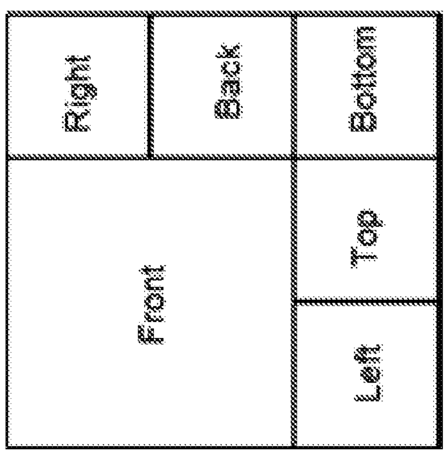
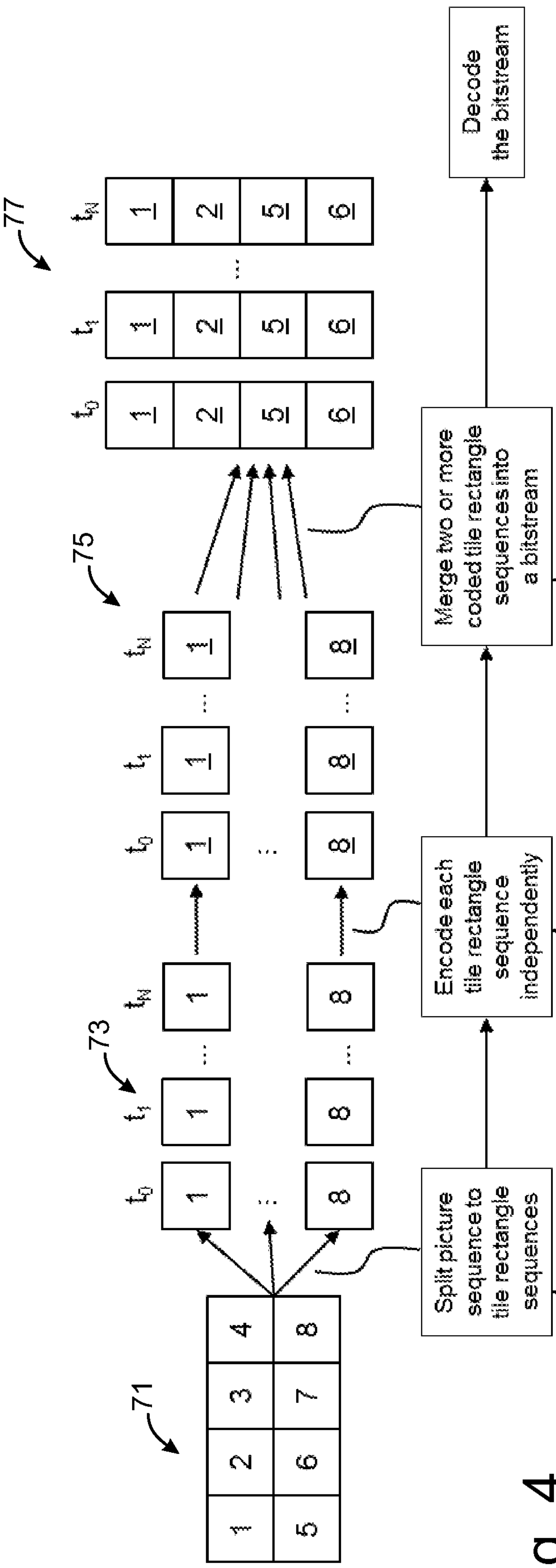

Front
Right
Back
Bottom
Top
Left

Fig. 4


71
1
2
3
4
5
6
7
8
72
Split picture sequence to tile rectangle sequences
73
74
Encode each tile rectangle sequence independently
75
76
Merge two or more coded tile rectangle sequences into a bitstream
77
Decode the bitstream

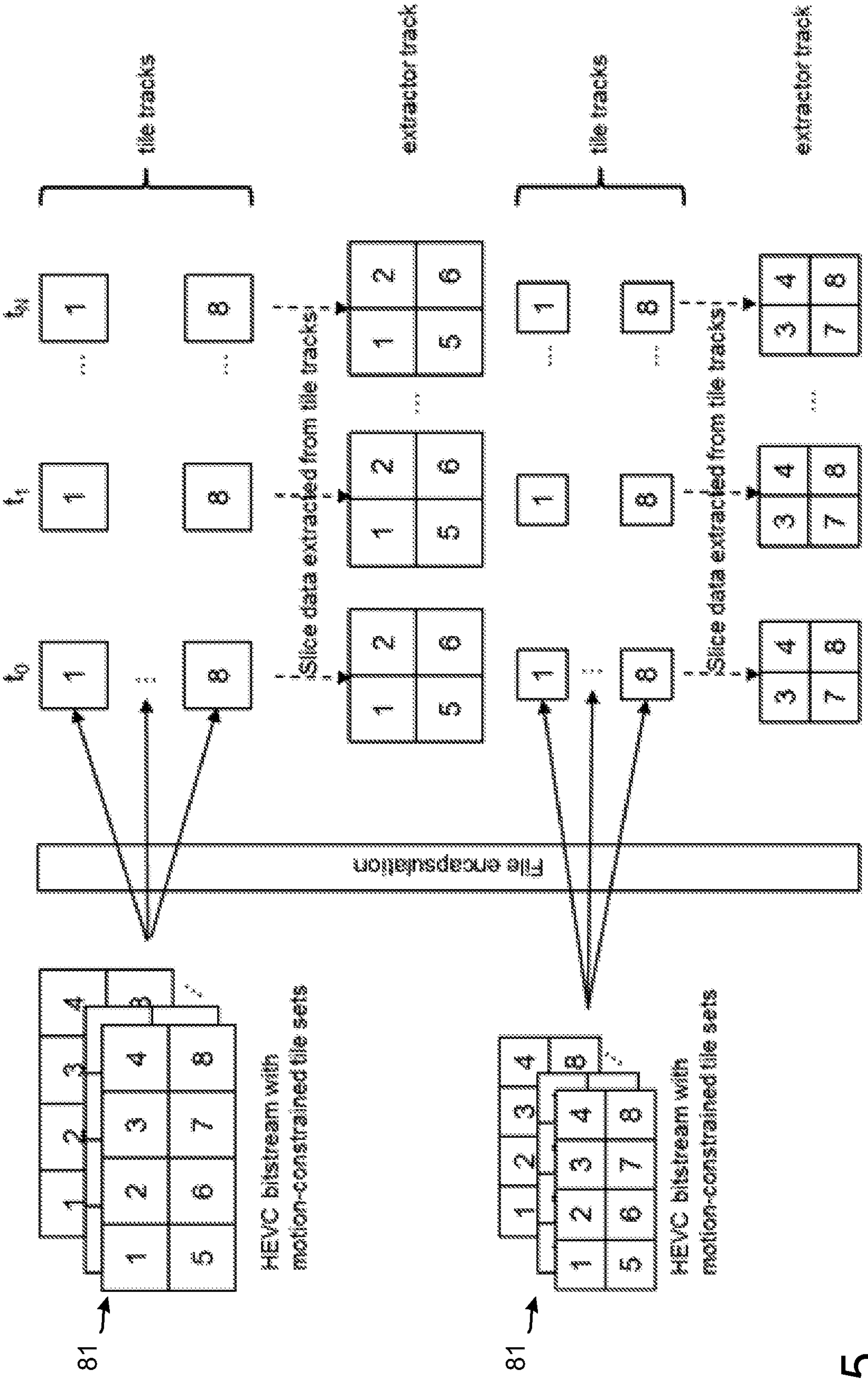

tile tracks
extractor track
tile tracks
extractor track
Slice data extracted from tile tracks
Slice data extracted from tile tracks
File encapsulation
HEVC bitstream with motion-constrained tile sets
HEVC bitstream with motion-constrained tile sets
81
81

Fig. 5

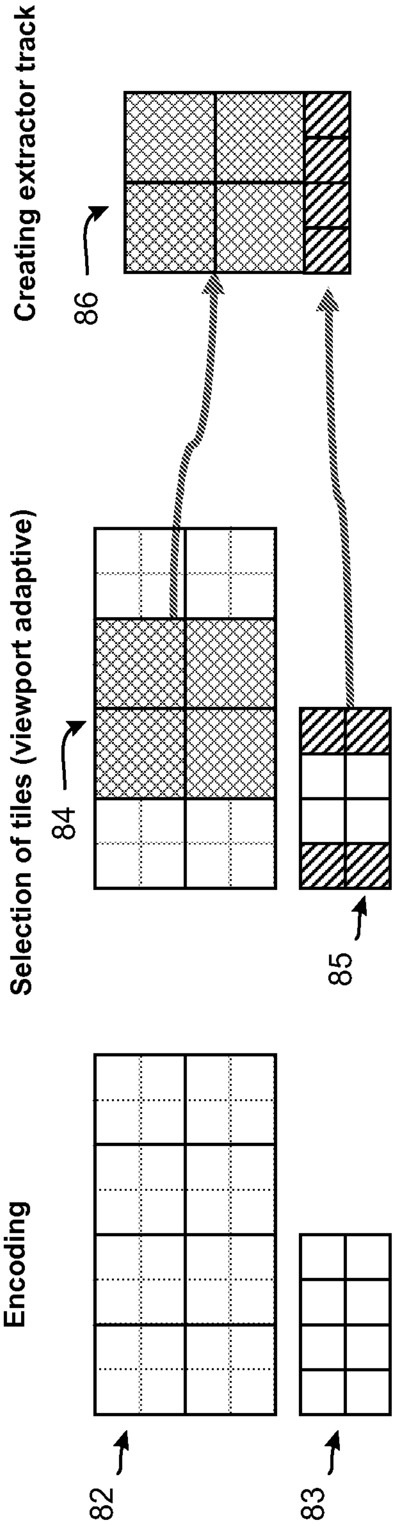

Encoding
Selection of tiles (viewport adaptive)
Creating extractor track
82
83
84
85
86

Fig. 6a

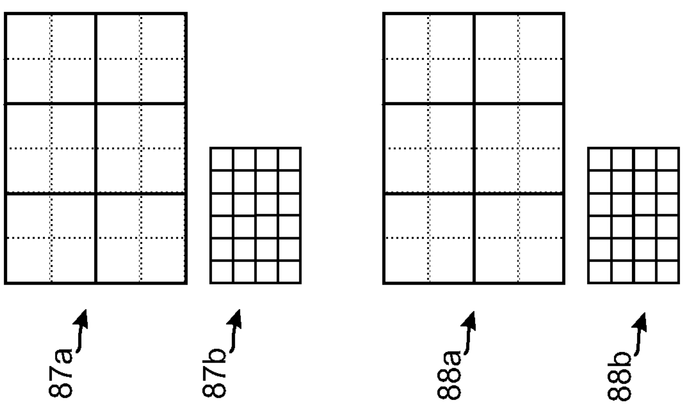

87a
87b
88a
88b

Fig. 6b

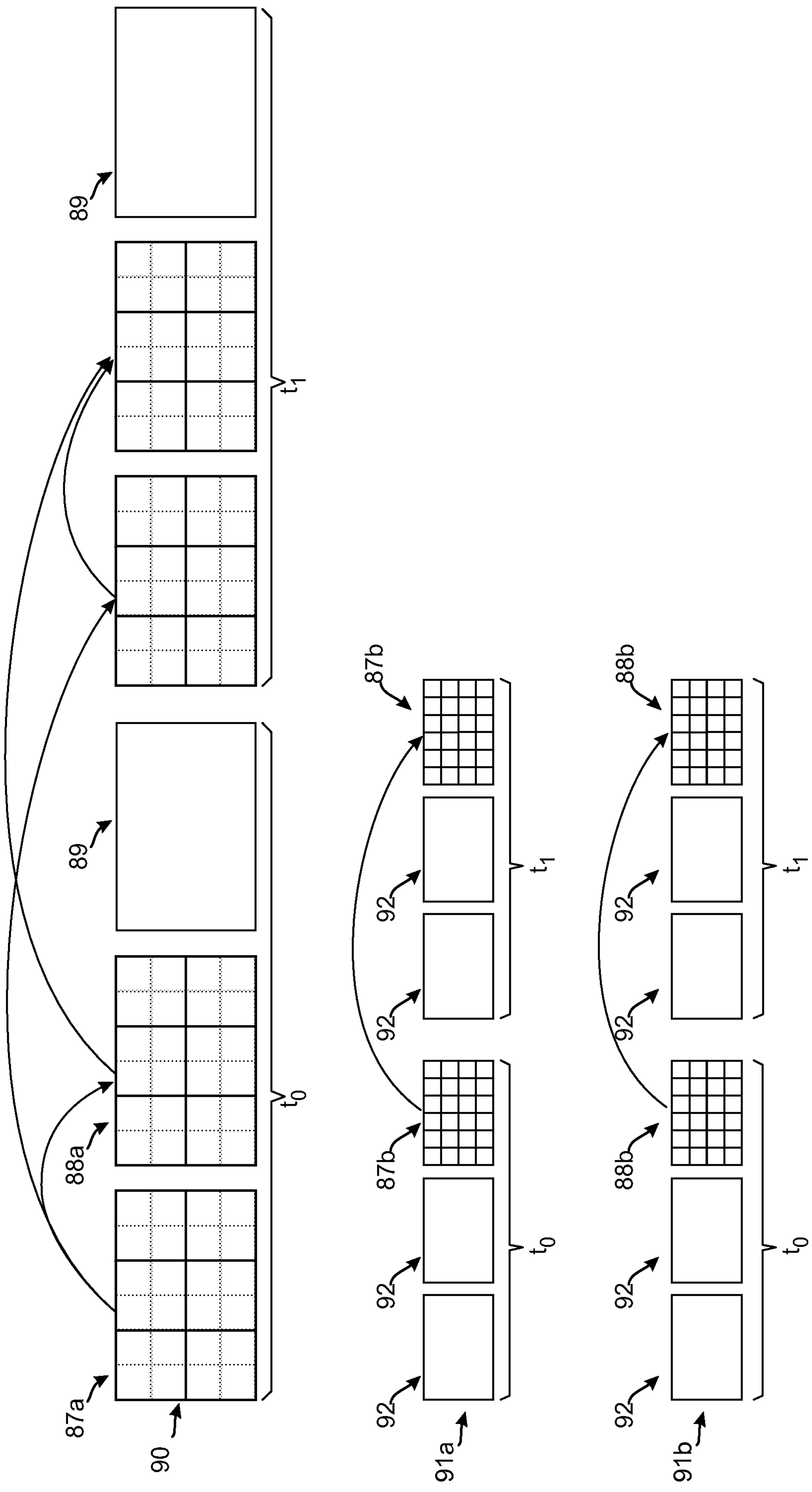

89
t1
89
88a
t0
87a
90
87b
92
t1
87b
92
t0
91a
88b
92
t1
88b
92
t0
91b

Fig. 6c

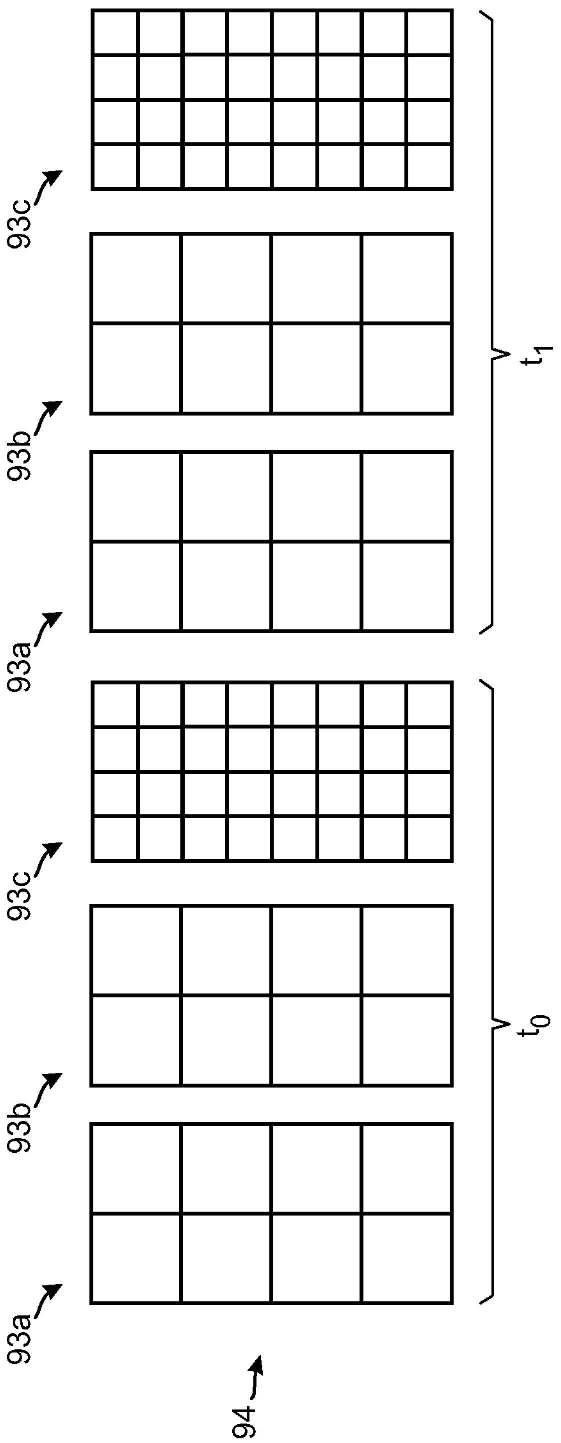


Fig. 6d

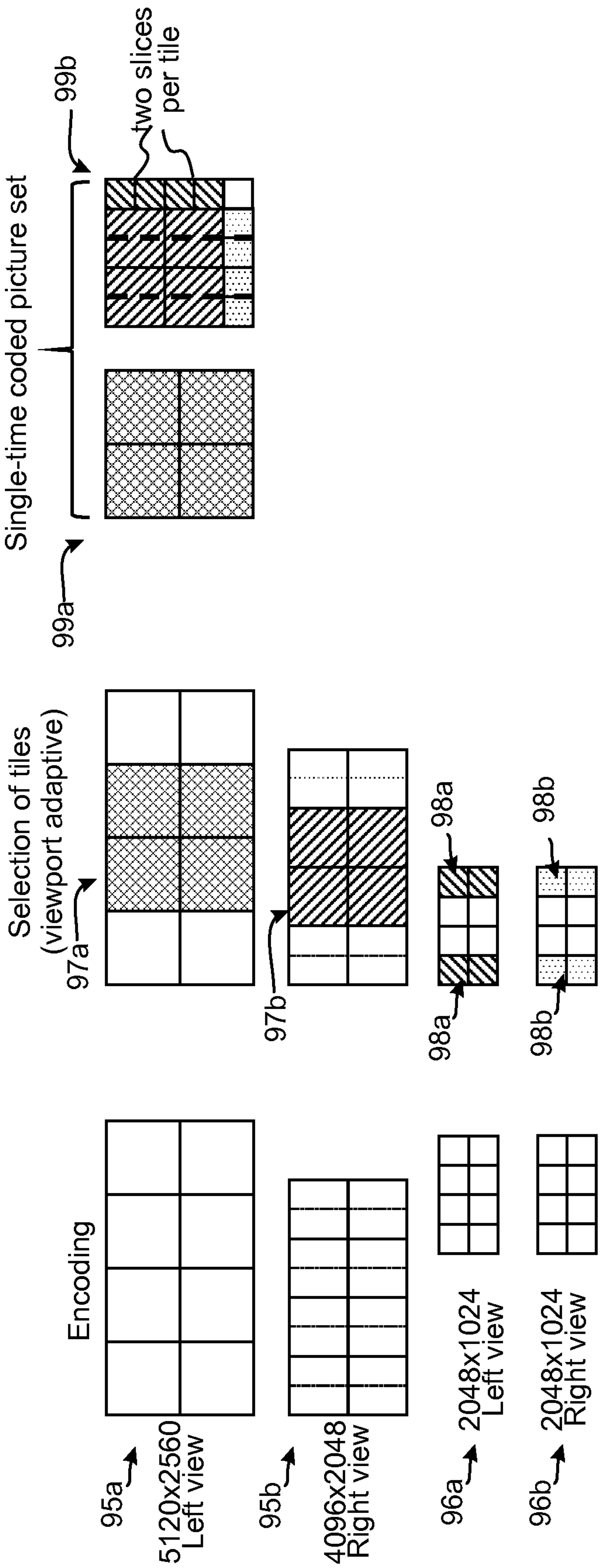

Encoding
5120x2560
Left view
95a
4096x2048
Right view
95b
2048x1024
Left view
96a
2048x1024
Right view
96b
Selection of tiles
(viewport adaptive)
97a
97b
98a
98a
98b
98b
Single-time coded picture set
99a
99b
two slices
per tile

Fig. 6e

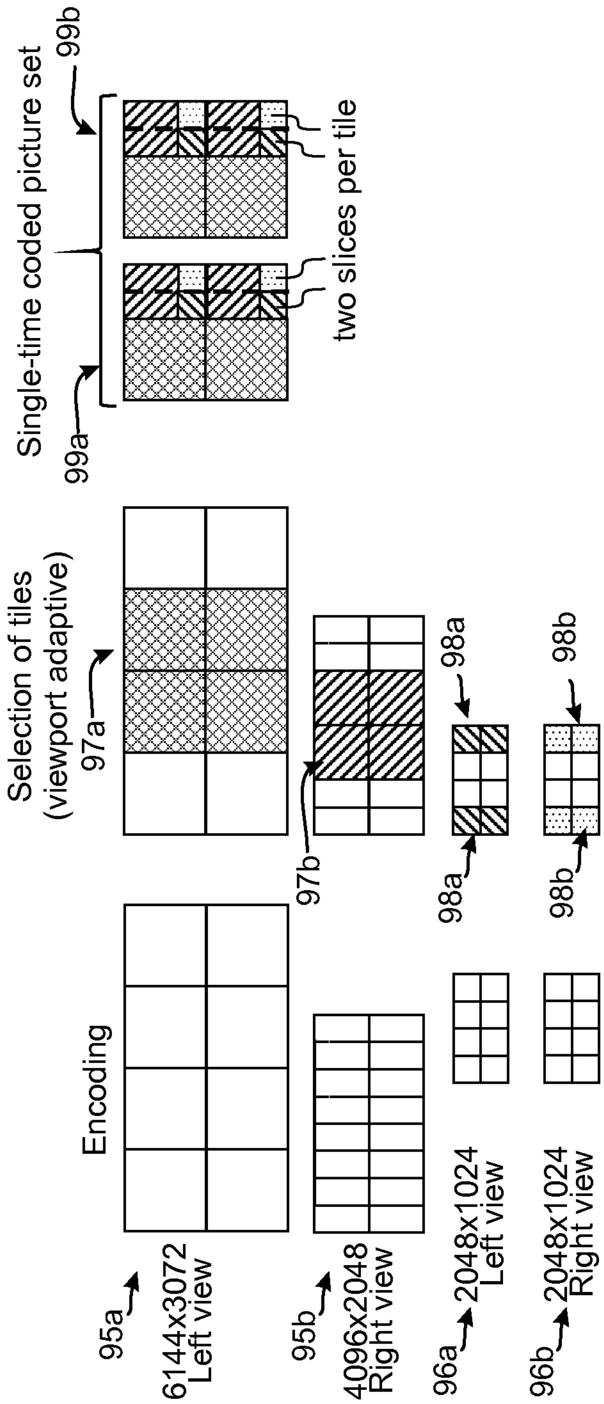

Encoding
Selection of tiles (viewport adaptive)
Single-time coded picture set
95a
6144x3072
Left view
95b
4096x2048
Right view
96a
2048x1024
Left view
96b
2048x1024
Right view
97a
97b
98a
98b
99a
99b
two slices per tile

Fig. 6f

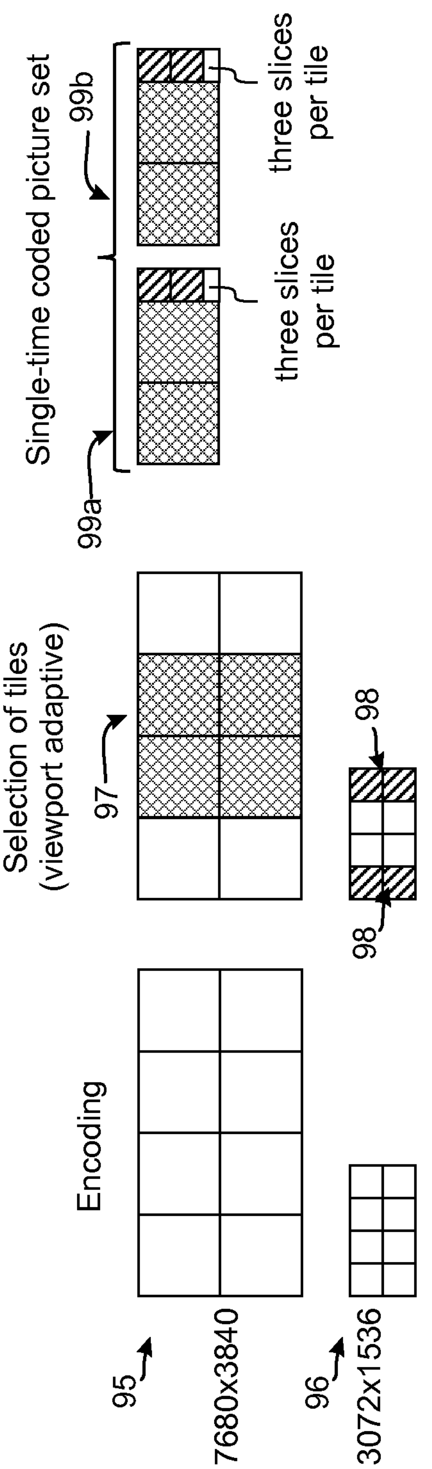

Encoding
Selection of tiles
(viewport adaptive)
Single-time coded picture set
95
7680x3840
96
3072x1536
97
98
98
99a
99b
three slices
per tile
three slices
per tile

Fig. 6g

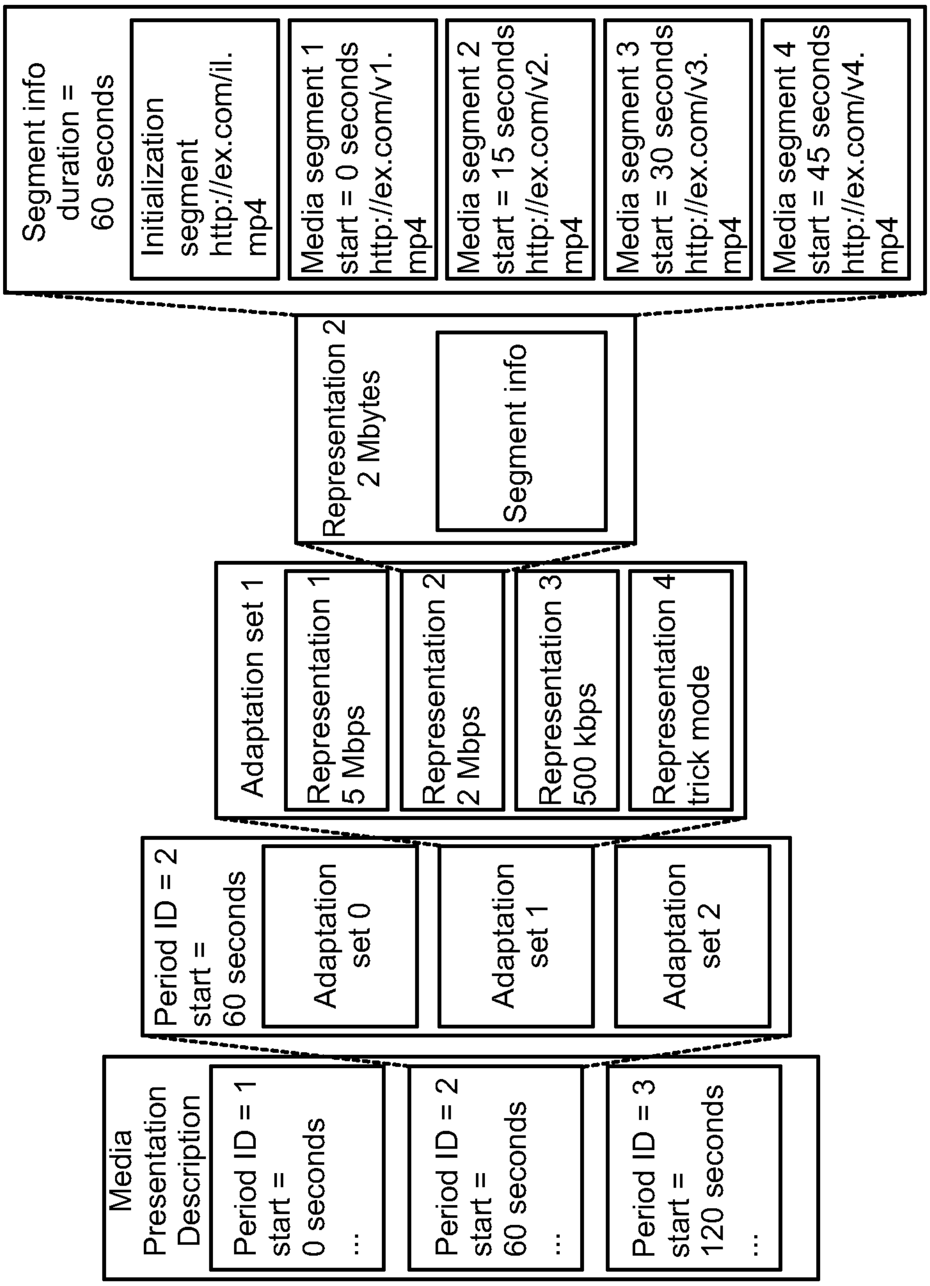

Media Presentation Description
Period ID = 1 start = 0 seconds ...
Period ID = 2 start = 60 seconds ...
Period ID = 3 start = 120 seconds ...
Period ID = 2 start = 60 seconds
Adaptation set 0
Adaptation set 1
Adaptation set 2
Adaptation set 1
Representation 1 5 Mbps
Representation 2 2 Mbps
Representation 3 500 kbps
Representation 4 trick mode
Representation 2 2 Mbytes
Segment info
Segment info duration = 60 seconds
Initialization segment http://ex.com/il.mp4
Media segment 1 start = 0 seconds http://ex.com/v1.mp4
Media segment 2 start = 15 seconds http://ex.com/v2.mp4
Media segment 3 start = 30 seconds http://ex.com/v3.mp4
Media segment 4 start = 45 seconds http://ex.com/v4.mp4

Fig. 7

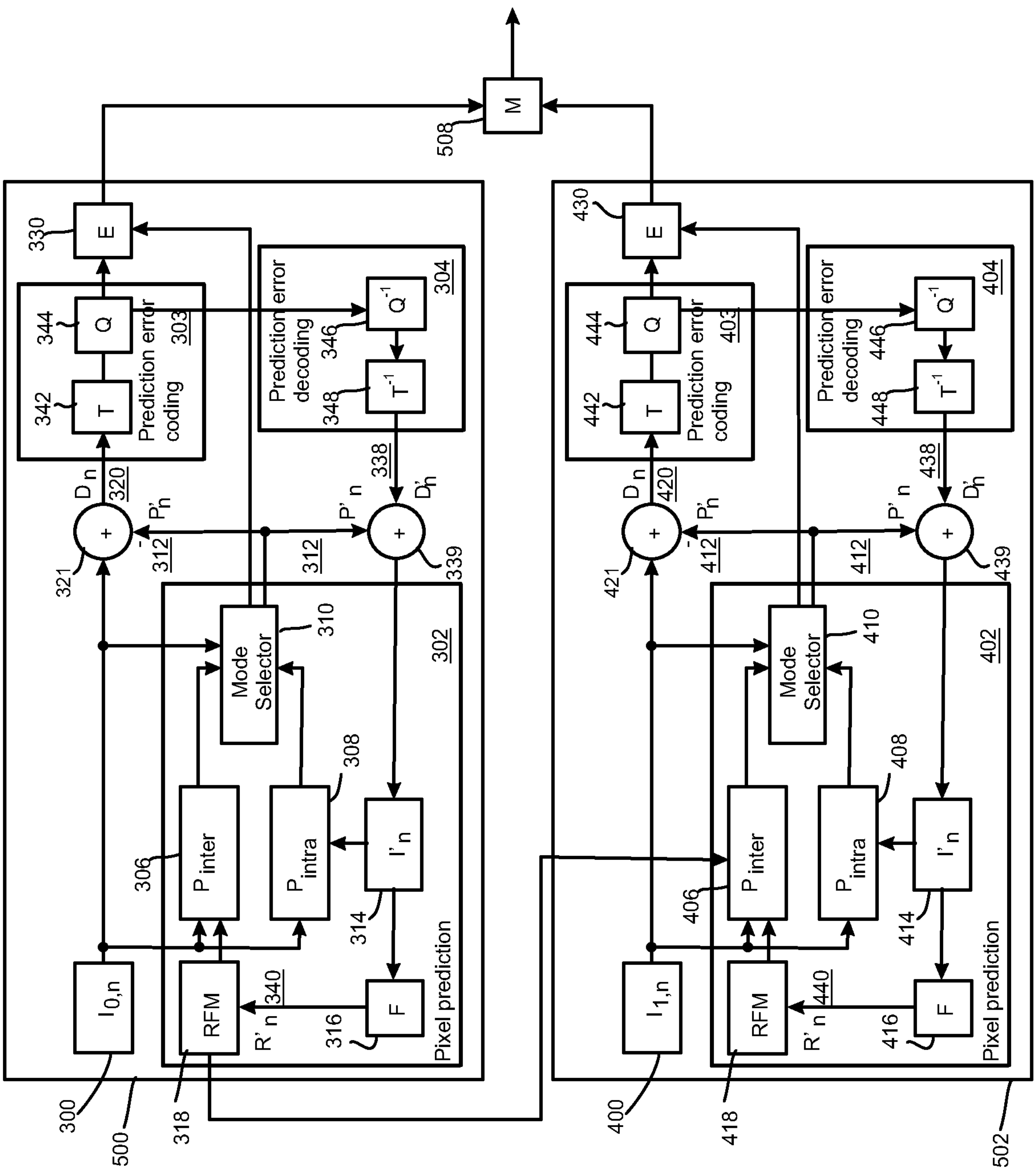

M
508
E
330
Q
344
T
342
Prediction error coding
303
Prediction error decoding
304
346
348
338
320
321
312
339
Mode Selector
310
302
308
306
314
316
340
RFM
F
Pixel prediction
300
500
318
E
430
Q
444
T
442
403
404
446
448
438
420
421
412
439
Mode Selector
410
402
408
406
414
416
440
RFM
F
Pixel prediction
400
418
502

Fig. 8a

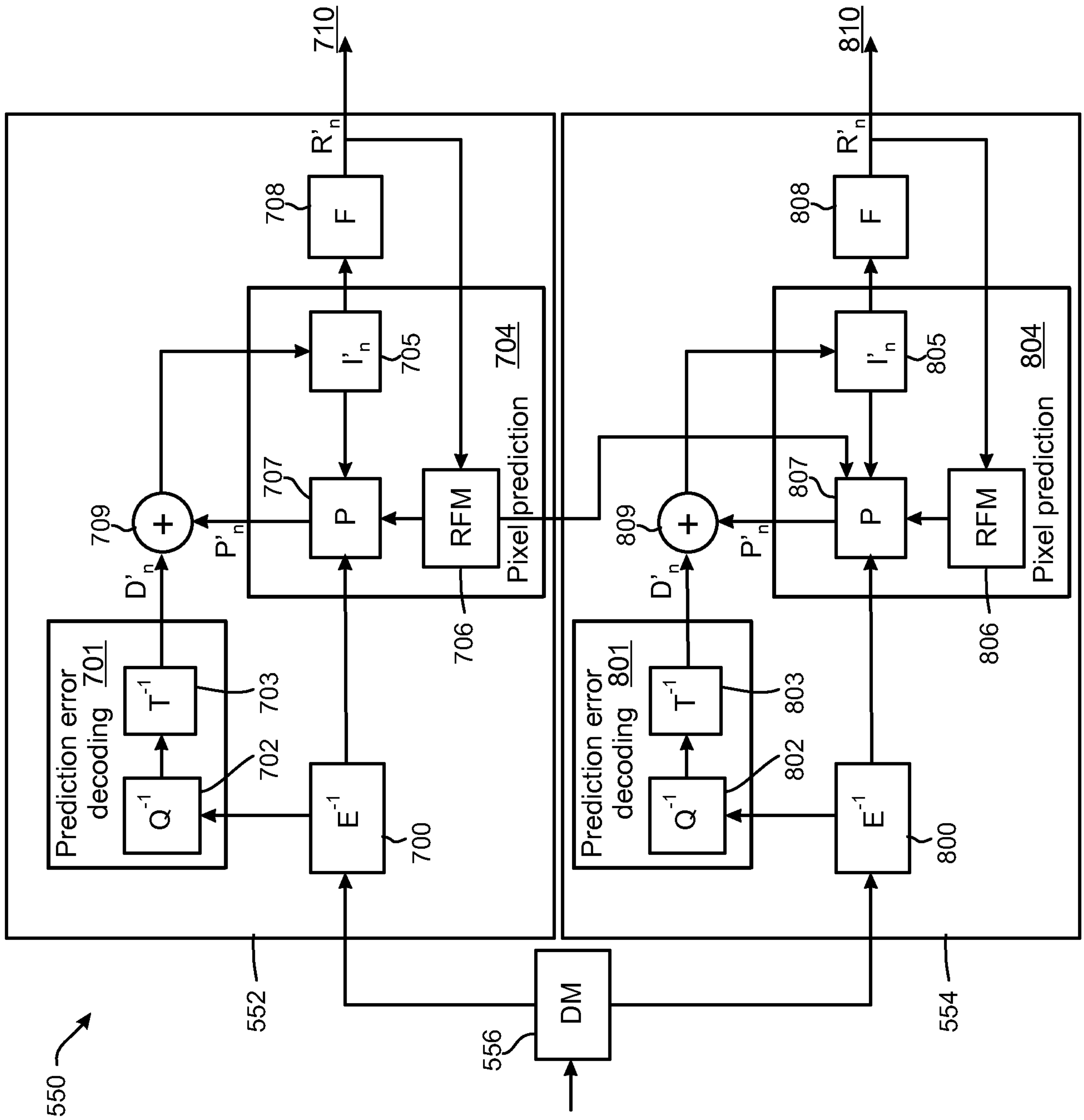

550
552
554
556
DM
E-1
700
Q-1
702
T-1
703
Prediction error decoding 701
D'n
709
+
P'n
P
707
RFM
706
I'n
705
Pixel prediction
704
F
708
R'n
710
E-1
800
Q-1
802
T-1
803
Prediction error decoding 801
D'n
809
+
P'n
P
807
RFM
806
I'n
805
Pixel prediction
804
F
808
R'n
810

Fig. 8b

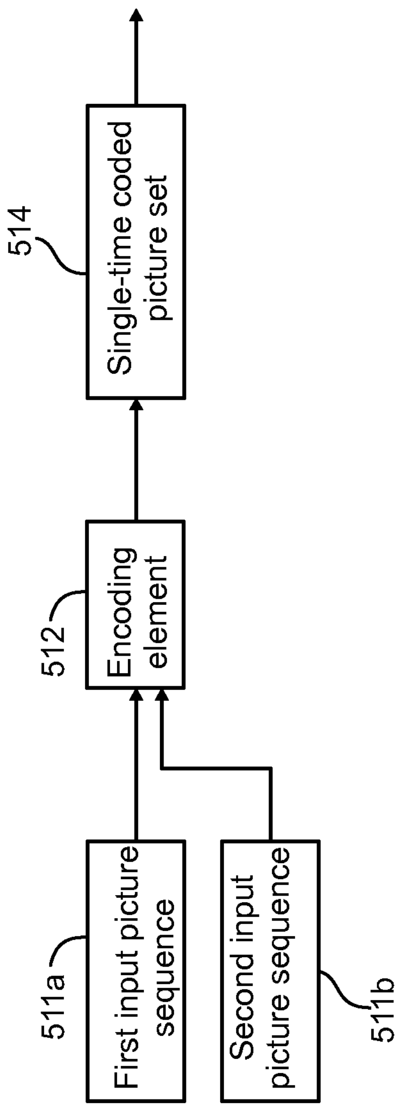


Fig. 9a

500

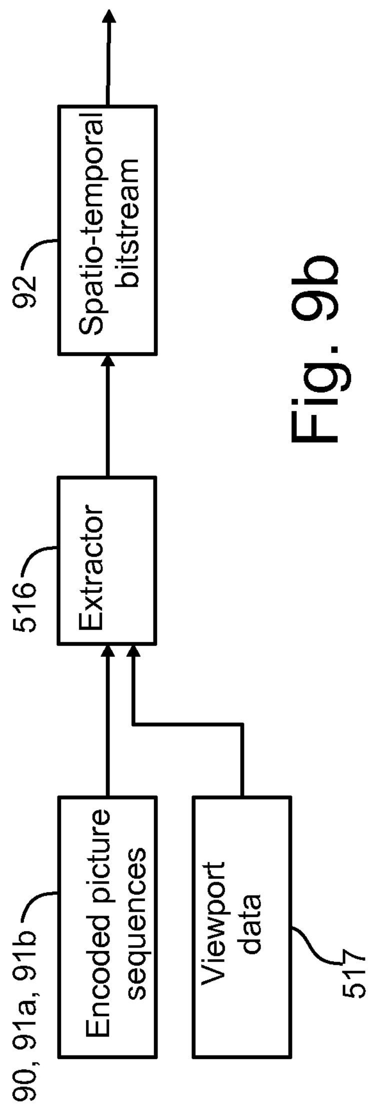


Fig. 9b

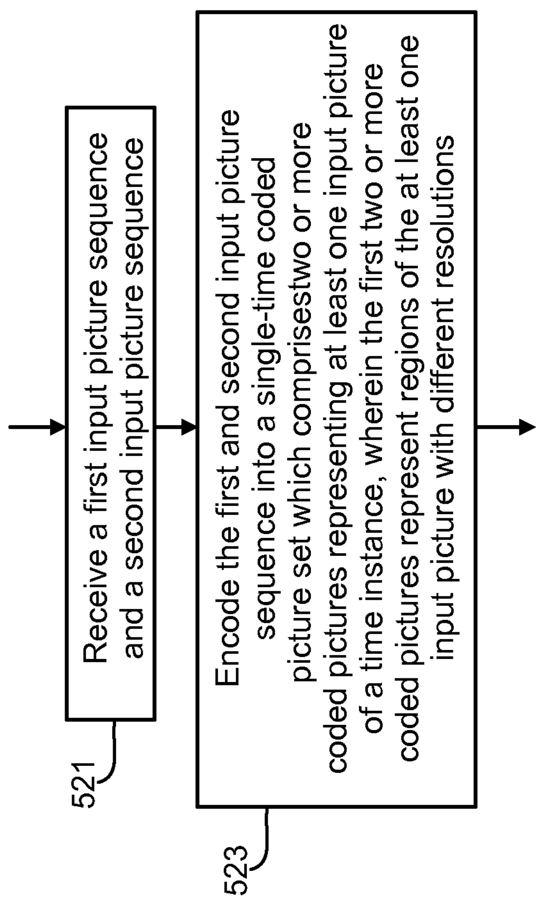

Receive a first input picture sequence and a second input picture sequence
521
Encode the first and second input picture sequence into a single-time coded picture set which comprisestwo or more coded pictures representing at least one input picture of a time instance, wherein the first two or more coded pictures represent regions of the at least one input picture with different resolutions
523

Fig. 10

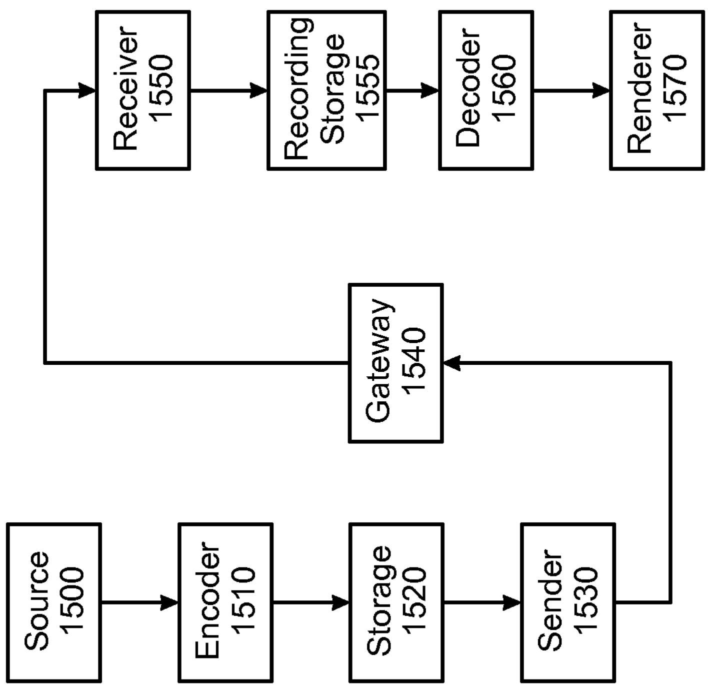

Source
1500
Encoder
1510
Storage
1520
Sender
1530
Gateway
1540
Receiver
1550
Recording
Storage
1555
Decoder
1560
Renderer
1570

Fig. 11

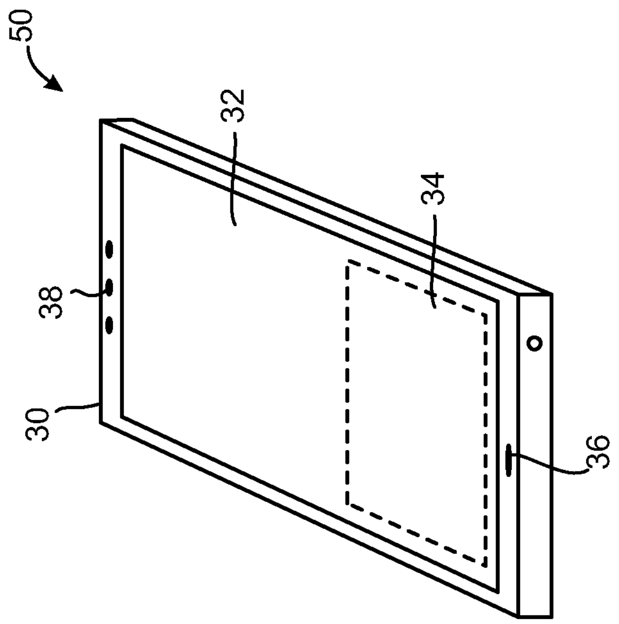

50
32
34
38
30
36

Fig. 13

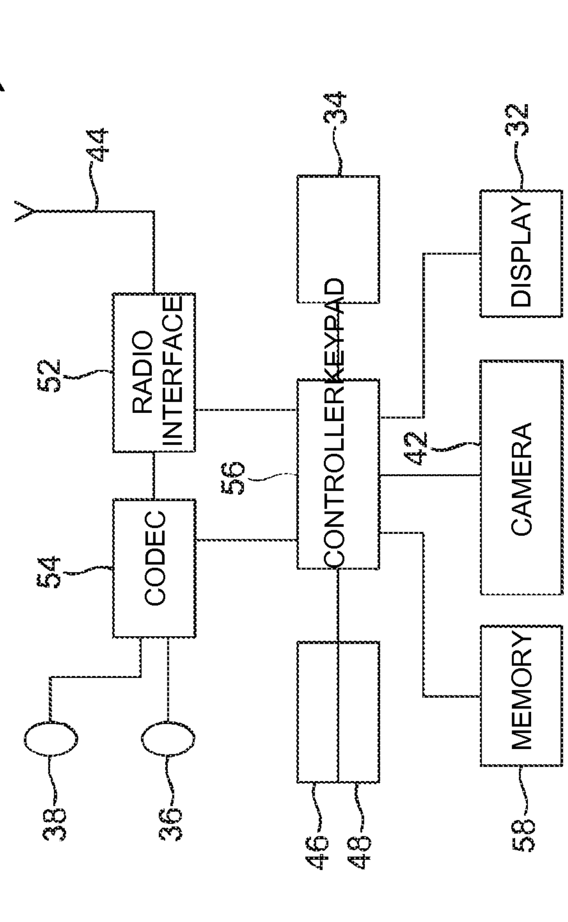

50
44
34
32
52
RADIO INTERFACE
KEYPAD
DISPLAY
56
CONTROLLER
42
CAMERA
54
CODEC
MEMORY
38
36
46
48
58

Fig. 12

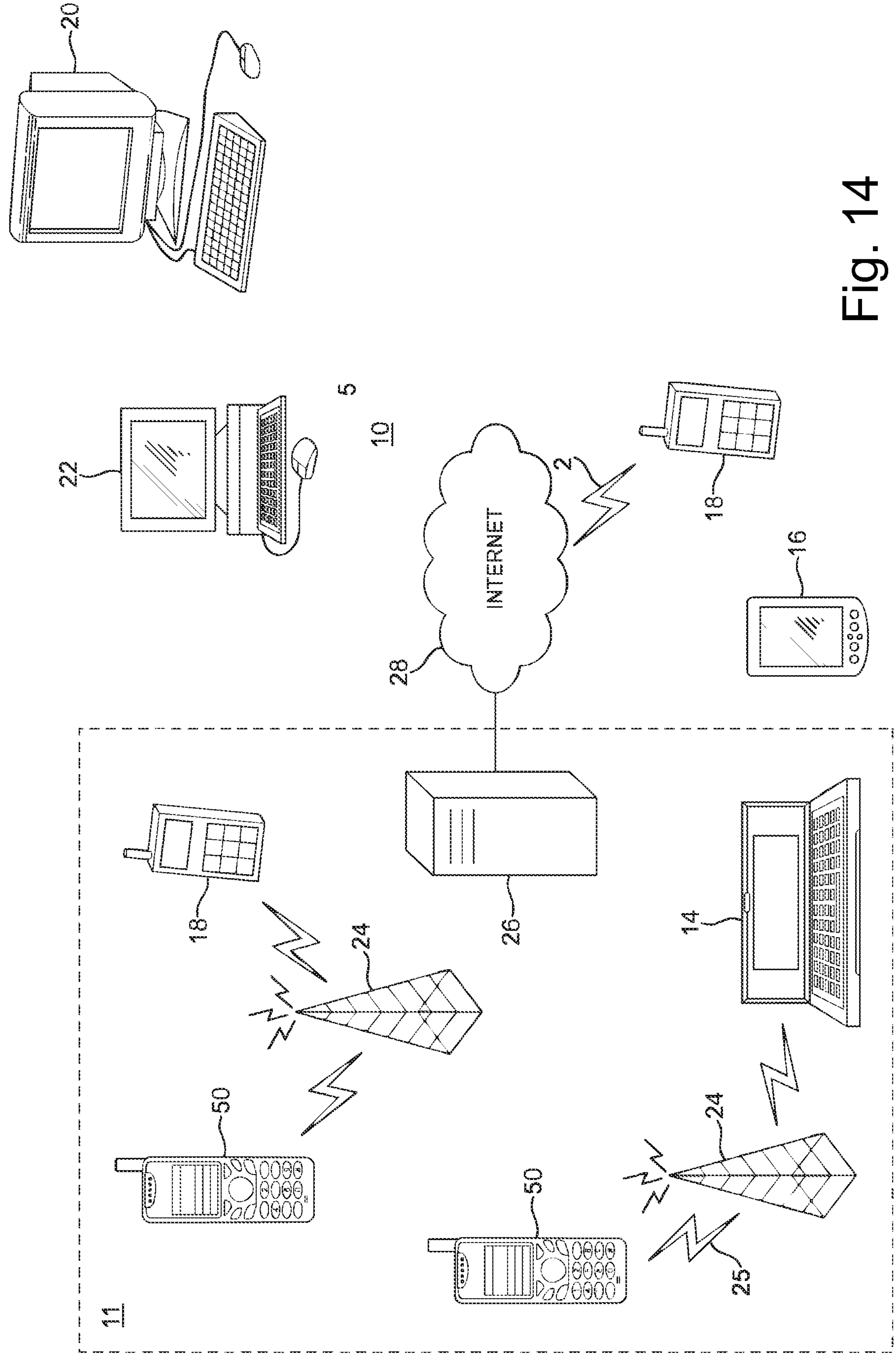


Fig. 14

INTERNATIONAL SEARCH REPORT

International application No.
PCT/FI2018/050587

A. CLASSIFICATION OF SUBJECT MATTER

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

FI, SE, NO, DK

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

EPODOC, EPO-Internal full-text databases, Full-text translation databases from Asian languages, WPIAP, XP3GPP, XPAIP, XPESP, XPETSI, XPI3E, XPIEE, XPIETF, XPIOP, XPIPCOM, XPJPEG, XPMISC, XPOAC, XPRD, XPTK, COMPDX, INSPEC, NPL, PRH-Internal, Internet

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	KAMMACHI-SREEDHAR, K. et al. Viewport-adaptive Encoding and Streaming of 360-degree Video for Virtual Reality Applications. In: IEEE International Symposium on Multimedia. [online], 2016, [retrieved on 2018-11-12]. Retrieved from <https://ieeexplore.ieee.org/stamp/stamp.jsp?tp=&arnumber=7823693><DOI:10.1109/ISM.2016.0126>	
X	sections I, II, II.D, IV; figures 4-6	1, 4, 7-8
Y	as above	2, 5, 9
X	FRAUNHOFER HHI, et al. FS_VR: Viewport-dependent baseline media profile with tile streaming (S4-170589). 3GPP TSG-SA4 Meeting #94 [online], April 2017, [retrieved on 2018-11-12]. Retrieved from <http://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_94/Docs/S4-170589.zip> sections 2.1, 2.2; figures 2, 4	1, 3-4, 6-15

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

* Special categories of cited documents:

"A" document defining the general state of the art which is not considered to be of particular relevance

"E" earlier application or patent but published on or after the international filing date

"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

"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

"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

"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

"&" document member of the same patent family

Date of the actual completion of the international search	Date of mailing of the international search report
14 November 2018 (14.11.2018)	20 November 2018 (20.11.2018)

Name and mailing address of the ISA/FI	Authorized officer
Finnish Patent and Registration Office FI-00091 PRH, FINLAND	Ari Hottinen
Facsimile No. +358 29 509 5328	Telephone No. +358 29 509 5000

INTERNATIONAL SEARCH REPORT

International application No.

PCT/FI2018/050587

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	MARTENS, G. Bandwidth management for ODV tiled streaming with MPEG-DASH, Master Thesis, Hasselt University. [online], September 2015, Master Thesis, Hasselt University, [retrieved on 2018-11-12]. Retrieved from <http://hdl.handle.net/1942/19390> the whole document, in particular chapters 3-4; figs. 4.1-4.4, 4.7-4.8, 4.13, 4.16, 4.33	1, 4, 7-8
Y	US 2017163994 A1 (SANCHEZ DE LA FUENTE YAGO [DE] et al.) 08 June 2017 (08.06.2017) paragraphs [0083]-[0084], [0091]-[0094], [0097]; figs. 6, 8, 10	2, 5, 9
A	EP 3147758 A1 (FACEBOOK INC [US]) 29 March 2017 (29.03.2017) the whole document, in particular paragraphs [0056]-[0057], [0077]-[0078]; Figure 5D	1-15
A	US 9497457 B1 (GUPTA YASHKET [US]) 15 November 2016 (15.11.2016) the whole document, in particular page 4, lines 15-59; Figures 6-9	1-15
P, X	EP 3346709 A1 (NOKIA TECHNOLOGIES OY [FI]) 11 July 2018 (11.07.2018) the whole document, in particular paragraphs [0195]-[0200], [0248]-[0253]; figs. 5-13	1, 4, 7-8, 10, 12, 14-15

INTERNATIONAL SEARCH REPORT
Information on Patent Family Members

International application No.
PCT/FI2018/050587

Patent document cited in search report	Publication date	Patent family members(s)	Publication date
US 2017163994 A1	08/06/2017	CN 106797495 A	31/05/2017
		EP 3183878 A2	28/06/2017
		JP 2017530613 A	12/10/2017
		KR 20170044169 A	24/04/2017
		WO 2016026526 A2	25/02/2016
EP 3147758 A1	29/03/2017	AU 2016326378 A1	12/04/2018
		CA 2999267 A1	30/03/2017
		CN 108293152 A	17/07/2018
		IL 258156 D0	31/05/2018
		KR 20180045049 A	03/05/2018
		MX 2018003593 A	18/06/2018
		US 2017084073 A1	23/03/2017
		US 9858706 B2	02/01/2018
		US 2017084086 A1	23/03/2017
		US 10096130 B2	09/10/2018
		US 2018108171 A1	19/04/2018
		WO 2017053370 A1	30/03/2017
US 9497457 B1	15/11/2016	None	
EP 3346709 A1	11/07/2018	None	

CLASSIFICATION OF SUBJECT MATTER

IPC
H04N 21/4728 (2011.01)
H04N 21/218 (2011.01)
H04N 21/2343 (2011.01)
H04N 19/167 (2014.01)
H04N 19/85 (2014.01)
H04N 19/119 (2014.01)