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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
05 March 2020 (05.03.2020)

WIPO | PCT

(10) International Publication Number
WO 2020/045593 A1

(51) International Patent Classification:
H04N 21/235 (2011.01)

(21) International Application Number:
PCT/JP2019/034011

(22) International Filing Date:
29 August 2019 (29.08.2019)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

62/725,236	30 August 2018 (30.08.2018)	US
62/742,904	08 October 2018 (08.10.2018)	US
62/785,436	27 December 2018 (27.12.2018)	US
62/815,229	07 March 2019 (07.03.2019)	US

(71) Applicant: **SHARP KABUSHIKI KAISHA** [JP/JP]; 1, Takumi-cho, Sakai-ku, Sakai City, Osaka, 5908522 (JP).

(72) Inventor: **DESHPANDE, Sachin G.**.

(74) Agent: **HARAKENZO WORLD PATENT & TRADEMARK**; Daiwa Minamimorimachi Building, 2-6, Tenjinbashi 2-chome Kita, Kita-ku, Osaka-shi, Osaka, 5300041 (JP).

(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,

(54) Title: SYSTEMS AND METHODS FOR SIGNALING SUB-PICTURE TIMED METADATA INFORMATION

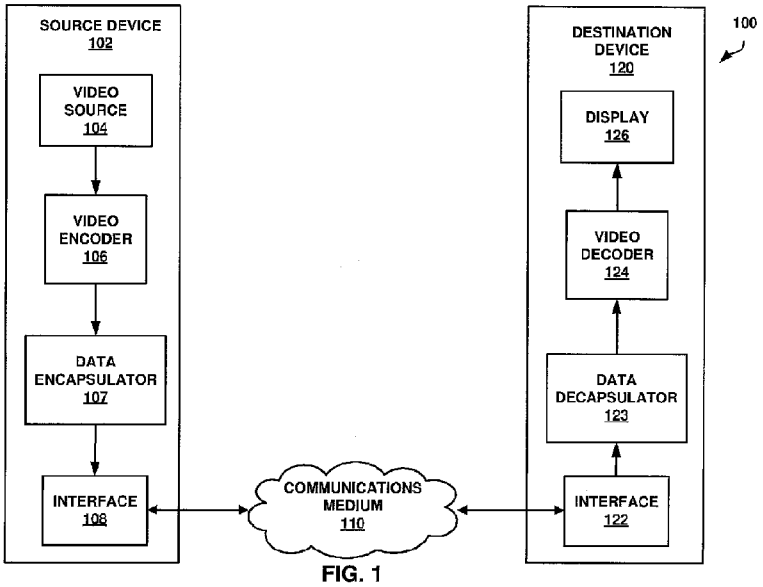


FIG. 1

(57) Abstract: A method of signaling information associated with an omnidirectional video is disclosed, the method comprising: encapsulating a timed metadata track associated with a particular representation; and signaling an association descriptor of the particular representation of the timed metadata track, wherein the association descriptor includes (i) a string in an association element of a type concerning a sub-picture composition identifier value (For example, ["SubPicCompositionId="aa"] in paragraph [0046].) and (ii) a constant of the association element. (For example, 'cdtg' as the value of the association@associationKindList attribute of the association element in paragraph [0046].)

WO 2020/045593 A1

[Continued on next page]

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).

Declarations under Rule 4.17:

— *as to applicant's entitlement to apply for and be granted a patent (Rule 4.17(ii))*

— *as to the applicant's entitlement to claim the priority of the earlier application (Rule 4.17(iii))*

Published:

— *with international search report (Art. 21(3))*

Description

Title of Invention: SYSTEMS AND METHODS FOR SIGNALING SUB-PICTURE TIMED METADATA INFORMATION

Technical Field

[0001] This disclosure relates to the field of interactive video distribution and more particularly to techniques for signaling of sub-picture timed metadata information in a virtual reality application.

Background Art

[0002] Digital media playback capabilities may be incorporated into a wide range of devices, including digital televisions, including so-called "smart" televisions, set-top boxes, laptop or desktop computers, tablet computers, digital recording devices, digital media players, video gaming devices, cellular phones, including so-called "smart" phones, dedicated video streaming devices, and the like. Digital media content (e.g., video and audio programming) may originate from a plurality of sources including, for example, over-the-air television providers, satellite television providers, cable television providers, online media service providers, including, so-called streaming service providers, and the like. Digital media content may be delivered over packet-switched networks, including bidirectional networks, such as Internet Protocol (IP) networks and unidirectional networks, such as digital broadcast networks.

[0003] Digital video included in digital media content may be coded according to a video coding standard. Video coding standards may incorporate video compression techniques. Examples of video coding standards include ISO/IEC MPEG-4 Visual and ITU-T H.264 (also known as ISO/IEC MPEG-4 AVC) and High-Efficiency Video Coding (HEVC). Video compression techniques enable data requirements for storing and transmitting video data to be reduced. Video compression techniques may reduce data requirements by exploiting the inherent redundancies in a video sequence. Video compression techniques may sub-divide a video sequence into successively smaller portions (i.e., groups of frames within a video sequence, a frame within a group of frames, slices within a frame, coding tree units (e.g., macroblocks) within a slice, coding blocks within a coding tree unit, etc.). Prediction coding techniques may be used to generate difference values between a unit of video data to be coded and a reference unit of video data. The difference values may be referred to as residual data. Residual data may be coded as quantized transform coefficients. Syntax elements may relate residual data and a reference coding unit. Residual data and syntax elements may be included in a compliant bitstream. Compliant bitstreams and associated metadata may be formatted according to data structures. Compliant bitstreams and associated

metadata may be transmitted from a source to a receiver device (e.g., a digital television or a smart phone) according to a transmission standard. Examples of transmission standards include Digital Video Broadcasting (DVB) standards, Integrated Services Digital Broadcasting Standards (ISDB) standards, and standards developed by the Advanced Television Systems Committee (ATSC), including, for example, the ATSC 2.0 standard. The ATSC is currently developing the so-called ATSC 3.0 suite of standards.

Summary of Invention

[0004] In one example, a method of signaling information associated with an omnidirectional video, the method comprising: encapsulating a timed metadata track associated with a particular representation; and signaling an association descriptor of the particular representation of the timed metadata track, wherein the association descriptor includes (i) a string in an association element of a type concerning a subpicture composition identifier value and (ii) a constant of the association element.

[0005] In one example, a method of determining information associated with an omnidirectional video, the method comprising: decapsulating a timed metadata track associated with a particular representation; and receiving an association descriptor of the particular representation of the timed metadata track, wherein the association descriptor includes (i) a string in an association element of a type concerning a subpicture composition identifier value and (ii) a constant of the association element.

Brief Description of Drawings

[0006] [fig.1]FIG. 1 is a block diagram illustrating an example of a system that may be configured to transmit coded video data according to one or more techniques of this this disclosure.

[fig.2A]FIGS. 2A is a conceptual diagram illustrating coded video data and corresponding data structures according to one or more techniques of this this disclosure.

[fig.2B]FIGS. 2B is a conceptual diagram illustrating coded video data and corresponding data structures according to one or more techniques of this this disclosure.

[fig.3]FIG. 3 is a conceptual diagram illustrating coded video data and corresponding data structures according to one or more techniques of this this disclosure.

[fig.4]FIG. 4 is a conceptual diagram illustrating an example of a coordinate system according to one or more techniques of this disclosure.

[fig.5A]FIG. 5A is a conceptual diagram illustrating examples of specifying regions on a sphere according to one or more techniques of this this disclosure.

[fig.5B]FIG. 5B is a conceptual diagram illustrating examples of specifying regions on a sphere according to one or more techniques of this this disclosure.

[fig.6]FIG. 6 is a conceptual diagram illustrating examples of a projected picture

region and a packed picture region according to one or more techniques of this disclosure.

[fig.7]FIG. 7 is a conceptual drawing illustrating an example of components that may be included in an implementation of a system that may be configured to transmit coded video data according to one or more techniques of this this disclosure.

[fig.8]FIG. 8 is a block diagram illustrating an example of a data encapsulator that may implement one or more techniques of this disclosure.

[fig.9]FIG. 9 is a block diagram illustrating an example of a receiver device that may implement one or more techniques of this disclosure.

Description of Embodiments

[0007] In general, this disclosure describes various techniques for signaling information associated with a virtual reality application. In particular, this disclosure describes techniques for signaling sub-picture timed metadata information. It should be noted that although in some examples, the techniques of this disclosure are described with respect to transmission standards, the techniques described herein may be generally applicable. For example, the techniques described herein are generally applicable to any of DVB standards, ISDB standards, ATSC Standards, Digital Terrestrial Multimedia Broadcast (DTMB) standards, Digital Multimedia Broadcast (DMB) standards, Hybrid Broadcast and Broadband Television (HbbTV) standards, World Wide Web Consortium (W3C) standards, and Universal Plug and Play (UPnP) standard. Further, it should be noted that although techniques of this disclosure are described with respect to ITU-T H.264 and ITU-T H.265, the techniques of this disclosure are generally applicable to video coding, including omnidirectional video coding. For example, the coding techniques described herein may be incorporated into video coding systems, (including video coding systems based on future video coding standards) including block structures, intra prediction techniques, inter prediction techniques, transform techniques, filtering techniques, and/or entropy coding techniques other than those included in ITU-T H.265. Thus, reference to ITU-T H.264 and ITU-T H.265 is for descriptive purposes and should not be construed to limit the scope of the techniques described herein. Further, it should be noted that incorporation by reference of documents herein should not be construed to limit or create ambiguity with respect to terms used herein. For example, in the case where an incorporated reference provides a different definition of a term than another incorporated reference and/or as the term is used herein, the term should be interpreted in a manner that broadly includes each respective definition and/or in a manner that includes each of the particular definitions in the alternative.

[0008] In one example, a method of signaling information associated with an omnidi-

rectional video comprises encapsulating a timed metadata track in a particular representation associated with a sub-picture composition and signaling an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0009] In one example, a device comprises one or more processors configured to encapsulate a timed metadata track in a particular representation associated with a sub-picture composition and signal an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0010] In one example, a non-transitory computer-readable storage medium comprises instructions stored thereon that, when executed, cause one or more processors of a device to encapsulate a timed metadata track in a particular representation associated with a sub-picture composition and signal an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0011] In one example, an apparatus comprises means for encapsulating a timed metadata track in a particular representation associated with a sub-picture composition and means for signaling an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0012] In one example, a method of determining information associated with an omnidirectional video comprises decapsulating a timed metadata track in a particular representation associated with a sub-picture composition and parsing an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0013] In one example, a device comprises one or more processors configured to decapsulate a timed metadata track in a particular representation associated with a sub-picture composition and parse an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0014] In one example, a non-transitory computer-readable storage medium comprises instructions stored thereon that, when executed, cause one or more processors of a device to decapsulate a timed metadata track in a particular representation associated with a sub-picture composition and parse an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0015] In one example, an apparatus comprises means for decapsulating a timed metadata track in a particular representation associated with a sub-picture composition and

parsing an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0016] The details of one or more examples are set forth in the accompanying drawings and the description below. Other features, objects, and advantages will be apparent from the description and drawings, and from the claims.

[0017] Video content typically includes video sequences comprised of a series of frames. A series of frames may also be referred to as a group of pictures (GOP). Each video frame or picture may include a one or more slices, where a slice includes a plurality of video blocks. A video block may be defined as the largest array of pixel values (also referred to as samples) that may be predictively coded. Video blocks may be ordered according to a scan pattern (e.g., a raster scan). A video encoder performs predictive encoding on video blocks and sub-divisions thereof. ITU-T H.264 specifies a macroblock including 16 x 16 luma samples. ITU-T H.265 specifies an analogous Coding Tree Unit (CTU) structure where a picture may be split into CTUs of equal size and each CTU may include Coding Tree Blocks (CTB) having 16 x 16, 32 x 32, or 64 x 64 luma samples. As used herein, the term video block may generally refer to an area of a picture or may more specifically refer to the largest array of pixel values that may be predictively coded, sub-divisions thereof, and/or corresponding structures. Further, according to ITU-T H.265, each video frame or picture may be partitioned to include one or more tiles, where a tile is a sequence of coding tree units corresponding to a rectangular area of a picture.

[0018] In ITU-T H.265, the CTBs of a CTU may be partitioned into Coding Blocks (CB) according to a corresponding quadtree block structure. According to ITU-T H.265, one luma CB together with two corresponding chroma CBs and associated syntax elements are referred to as a coding unit (CU). A CU is associated with a prediction unit (PU) structure defining one or more prediction units (PU) for the CU, where a PU is associated with corresponding reference samples. That is, in ITU-T H.265 the decision to code a picture area using intra prediction or inter prediction is made at the CU level and for a CU one or more predictions corresponding to intra prediction or inter prediction may be used to generate reference samples for CBs of the CU. In ITU-T H.265, a PU may include luma and chroma prediction blocks (PBs), where square PBs are supported for intra prediction and rectangular PBs are supported for inter prediction. Intra prediction data (e.g., intra prediction mode syntax elements) or inter prediction data (e.g., motion data syntax elements) may associate PUs with corresponding reference samples. Residual data may include respective arrays of difference values corresponding to each component of video data (e.g., luma (Y) and chroma (Cb and Cr)). Residual data may be in the pixel domain. A transform, such as, a discrete

cosine transform (DCT), a discrete sine transform (DST), an integer transform, a wavelet transform, or a conceptually similar transform, may be applied to pixel difference values to generate transform coefficients. It should be noted that in ITU-T H.265, CUs may be further sub-divided into Transform Units (TUs). That is, an array of pixel difference values may be sub-divided for purposes of generating transform coefficients (e.g., four 8 x 8 transforms may be applied to a 16 x 16 array of residual values corresponding to a 16 x16 luma CB), such sub-divisions may be referred to as Transform Blocks (TBs). Transform coefficients may be quantized according to a quantization parameter (QP). Quantized transform coefficients (which may be referred to as level values) may be entropy coded according to an entropy encoding technique (e.g., content adaptive variable length coding (CAVLC), context adaptive binary arithmetic coding (CABAC), probability interval partitioning entropy coding (PIPE), etc.). Further, syntax elements, such as, a syntax element indicating a prediction mode, may also be entropy coded. Entropy encoded quantized transform coefficients and corresponding entropy encoded syntax elements may form a compliant bitstream that can be used to reproduce video data. A binarization process may be performed on syntax elements as part of an entropy coding process. Binarization refers to the process of converting a syntax value into a series of one or more bits. These bits may be referred to as "bins."

[0019] Virtual Reality (VR) applications may include video content that may be rendered with a head-mounted display, where only the area of the spherical video that corresponds to the orientation of the user's head is rendered. VR applications may be enabled by omnidirectional video, which is also referred to as 360 degree spherical video of 360 degree video. Omnidirectional video is typically captured by multiple cameras that cover up to 360 degrees of a scene. A distinct feature of omnidirectional video compared to normal video is that, typically only a subset of the entire captured video region is displayed, i.e., the area corresponding to the current user's field of view (FOV) is displayed. A FOV is sometimes also referred to as viewport. In other cases, a viewport may be described as part of the spherical video that is currently displayed and viewed by the user. It should be noted that the size of the viewport can be smaller than or equal to the field of view. Further, it should be noted that omnidirectional video may be captured using monoscopic or stereoscopic cameras. Monoscopic cameras may include cameras that capture a single view of an object. Stereoscopic cameras may include cameras that capture multiple views of the same object (e.g., views are captured using two lenses at slightly different angles). It should be noted that in some cases, the center point of a viewport may be referred to as a viewpoint. However, as used herein, the term viewpoint when associated with a camera (e.g., camera viewpoint), may refer to information associated with a camera used to capture a

view(s) of an object (e.g., camera parameters). Further, it should be noted that in some cases, images for use in omnidirectional video applications may be captured using ultra wide-angle lens (i.e., so-called fisheye lens). In any case, the process for creating 360 degree spherical video may be generally described as stitching together input images and projecting the stitched together input images onto a three-dimensional structure (e.g., a sphere or cube), which may result in so-called projected frames. Further, in some cases, regions of projected frames may be transformed, resized, and relocated, which may result in a so-called packed frame.

[0020] Transmission systems may be configured to transmit omnidirectional video to one or more computing devices. Computing devices and/or transmission systems may be based on models including one or more abstraction layers, where data at each abstraction layer is represented according to particular structures, e.g., packet structures, modulation schemes, etc. An example of a model including defined abstraction layers is the so-called Open Systems Interconnection (OSI) model. The OSI model defines a 7-layer stack model, including an application layer, a presentation layer, a session layer, a transport layer, a network layer, a data link layer, and a physical layer. It should be noted that the use of the terms upper and lower with respect to describing the layers in a stack model may be based on the application layer being the uppermost layer and the physical layer being the lowermost layer. Further, in some cases, the term "Layer 1" or "L1" may be used to refer to a physical layer, the term "Layer 2" or "L2" may be used to refer to a link layer, and the term "Layer 3" or "L3" or "IP layer" may be used to refer to the network layer.

[0021] A physical layer may generally refer to a layer at which electrical signals form digital data. For example, a physical layer may refer to a layer that defines how modulated radio frequency (RF) symbols form a frame of digital data. A data link layer, which may also be referred to as a link layer, may refer to an abstraction used prior to physical layer processing at a sending side and after physical layer reception at a receiving side. As used herein, a link layer may refer to an abstraction used to transport data from a network layer to a physical layer at a sending side and used to transport data from a physical layer to a network layer at a receiving side. It should be noted that a sending side and a receiving side are logical roles and a single device may operate as both a sending side in one instance and as a receiving side in another instance. A link layer may abstract various types of data (e.g., video, audio, or application files) encapsulated in particular packet types (e.g., Motion Picture Expert Group - Transport Stream (MPEG-TS) packets, Internet Protocol Version 4 (IPv4) packets, etc.) into a single generic format for processing by a physical layer. A network layer may generally refer to a layer at which logical addressing occurs. That is, a network layer may generally provide addressing information (e.g., Internet Protocol (IP) addresses)

such that data packets can be delivered to a particular node (e.g., a computing device) within a network. As used herein, the term network layer may refer to a layer above a link layer and/or a layer having data in a structure such that it may be received for link layer processing. Each of a transport layer, a session layer, a presentation layer, and an application layer may define how data is delivered for use by a user application.

[0022] Wang et al., ISO/IEC JTC1/SC29/WG11 W17827 "WD 2 of ISO/IEC 23090-2 OMAF 2nd edition," August 2018, Ljubljana, Slovenia, which is incorporated by reference and herein referred to as Wang, defines a media application format that enables omnidirectional media applications. Wang specifies a coordinate system for omnidirectional video; projection and rectangular region-wise packing methods that may be used for conversion of a spherical video sequence or image into a two-dimensional rectangular video sequence or image, respectively; storage of omnidirectional media and the associated metadata using the ISO Base Media File Format (ISOBMFF); encapsulation, signaling, and streaming of omnidirectional media in a media streaming system; and media profiles and presentation profiles. It should be noted that for the sake of brevity, a complete description of Wang is not provided herein. However, reference is made to relevant sections of Wang.

[0023] Wang provides media profiles where video is coded according to ITU-T H.265. ITU-T H.265 is described in High Efficiency Video Coding (HEVC), Rec. ITU-T H.265 December 2016, which is incorporated by reference, and referred to herein as ITU-T H.265. As described above, according to ITU-T H.265, each video frame or picture may be partitioned to include one or more slices and further partitioned to include one or more tiles. FIGS. 2A-2B are conceptual diagrams illustrating an example of a group of pictures including slices and further partitioning pictures into tiles. In the example illustrated in FIG. 2A, Pic_4 is illustrated as including two slices (i.e., $Slice_1$ and $Slice_2$) where each slice includes a sequence of CTUs (e.g., in raster scan order). In the example illustrated in FIG. 2B, Pic_4 is illustrated as including six tiles (i.e., $Tile_1$ to $Tile_6$), where each tile is rectangular and includes a sequence of CTUs. It should be noted that in ITU-T H.265, a tile may consist of coding tree units contained in more than one slice and a slice may consist of coding tree units contained in more than one tile. However, ITU-T H.265 provides that one or both of the following conditions shall be fulfilled: (1) All coding tree units in a slice belong to the same tile; and (2) All coding tree units in a tile belong to the same slice.

[0024] 360 degree spherical video may include regions. Referring to the example illustrated in FIG. 3, the 360 degree spherical video includes Regions A, B, and C and as illustrated in FIG. 3, tiles (i.e., $Tile_1$ to $Tile_6$) may form a region of an omnidirectional video. In the example illustrated in FIG. 3, each of the regions are illustrated as including CTUs. As described above, CTUs may form slices of coded video data and/

or tiles of video data. Further, as described above, video coding techniques may code areas of a picture according to video blocks, sub-divisions thereof, and/or corresponding structures and it should be noted that video coding techniques enable video coding parameters to be adjusted at various levels of a video coding structure, e.g., adjusted for slices, tiles, video blocks, and/or at sub-divisions. In one example, the 360 degree video illustrated in FIG. 3 may represent a sporting event where Region A and Region C include views of the stands of a stadium and Regions B includes a view of the playing field (e.g., the video is captured by a 360 degree camera placed at the 50-yard line).

[0025] As described above, a viewport may be part of the spherical video that is currently displayed and viewed by the user. As such, regions of omnidirectional video may be selectively delivered depending on the user's viewport, i.e., viewport-dependent delivery may be enabled in omnidirectional video streaming. Typically, to enable viewport-dependent delivery, source content is split into sub-picture sequences before encoding, where each sub-picture sequence covers a subset of the spatial area of the omnidirectional video content, and sub-picture sequences are then encoded independently from each other as a single-layer bitstream. For example, referring to FIG. 3, each of Region A, Region B, and Region C, or portions thereof, may correspond to independently coded sub-picture bitstreams. Each sub-picture bitstream may be encapsulated in a file as its own track and tracks may be selectively delivered to a receiver device based on viewport information. It should be noted that in some cases, it is possible that sub-pictures overlap. For example, referring to FIG. 3, $Tile_1$, $Tile_2$, $Tile_4$, and $Tile_5$ may form a sub-picture and $Tile_2$, $Tile_3$, $Tile_5$, and $Tile_6$ may form a sub-picture. Thus, a particular sample may be included in multiple sub-pictures. Wang provides where a composition-aligned sample includes one of a sample in a track that is associated with another track, the sample has the same composition time as a particular sample in the another track, or, when a sample with the same composition time is not available in the another track, the closest preceding composition time relative to that of a particular sample in the another track. Further, Wang provides where a constituent picture includes part of a spatially frame-packed stereoscopic picture that corresponds to one view, or a picture itself when frame packing is not in use or the temporal interleaving frame packing arrangement is in use.

[0026] As described above, Wang specifies a coordinate system for omnidirectional video. In Wang, the coordinate system consists of a unit sphere and three coordinate axes, namely the X (back-to-front) axis, the Y (lateral, side-to-side) axis, and the Z (vertical, up) axis, where the three axes cross at the center of the sphere. The location of a point on the sphere is identified by a pair of sphere coordinates azimuth (φ) and elevation (θ). FIG. 4 illustrates the relation of the sphere coordinates azimuth (φ) and elevation

(θ) to the X, Y, and Z coordinate axes as specified in Wang. It should be noted that in Wang the value ranges of azimuth is -180.0, inclusive, to 180.0, exclusive, degrees and the value range of elevation is -90.0 to 90.0, inclusive, degrees. Wang specifies where a region on a sphere may be specified by four great circles, where a great circle (also referred to as a Riemannian circle) is an intersection of the sphere and a plane that passes through the center point of the sphere, where the center of the sphere and the center of a great circle are co-located. Wang further describes where a region on a sphere may be specified by two azimuth circles and two elevation circles, where a azimuth circle is a circle on the sphere connecting all points with the same azimuth value, and an elevation circle is a circle on the sphere connecting all points with the same elevation value. The sphere region structure in Wang forms the basis for signaling various types of metadata.

[0027] It should be noted that with respect to the equations used herein, the following arithmetic operators may be used:

+Addition

− Subtraction (as a two-argument operator) or negation (as a unary prefix operator)

* Multiplication, including matrix multiplication

x^y Exponentiation. Specifies x to the power of y. In other contexts, such notation is used for superscripting not intended for interpretation as exponentiation.

/ Integer division with truncation of the result toward zero. For example, 7 / 4 and −7 / −4 are truncated to 1 and −7 / 4 and 7 / −4 are truncated to −1.

÷ Used to denote division in mathematical equations where no truncation or rounding is intended.

$\frac{x}{y}$ Used to denote division in mathematical equations where no truncation or rounding is intended.

x % y Modulus. Remainder of x divided by y, defined only for integers x and y with x >= 0 and y > 0.

It should be noted that with respect to the equations used herein, the following logical operators may be used:

x && y Boolean logical "and" of x and y

x | | y Boolean logical "or" of x and y

! Boolean logical "not"

x ? y : z If x is TRUE or not equal to 0, evaluates to the value of y; otherwise, evaluates to the value of z.

It should be noted that with respect to the equations used herein, the following relational operators may be used:

>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
==	Equal to
!=	Not equal to

It should be noted in the syntax used herein, unsigned int(n) refers to an unsigned integer having n-bits. Further, bit(n) refers to a bit value having n-bits.

[0028]

As described above, Wang specifies how to store omnidirectional media and the associated metadata using the International Organization for Standardization (ISO) base media file format (ISOBMFF). Wang specifies where a file format that supports metadata specifying the area of the spherical surface covered by the projected frame. In particular, Wang includes a sphere region structure specifying a sphere region having the following definition, syntax and semantic:

Definition

The sphere region structure (SphereRegionStruct) specifies a sphere region.

When centre_tilt is equal to 0, the sphere region specified by this structure is derived as follows:

–If both azimuth_range and elevation_range are equal to 0, the sphere region specified by this structure is a point on a spherical surface.

–Otherwise, the sphere region is defined using variables centreAzimuth, centreElevation, cAzimuth1, cAzimuth, cElevation1, and cElevation2 derived as follows:

centreAzimuth = centre_azimuth ÷ 65536

centreElevation = centre_elevation ÷ 65536

cAzimuth1 = (centre_azimuth – azimuth_range ÷ 2) ÷ 65536

cAzimuth2 = (centre_azimuth + azimuth_range ÷ 2) ÷ 65536

cElevation1 = (centre_elevation – elevation_range ÷ 2) ÷ 65536

cElevation2 = (centre_elevation + elevation_range ÷ 2) ÷ 65536

The sphere region is defined as follows with reference to the shape type value specified in the semantics of the structure containing this instance of SphereRegionStruct:

–When the shape type value is equal to 0, the sphere region is specified by four great circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in FIG. 5A.

–When the shape type value is equal to 1, the sphere region is specified by two azimuth circles and two elevation circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in FIG. 5B.

When centre_tilt is not equal to 0, the sphere region is firstly derived as above and then a tilt rotation is applied along the axis originating from the sphere origin passing through the centre point of the sphere region, where the angle value increases clockwise when looking from the origin towards the positive end of the axis. The final sphere region is the one after applying the tilt rotation.

Shape type value equal to 0 specifies that the sphere region is specified by four great circles as illustrated in FIG. 5A.

Shape type value equal to 1 specifies that the sphere region is specified by two azimuth circles and two elevation circles as illustrated in FIG. 5B.

Shape type values greater than 1 are reserved.

Syntax

```
aligned(8) SphereRegionStruct(range_included_flag) {
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        singed int(32) centre_tilt;
        if (range_included_flag) {
                unsigned int(32) azimuth_range;
                unsigned int(32) elevation_range;
        }
        unsigned int(1) interpolate;
        bit(7) reserved = 0;
}
```

Semantics

centre_azimuth and centre_elevation specify the centre of the sphere region. centre_azimuth shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. centre_elevation shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive.

centre_tilt specifies the tilt angle of the sphere region. centre_tilt shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive.

azimuth_range and elevation_range, when present, specify the azimuth and elevation ranges, respectively, of the sphere region specified by this structure in units of 2^{-16} degrees. azimuth_range and elevation_range specify the range through the centre point of the sphere region, as illustrated by FIG. 5A or FIG. 5B. When azimuth_range and elevation_range are not present in this instance of SphereRegionStruct, they are inferred as specified in the semantics of the structure containing this instance of SphereRegionStruct. azimuth_range shall be in the range of 0 to $360 * 2^{16}$, inclusive. elevation_range shall be in the range of 0 to $180 * 2^{16}$, inclusive.

The semantics of interpolate are specified by the semantics of the structure containing this instance of SphereRegionStruct.

It should be noted that Wang et al., ISO/IEC JTC1/SC29/WG11 W18227 "WD 4 of ISO/IEC 23090-2 OMAF 2nd edition," January 2019, Marrakech, Morroco, is an update to Wang, is incorporated by reference and referred to herein as Wang2. Wang2 includes the same definition, syntax and semantics for a sphere region structure specifying a sphere region as Wang.

[0029]

As described above, the sphere region structure in Wang forms the basis for signaling various types of metadata. With respect to specifying a generic timed metadata track syntax for sphere regions, Wang specifies a sample entry and a sample format. The sample entry structure is specified as having the following definition, syntax and semantics:

Definition

Exactly one SphereRegionConfigBox shall be present in the sample entry. SphereRegionConfigBox specifies the shape of the sphere region specified by the samples. When the azimuth and elevation ranges of the sphere region in the samples do not change, they may be indicated in the sample entry.

Syntax

```
class SphereRegionSampleEntry(type) extends MetaDataSampleEntry(type) {
        SphereRegionConfigBox(); // mandatory
        Box[] other_boxes; // optional
}
class SphereRegionConfigBox extends FullBox('rosc', 0, 0) {
        unsigned int(8) shape_type;
        bit(7) reserved = 0;
        unsigned int(1) dynamic_range_flag;
        if (dynamic_range_flag == 0) {
                unsigned int(32) static_azimuth_range;
                unsigned int(32) static_elevation_range;
        }
        unsigned int(8) num_regions;
}
```

Semantics

shape_type equal to 0 specifies that the sphere region is specified by four great circles. shape_type equal to 1 specifies that the sphere region is specified by two azimuth circles and two elevation circles. shape_type values greater than 1 are reserved. The value of shape_type is used as the shape type value when applying the clause describing the Sphere region (provided above) to the semantics of the samples of the sphere region metadata track.

dynamic_range_flag equal to 0 specifies that the azimuth and elevation ranges of the sphere region remain unchanged in all samples referring to this sample entry. dynamic_range_flag equal to 1 specifies that the azimuth and elevation ranges of the sphere region are indicated in the sample format.

static_azimuth_range and static_elevation_range specify the azimuth and elevation

ranges, respectively, of the sphere region for each sample referring to this sample entry in units of 2^{-16} degrees. static_azimuth_range and static_elevation_range specify the ranges through the centre point of the sphere region, as illustrated by FIG. 5A or FIG. 5B. static_azimuth_range shall be in the range of 0 to 360 * 2^{16}, inclusive. static_elevation_range shall be in the range of 0 to 180 * 2^{16}, inclusive. When static_azimuth_range and static_elevation_range are present and are both equal to 0, the sphere region for each sample referring to this sample entry is a point on a spherical surface. When static_azimuth_range and static_elevation_range are present, the values of azimuth_range and elevation_range are inferred to be equal to static_azimuth_range and static_elevation_range, respectively, when applying the clause describing the Sphere region (provided above) to the semantics of the samples of the sphere region metadata track.

num_regions specifies the number of sphere regions in the samples referring to this sample entry. num_regions shall be equal to 1. Other values of num_regions are reserved.

The sample format structure is specified as having the following definition, syntax and semantics:

Definition

Each sample specifies a sphere region. The SphereRegionSample structure may be extended in derived track formats.

Syntax

```
aligned(8) SphereRegionSample() {
  for (i = 0; i < num_regions; i++)
        SphereRegionStruct(dynamic_range_flag)
}
```

Semantics

The sphere region structure clause, provided above, applies to the sample that contains the SphereRegionStruct structure.

Let the target media samples be the media samples in the referenced media tracks with composition times greater than or equal to the composition time of this sample and less than the composition time of the next sample.

interpolate equal to 0 specifies that the values of centre_azimuth, centre_elevation, centre_tilt, azimuth_range (if present), and elevation_range (if present) in this sample apply to the target media samples. interpolate equal to 1 specifies that the values of centre_azimuth, centre_elevation, centre_tilt, azimuth_range (if present), and elevation_range (if present) that apply to the target media samples are linearly interpolated from the values of the corresponding fields in this sample and the previous sample.

The value of interpolate for a sync sample, the first sample of the track, and the first sample of a track fragment shall be equal to 0.

In Wang timed metadata may be signaled based on a sample entry and a sample format. For example, Wang includes an initial viewing orientation metadata having the following definition, syntax and semantics:

Definition

This metadata indicates initial viewing orientations that should be used when playing the associated media tracks or a single omnidirectional image stored as an image item. In the absence of this type of metadata centre_azimuth, centre_elevation, and centre_tilt should all be inferred to be equal to 0.

An OMAF (omnidirectional media format) player should use the indicated or inferred centre_azimuth, centre_elevation, and centre_tilt values as follows:

–If the orientation/viewport metadata of the OMAF player is obtained on the basis of an orientation sensor included in or attached to a viewing device, the OMAF player should

- obey only the centre_azimuth value, and
- ignore the values of centre_elevation and centre_tilt and use the respective values from the orientation sensor instead.

–Otherwise, the OMAF player should obey all three of centre_azimuth, centre_elevation, and centre_tilt.

The track sample entry type 'initial view orientation timed metadata' shall be used.

shape_type shall be equal to 0, dynamic_range_flag shall be equal to 0, static_azimuth_range shall be equal to 0, and static_elevation_range shall be equal to 0 in the SphereRegionConfigBox of the sample entry.

NOTE: This metadata applies to any viewport regardless of which azimuth and elevation ranges are covered by the viewport. Thus, dynamic_range_flag, static_azimuth_range, and static_elevation_range do not affect the dimensions of the viewport that this metadata concerns and are hence required to be equal to 0. When the OMAF player obeys the centre_tilt value as concluded above, the value of centre_tilt could be interpreted by setting the azimuth and elevation ranges for the sphere region of the viewport equal to those that are actually used in displaying the viewport.

Syntax

```
class InitialViewingOrientationSample() extends SphereRegionSample() {
        unsigned int(1) refresh_flag;
        bit(7) reserved = 0;
}
```

Semantics

NOTE 1: As the sample structure extends from SphereRegionSample, the syntax elements of SphereRegionSample are included in the sample.

centre_azimuth, centre_elevation, and centre_tilt specify the viewing orientation in units of 2^{-16} degrees relative to the global coordinate axes. centre_azimuth and centre_elevation indicate the centre of the viewport, and centre_tilt indicates the tilt angle of the viewport.

interpolate shall be equal to 0.

refresh_flag equal to 0 specifies that the indicated viewing orientation should be used when starting the playback from a time-parallel sample in an associated media track. refresh_flag equal to 1 specifies that the indicated viewing orientation should always be used when rendering the time-parallel sample of each associated media track, i.e., both in continuous playback and when starting the playback from the time-parallel sample.

NOTE 2: refresh_flag equal to 1 enables the content author to indicate that a particular viewing orientation is recommended even when playing the video continuously. For example, refresh_flag equal to 1 could be indicated for a scene cut position.

Further, Wang specifies a recommended viewport timed metadata track as follows:

The recommended viewport timed metadata track indicates the viewport that should be displayed when the user does not have control of the viewing orientation or has released control of the viewing orientation.

NOTE: The recommended viewport timed metadata track may be used for indicating a recommended viewport based on a director's cut or based on measurements of viewing statistics.

The track sample entry type 'rcvp' shall be used.
The sample entry of this sample entry type is specified as follows:

```
class RcvpSampleEntry() extends SphereRegionSampleEntry('rcvp') {
        RcvpInfoBox(); // mandatory
}
class RcvpInfoBox extends FullBox('rvif', 0, 0) {
        unsigned int(8) viewport_type;
        string viewport_description;
}
```

viewport_type specifies the type of the recommended viewport as listed in Table 1.

Value	Description
0	A recommended viewport per the director's cut, i.e., a viewport suggested according to the creative intent of the content author or content provider
1	A recommended viewport selected based on measurements of viewing statistics
2..239	Reserved
240..255	Unspecified (for use by applications or external specifications)

Table 1

viewport_description is null-terminated UTF-8 string that provides a textual description of the recommended viewport.

The sample syntax of SphereRegionSample shall be used.

shape_type shall be equal to 0 in the SphereRegionConfigBox of the sample entry.

static_azimuth_range and static_elevation_range, when present, or azimuth_range and elevation_range, when present, indicate the azimuth and elevation ranges, respectively, of the recommended viewport.

centre_azimuth and centre_elevation indicate the centre point of the recommended viewport relative to the global coordinate axes. centre_tilt indicates the tilt angle of the recommended viewport.

Wang further includes an overlay structure for enabling turning on and off overlays (e.g., logos). An overlay may be defined as rendering of visual media over 360-degree video content. The visual media may include one or more of videos, images and text. In particular, Wang provides the following definition, syntax, and semantics for an overlay structure:

Definition

OverlayStruct specifies the overlay related metadata per each overlay.

Syntax

```
aligned(8) class SingleOverlayStruct() {
        unsigned int(16) overlay_id;
        for (i = 0; i < num_flag_bytes * 8; i++)
                unsigned int(1) overlay_control_flag[i];

        for (i = 0; i < num_flag_bytes * 8; i++){
                if (overlay_control_flag[i]) {
                        unsigned int(1) overlay_control_essential_flag[i];
                        unsigned int(15) byte_count[i];
                        unsigned int(8) overlay_control_struct[i][byte_count[i]];
                }
        }
}
aligned(8) class OverlayStruct() {
        unsigned int(16) num_overlays;
        unsigned int(8) num_flag_bytes;
        for (i = 0; i < num_overlays; i++)
                SingleOverlayStruct();
}
```

Semantics

num_overlays specifies the number of overlays described by this structure. num_overlays equal to 0 is reserved.

num_flag_bytes specifies the number of bytes allocated collectively by the overlay_control_flag[i] syntax elements. num_flag_bytes equal to 0 is reserved.

overlay_id provides a unique identifier for the overlay. No two overlays shall have the same overlay_id value.

overlay_control_flag[i] when set to 1 defines that the structure as defined by the i-th overlay_control_struct[i] is present. OMAF players shall allow both values of

overlay_control_flag[i] for all values of i.

overlay_control_essential_flag[i] equal to 0 specifies that OMAF players are not required to process the structure as defined by the i-th overlay_control_struct[i]. overlay_control_essential_flag[i] equal to 1 specifies that OMAF players shall process the structure as defined by the i-th overlay_control_struct[i]. When overlay_control_essential_flag[i] equal to 1 and an OMAF player is not capable of parsing or processing the structure as defined by the i-th overlay_control_struct[i], the OMAF player shall display neither the overlays specified by this structure nor the background visual media.

byte_count[i] gives the byte count of the structure represented by the i-th overlay_control_struct[i].
overlay_control_struct[i][byte_count[i]] defines the i-th structure with a byte count as defined by byte_count[i].

Wang further includes a dynamic overlay timed metadata track which indicates which overlays are active at particular times and depending upon the application the active overlay(s) may change over time and indicates overlay parameters that may be dynamically changing over time. In Wang, the overlay timed metadata track is linked to the respective visual media tracks by utilizing the 'cdsc' track reference. In Wang, a dynamic overlay timed metadata track includes the following sample entry structure and sample syntax and semantics:

Sample Entry

```
aligned(8) class OverlaySampleEntry extends MetadataSampleEntry('dyol')
{
        OverlayConfigBox();
}
```

The sample entry of an overlay timed metadata track contains an OverlayConfigBox that includes the default syntax element values of OverlayStruct that apply selectively when both of the following conditions are true:

- The same overlay_id is present in a sample.
- When byte_count[i] is present and equal to 0 for a particular overlay_id in the OverlayStruct of an overlay timed metadata sample, overlay_control_struct[j][byte_count[j]] of the sample entry and for the same overlay_id value applies.

Sample

```
aligned(8) class OverlaySample {
        unsigned int(15) num_active_overlays_by_id;
        unsigned int(1)  addl_active_overlays_flag;
        for (i = 0; i < num_active_overlays_by_id; i++)
                unsigned int(16) active_overlay_id;
        if(addl_active_overlays_flag)
                OverlayStruct();
}
```

num_active_overlays_by_id specifies the number of overlays from the OverlayStruct() structure signalled in the sample entry OverlaySampleEntry that are active. A value of 0 indicates that no overlays from the sample entry are active.

addl_active_overlays_flag equal to 1 specifies that additional active overlays are signalled in the sample directly in the overlay structure (OverlayStruct()). addl_active_overlays_flag equal to 0 specifies that no additional active overlays are signalled in the sample directly in the overlay structure (OverlayStruct()).

active_overlay_id provides overlay identifier for the overlay signalled from the sample entry, which is currently active. For each active_overlay_id, the OverlayStruct() structure in the sample entry OverlaySampleEntry shall include an overlay with a matching overlay_id value.

An OMAF player shall display only the active overlays at any particular time and shall not display inactive overlays. num_overlays of a sample is not required to be equal to num_overlays in the sample entry, and the set of overlay_id values of a sample is not required to be the same as the set of overlay_id values in the sample entry.

Activation of particular overlays by a sample results in deactivation of any previously signalled overlays from previous sample(s).

Wang further provides where the association of timed metadata tracks to media tracks or track groups includes the (1) Association with media tracks by the 'cdsc' track reference and (2) Association with media tracks by the 'cdtg' track reference. When a timed metadata track is linked to one or more media tracks with a 'cdsc' track reference, it describes each media track individually.

[0030] A timed metadata track containing a 'cdtg' track reference describes the referenced media tracks and track groups collectively. The 'cdtg' track reference shall only be present in timed metadata tracks. A timed metadata track containing 'cdsc' track reference to a track_group_id value describes each track in the track group individually. When a timed metadata track contains a 'cdtg' track reference to a track group of type '2dcc', the timed metadata track describes the composition pictures.

[0031]

As described above, Wang specifies projection and rectangular region-wise packing methods that may be used for conversion of a spherical video sequence into a two-dimensional rectangular video sequence. In this manner, Wang specifies a region-wise packing structure having the following definition, syntax, and semantics:

Definition

RegionWisePackingStruct specifies the mapping between packed regions and the respective projected regions and specifies the location and size of the guard bands, if any.

NOTE: Among other information the RegionWisePackingStruct also provides the content coverage information in the 2D Cartesian picture domain.

A decoded picture in the semantics of this clause is either one of the following depending on the container for this syntax structure:

- For video, the decoded picture is the decoding output resulting from a sample of the video track.
- For an image item, the decoded picture is a reconstructed image of the image item.

The content of RegionWisePackingStruct is informatively summarized below, while the normative semantics follow subsequently in this clause:

- The width and height of the projected picture are explicitly signalled with proj_picture_width and proj_picture_height, respectively.
- The width and height of the packed picture are explicitly signalled with packed_picture_width and packed_picture_height, respectively.
- When the projected picture is stereoscopic and has the top-bottom or side-by-side frame packing arrangement, constituent_picture_matching_flag equal to 1 specifies that
  - o the projected region information, packed region information, and guard band region information in this syntax structure apply individually to each constituent picture,
  - o the packed picture and the projected picture have the same stereoscopic frame packing format, and
  - o the number of projected regions and packed regions is double of that indicated by the value of num_regions in the syntax structure.
- RegionWisePackingStruct contains a loop, in which a loop entry corresponds to the respective projected regions and packed regions in both constituent pictures (when constituent_picture_matching_flag equal to 1) or to a projected region and the respective packed region (when constituent_picture_matching_flag equal to 0), and the loop entry the contains the following:
  - o a flag indicating the presence of guard bands for the packed region,
  - o the packing type (however, only rectangular region-wise packing is

specified in Wang),

- o the mapping between a projected region and the respective packed region in the rectangular region packing structure RectRegionPacking(i),
- o when guard bands are present, the guard band structure for the packed region GuardBand(i).

The content of the rectangular region packing structure RectRegionPacking(i) is informatively summarized below, while the normative semantics follow subsequently in this clause:

- proj_reg_width[i], proj_reg_height[i], proj_reg_top[i], and proj_reg_left[i] specify the width, height, top offset, and left offset, respectively, of the i-th projected region.
- transform_type[i] specifies the rotation and mirroring, if any, that are applied to the i-th packed region to remap it to the i-th projected region.
- packed_reg_width[i], packed_reg_height[i], packed_reg_top[i], and packed_reg_left[i] specify the width, height, the top offset, and the left offset, respectively, of the i-th packed region.

The content of the guard band structure GuardBand(i) is informatively summarized below, while the normative semantics follow subsequently in this clause:

- left_gb_width[i], right_gb_width[i], top_gb_height[i], or bottom_gb_height[i] specify the guard band size on the left side of, the right side of, above, or below, respectively, the i-th packed region.
- gb_not_used_for_pred_flag[i] indicates if the encoding was constrained in a manner that guards bands are not used as a reference in the inter prediction process.
- gb_type[i][j] specifies the type of the guard bands for the i-th packed region.

FIG. 6 illustrates an example of the position and size of a projected region within a projected picture (on the left side) as well as that of a packed region within a packed picture with guard bands (on the right side). This example applies when the value of constituent_picture_matching_flag is equal to 0.

Syntax

```
aligned(8) class RectRegionPacking(i) {
    unsigned int(32) proj_reg_width[i];
    unsigned int(32) proj_reg_height[i];
    unsigned int(32) proj_reg_top[i];
    unsigned int(32) proj_reg_left[i];
    unsigned int(3)  transform_type[i];
    bit(5) reserved = 0;
    unsigned int(16) packed_reg_width[i];
    unsigned int(16) packed_reg_height[i];
    unsigned int(16) packed_reg_top[i];
    unsigned int(16) packed_reg_left[i];
```

}

Semantics

proj_reg_width[i], proj_reg_height[i], proj_reg_top[i], and proj_reg_left[i] specify the width, height, top offset, and left offset, respectively, of the i-th projected region, either within the projected picture (when constituent_picture_matching_flag is equal to 0) or within the constituent picture of the projected picture (when constituent_picture_matching_flag is equal to 1). proj_reg_width[i], proj_reg_height[i], proj_reg_top[i] and proj_reg_left[i] are indicated in relative projected picture sample units.

NOTE 1: Two projected regions may partially or entirely overlap with each other. When there is an indication of quality difference, e.g., by a region-wise quality ranking indication, then for the overlapping area of any two overlapping projected regions, the packed region corresponding to the projected region that is indicated to have higher quality should be used for rendering.

transform_type[i] specifies the rotation and mirroring that is applied to the i-th packed region to remap it to the i-th projected region. When transform_type[i] specifies both rotation and mirroring, rotation is applied before mirroring for converting sample locations of a packed region to sample locations of a projected region. The following values are specified:

0: no transform
1: mirroring horizontally
2: rotation by 180 degrees (counter-clockwise)
3: rotation by 180 degrees (counter-clockwise) before mirroring horizontally
4: rotation by 90 degrees (counter-clockwise) before mirroring horizontally
5: rotation by 90 degrees (counter-clockwise)
6: rotation by 270 degrees (counter-clockwise) before mirroring horizontally
7: rotation by 270 degrees (counter-clockwise)

NOTE 2: Wang specifies the semantics of transform_type[i] for converting a sample location of a packed region in a packed picture to a sample location of a projected region in a projected picture.

packed_reg_width[i], packed_reg_height[i], packed_reg_top[i], and packed_reg_left[i] specify the width, height, the offset, and the left offset, respectively, of the i-th packed region, either within the packed picture (when constituent_picture_matching_flag is equal to 0) or within each constituent picture of the packed picture (when constituent_picture_matching_flag is equal to 1). packed_reg_width[i], packed_reg_height[i], packed_reg_top[i], and packed_reg_left[i] are indicated in relative packed picture sample units. packed_reg_width[i], packed_reg_height[i], packed_reg_top[i], and packed_reg_left[i] shall represent integer horizontal and vertical coordinates of luma sample units within the decoded pictures.

NOTE: Two packed regions may partially or entirely overlap with each other.

Wang further specifies the inverse of the rectangular region-wise packing process for remapping of a luma sample location in a packed region onto a luma sample location of the corresponding projected region:

Inputs to this process are:

- sample location (x, y) within the packed region, where x and y are in relative packed picture sample units, while the sample location is at an integer sample location within the packed picture,
- the width and the height (projRegWidth, projRegHeight) of the projected region, in relative projected picture sample units,
- the width and the height (packedRegWidth, packedRegHeight) of the packed region, in relative packed picture sample units,
- transform type (transformType), and
- offset values for the sampling position (offsetX, offsetY) in the range of 0, inclusive, to 1, exclusive, in horizontal and vertical relative packed picture sample units, respectively.

NOTE: offsetX and offsetY both equal to 0.5 indicate a sampling position that is in the centre point of a sample in packed picture sample units.

Outputs of this process are:

- the centre point of the sample location (hPos, vPos) within the projected region, where hPos and vPos are in relative projected picture sample units and may have non-integer real values.

The outputs are derived as follows:

```
if( transformType == 0 || transformType == 1 || transformType == 2
|| transformType == 3 ) {
        horRatio = projRegWidth ÷ packedRegWidth
        verRatio = projRegHeight ÷ packedRegHeight
 } else if ( transformType == 4 || transformType == 5 || transformType
== 6 || transformType == 7 ) {
        horRatio = projRegWidth ÷ packedRegHeight
        verRatio = projRegHeight ÷ packedRegWidth
 }
 if( transformType == 0 ) {
        hPos = horRatio * ( x + offsetX )
        vPos = verRatio * ( y + offsetY )
 } else if ( transformType == 1 ) {
        hPos = horRatio * ( packedRegWidth − x − offsetX ) (5 4)
        vPos = verRatio * ( y + offsetY )
 } else if ( transformType == 2 ) {
```

```
            hPos = horRatio * ( packedRegWidth – x – offsetX )
            vPos = verRatio * ( packedRegHeight – y – offsetY )
    } else if ( transformType  ==  3 ) {
            hPos = horRatio * ( x + offsetX )
            vPos = verRatio * ( packedRegHeight – y – offsetY )
    } else if ( transformType  ==  4 ) {
            hPos = horRatio * ( y + offsetY )
            vPos = verRatio * ( x + offsetX )
    } else if ( transformType  ==  5 ) {
            hPos = horRatio * ( y + offsetY )
            vPos = verRatio * ( packedRegWidth – x – offsetX )
    } else if ( transformType  == 6 ) {
            hPos = horRatio * ( packedRegHeight – y – offsetY )
            vPos = verRatio * ( packedRegWidth – x – offsetX )
    } else if ( transformType  ==  7 ) {
            hPos = horRatio * ( packedRegHeight – y – offsetY )
            vPos = verRatio * ( x+ offsetX )
    }
```

It should be noted that for the sake for brevity the complete syntax and semantics of the rectangular region packing structure, the guard band structure, and the region-wise packing structure are not provide herein. Further, the complete derivation of region-wise packing variables and constraints for the syntax elements of the region-wise packing structure are not provide herein. However, reference is made to the relevant section of Wang.

[0032] As described above, Wang specifies encapsulation, signaling, and streaming of omni-directional media in a media streaming system. In particular, Wang specifies how to encapsulate, signal, and stream omnidirectional media using dynamic adaptive streaming over Hypertext Transfer Protocol (HTTP) (DASH). DASH is described in ISO/IEC: ISO/IEC 23009-1:2014, "Information technology - Dynamic adaptive streaming over HTTP (DASH) - Part 1: Media presentation description and segment formats," International Organization for Standardization, 2nd Edition, 5/15/2014 (hereinafter, "ISO/IEC 23009-1:2014"), which is incorporated by reference herein. A DASH media presentation may include data segments, video segments, and audio segments. In some examples, a DASH Media Presentation may correspond to a linear service or part of a linear service of a given duration defined by a service provider (e.g., a single TV program, or the set of contiguous linear TV programs over a period of time). According to DASH, a Media Presentation Description (MPD) is a document that includes metadata required by a DASH Client to construct appropriate HTTP-URLs to access segments and to provide the streaming service to the user. A MPD document fragment may include a set of eXtensible Markup Language

(XML)-encoded metadata fragments. The contents of the MPD provide the resource identifiers for segments and the context for the identified resources within the Media Presentation. The data structure and semantics of the MPD fragment are described with respect to ISO/IEC 23009-1:2014. Further, it should be noted that draft editions of ISO/IEC 23009-1 are currently being proposed. Thus, as used herein, a MPD may include a MPD as described in ISO/IEC 23009-1:2014, currently proposed MPDs, and/or combinations thereof. In ISO/IEC 23009-1:2014, a media presentation as described in a MPD may include a sequence of one or more Periods, where each Period may include one or more Adaptation Sets. It should be noted that in the case where an Adaptation Set includes multiple media content components, then each media content component may be described individually. Each Adaptation Set may include one or more Representations. In ISO/IEC 23009-1:2014 each Representation is provided: (1) as a single Segment, where Subsegments are aligned across Representations with an Adaptation Set; and (2) as a sequence of Segments where each Segment is addressable by a template-generated Universal Resource Locator (URL). The properties of each media content component may be described by an AdaptationSet element and/or elements within an Adaption Set, including for example, a ContentComponent element. It should be noted that the sphere region structure forms the basis of DASH descriptor signaling for various descriptors.

[0033]

Wang provides where a timed metadata track, e.g., of track sample entry type 'invo,' 'rcvp,' or 'dyol' as specified above, may be encapsulated in a DASH representation, where the @associationId attribute of this metadata representation shall contain one or more values of the @id attribute of the representation(s) containing the omnidirectional media carried by the media track(s) that are associated with the timed metadata track through a 'cdsc' track reference as specified above and the @associationType attribute of this metadata representation shall be equal to 'cdsc'. Wang further provides the following with respect to signaling of association:

A SupplementalProperty element with a @schemeIdUri attribute equal to "urn:mpeg:mpegI:omaf:2018:assoc" is referred to as an association descriptor.

One or more association descriptors may be present at adaptation set level, representation level, preselection level.

An association descriptor included inside an adaptation set/ representation/ preselection element indicates that the parent element of this element's descriptor (i.e. adaptation set/ representation/ preselection element) is associated with one or more elements in the MPD indicated by the XPath query in the omaf2:association element and the association type signaled by omaf2:@associationKindList.

The @value attribute of the association descriptor shall not be present. The association descriptor shall include one or more association elements with attribute as specified in Table 2.

Elements and attributes for association descriptor	**Use**	**Data type**	**Description**
association	0..N	omaf2:AssociationType	Element which specifies a list of XPath query string(s) which are evaluated to determine the elements (including certain values for their

			attributes) that are associated with the parent element of this association element's descriptor. The XPath query shall evaluate to one or more elements.
association@associationKindList	M	omaf2: listOfAssociationValues	Values in this list specify the kind of association between the parent element of this association element's descriptor and the elements it is associated with. If this list includes a single entry then the parent element of this association element's descriptor is associated collectively with all the elements resulting from evaluation of all XPath queries signaled in this association element with the kind of association indicated by this attribute. If this list includes multiple entries then the number of entries in the list shall be equal to the number of entries in the list signaled in this attribute's association element. In this case the parent element of this association element's descriptor is

			associated with the element(s) specified by corresponding collocated XPath query in the association element individually (if the XPath query results in a single element) or collectively if the XPath query results in multiple elements with the type of association indicated by the collocated value in this attribute.

Table 2

The data types for various elements and attributes shall be as defined in the XML schema. A XML schema for this shall be as shown below. The schema shall be represented in a XML schema that has namespace urn:mpeg:mpegI:omaf:2018 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="urn:mpeg:mpegI:omaf:2018"
    xmlns:omaf="urn:mpeg:mpegI:omaf:2017"
    xmlns:omaf2="urn:mpeg:mpegI:omaf:2018"
    elementFormDefault="qualified">
    <xs:element name="association" type="omaf2:AssociationType"/>
    <xs:simpleType name="listOfAssociationValues">
        <xs:restriction>
            <xs:simpleType>
                <xs:list itemType="xs:string"/>
```

```
                </xs:simpleType>
                <xs:minLength value="1"/>
            </xs:restriction>
        </xs:simpleType>
        <xs:complexType name="AssociationType">
            <xs:simpleContent>
                <xs:extension base="omaf2:listOfAssociationValues">
                    <xs:attribute name="associationKindList"
type="omaf2:listOfAssociationValues" use="required"/>
                    <xs:anyAttribute processContents="skip"/>
                </xs:extension>
            </xs:simpleContent>
        </xs:complexType>
</xs:schema>
```

Wang further provides the following with respect to signaling of sub-picture representations:

Sub-picture representations carrying sub-picture tracks belonging to the same 2D spatial relationship track group may be indicated by a sub-picture composition identifier element SubPicCompositionId signaled as a child element of AdaptationSet element as specified in Table 3.

SubPicCompositionId element may be present at adaptation set level and shall not be present at any other level.

Element	Use	Data type	Description
SubPicCompositionId	0..N	xs: unsignedShort	Specifies the identifier of Adaptation Set which includes sub-picture representations carrying sub-picture tracks belonging to the same 2D spatial relationship track group. All Adaptation Sets in a Period that have the same value of SubPicCompositionId together form a sub-picture composition.

Table 3

The data type for the element shall be as defined in the XML schema. An XML schema for this element shall be as shown below. The normative schema shall be represented in an XML schema that has namespace urn:mpeg:mpegI:omaf:2018 and is specified as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
        targetNamespace="urn:mpeg:mpegI:omaf:2018"
        xmlns:omaf2="urn:mpeg:mpegI:omaf:2018"
        elementFormDefault="qualified">
    <xs:element name="SubPicCompositionId" type="xs:unsignedShort"/>
</xs:schema>
```

As described above Wang provides an association descriptor which allows specifying association between DASH adaptation sets/ representations and a sub-picture com-

position identifier element (SubPicCompositionId) which may be signalled as a child element of AdaptationSet element. However, Wang fails to provide a mechanism for signaling the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation. As described in further detail below, the techniques described herein, may be used for signaling the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation.

[0034] FIG. 1 is a block diagram illustrating an example of a system that may be configured to code (i.e., encode and/or decode) video data according to one or more techniques of this disclosure. System 100 represents an example of a system that may encapsulate video data according to one or more techniques of this disclosure. As illustrated in FIG. 1, system 100 includes source device 102, communications medium 110, and destination device 120. In the example illustrated in FIG. 1, source device 102 may include any device configured to encode video data and transmit encoded video data to communications medium 110. Destination device 120 may include any device configured to receive encoded video data via communications medium 110 and to decode encoded video data. Source device 102 and/or destination device 120 may include computing devices equipped for wired and/or wireless communications and may include, for example, set top boxes, digital video recorders, televisions, desktop, laptop or tablet computers, gaming consoles, medical imagining devices, and mobile devices, including, for example, smartphones, cellular telephones, personal gaming devices.

[0035] Communications medium 110 may include any combination of wireless and wired communication media, and/or storage devices. Communications medium 110 may include coaxial cables, fiber optic cables, twisted pair cables, wireless transmitters and receivers, routers, switches, repeaters, base stations, or any other equipment that may be useful to facilitate communications between various devices and sites. Communications medium 110 may include one or more networks. For example, communications medium 110 may include a network configured to enable access to the World Wide Web, for example, the Internet. A network may operate according to a combination of one or more telecommunication protocols. Telecommunications protocols may include proprietary aspects and/or may include standardized telecommunication protocols. Examples of standardized telecommunications protocols include Digital Video Broadcasting (DVB) standards, Advanced Television Systems Committee (ATSC) standards, Integrated Services Digital Broadcasting (ISDB) standards, Data Over Cable Service Interface Specification (DOCSIS) standards, Global System Mobile Communications (GSM) standards, code division multiple access (CDMA) standards, 3rd Generation Partnership Project (3GPP) standards, European Telecom-

munications Standards Institute (ETSI) standards, Internet Protocol (IP) standards, Wireless Application Protocol (WAP) standards, and Institute of Electrical and Electronics Engineers (IEEE) standards.

[0036] Storage devices may include any type of device or storage medium capable of storing data. A storage medium may include a tangible or non-transitory computer-readable media. A computer readable medium may include optical discs, flash memory, magnetic memory, or any other suitable digital storage media. In some examples, a memory device or portions thereof may be described as non-volatile memory and in other examples portions of memory devices may be described as volatile memory. Examples of volatile memories may include random access memories (RAM), dynamic random access memories (DRAM), and static random access memories (SRAM). Examples of non-volatile memories may include magnetic hard discs, optical discs, floppy discs, flash memories, or forms of electrically programmable memories (EPROM) or electrically erasable and programmable (EEPROM) memories. Storage device(s) may include memory cards (e.g., a Secure Digital (SD) memory card), internal/external hard disk drives, and/or internal/external solid state drives. Data may be stored on a storage device according to a defined file format.

[0037] FIG. 7 is a conceptual drawing illustrating an example of components that may be included in an implementation of system 100. In the example implementation illustrated in FIG. 7, system 100 includes one or more computing devices 402A-402N, television service network 404, television service provider site 406, wide area network 408, local area network 410, and one or more content provider sites 412A-412N. The implementation illustrated in FIG. 7 represents an example of a system that may be configured to allow digital media content, such as, for example, a movie, a live sporting event, etc., and data and applications and media presentations associated therewith to be distributed to and accessed by a plurality of computing devices, such as computing devices 402A-402N. In the example illustrated in FIG. 7, computing devices 402A-402N may include any device configured to receive data from one or more of television service network 404, wide area network 408, and/or local area network 410. For example, computing devices 402A-402N may be equipped for wired and/or wireless communications and may be configured to receive services through one or more data channels and may include televisions, including so-called smart televisions, set top boxes, and digital video recorders. Further, computing devices 402A-402N may include desktop, laptop, or tablet computers, gaming consoles, mobile devices, including, for example, “smart” phones, cellular telephones, and personal gaming devices.

[0038] Television service network 404 is an example of a network configured to enable digital media content, which may include television services, to be distributed. For

example, television service network 404 may include public over-the-air television networks, public or subscription-based satellite television service provider networks, and public or subscription-based cable television provider networks and/or over the top or Internet service providers. It should be noted that although in some examples television service network 404 may primarily be used to enable television services to be provided, television service network 404 may also enable other types of data and services to be provided according to any combination of the telecommunication protocols described herein. Further, it should be noted that in some examples, television service network 404 may enable two-way communications between television service provider site 406 and one or more of computing devices 402A-402N. Television service network 404 may comprise any combination of wireless and/or wired communication media. Television service network 404 may include coaxial cables, fiber optic cables, twisted pair cables, wireless transmitters and receivers, routers, switches, repeaters, base stations, or any other equipment that may be useful to facilitate communications between various devices and sites. Television service network 404 may operate according to a combination of one or more telecommunication protocols. Telecommunications protocols may include proprietary aspects and/or may include standardized telecommunication protocols. Examples of standardized telecommunications protocols include DVB standards, ATSC standards, ISDB standards, DTMB standards, DMB standards, Data Over Cable Service Interface Specification (DOCSIS) standards, HbbTV standards, W3C standards, and UPnP standards.

[0039] Referring again to FIG. 7, television service provider site 406 may be configured to distribute television service via television service network 404. For example, television service provider site 406 may include one or more broadcast stations, a cable television provider, or a satellite television provider, or an Internet-based television provider. For example, television service provider site 406 may be configured to receive a transmission including television programming through a satellite uplink/downlink. Further, as illustrated in FIG. 7, television service provider site 406 may be in communication with wide area network 408 and may be configured to receive data from content provider sites 412A-412N. It should be noted that in some examples, television service provider site 406 may include a television studio and content may originate therefrom.

[0040] Wide area network 408 may include a packet based network and operate according to a combination of one or more telecommunication protocols. Telecommunications protocols may include proprietary aspects and/or may include standardized telecommunication protocols. Examples of standardized telecommunications protocols include Global System Mobile Communications (GSM) standards, code division multiple

access (CDMA) standards, 3rd Generation Partnership Project (3GPP) standards, European Telecommunications Standards Institute (ETSI) standards, European standards (EN), IP standards, Wireless Application Protocol (WAP) standards, and Institute of Electrical and Electronics Engineers (IEEE) standards, such as, for example, one or more of the IEEE 802 standards (e.g., Wi-Fi). Wide area network 408 may comprise any combination of wireless and/or wired communication media. Wide area network 480 may include coaxial cables, fiber optic cables, twisted pair cables, Ethernet cables, wireless transmitters and receivers, routers, switches, repeaters, base stations, or any other equipment that may be useful to facilitate communications between various devices and sites. In one example, wide area network 408 may include the Internet. Local area network 410 may include a packet based network and operate according to a combination of one or more telecommunication protocols. Local area network 410 may be distinguished from wide area network 408 based on levels of access and/or physical infrastructure. For example, local area network 410 may include a secure home network.

[0041] Referring again to FIG. 7, content provider sites 412A-412N represent examples of sites that may provide multimedia content to television service provider site 406 and/or computing devices 402A-402N. For example, a content provider site may include a studio having one or more studio content servers configured to provide multimedia files and/or streams to television service provider site 406. In one example, content provider sites 412A-412N may be configured to provide multimedia content using the IP suite. For example, a content provider site may be configured to provide multimedia content to a receiver device according to Real Time Streaming Protocol (RTSP), HTTP, or the like. Further, content provider sites 412A-412N may be configured to provide data, including hypertext based content, and the like, to one or more of receiver devices computing devices 402A-402N and/or television service provider site 406 through wide area network 408. Content provider sites 412A-412N may include one or more web servers. Data provided by data provider site 412A-412N may be defined according to data formats.

[0042] Referring again to FIG. 1, source device 102 includes video source 104, video encoder 106, data encapsulator 107, and interface 108. Video source 104 may include any device configured to capture and/or store video data. For example, video source 104 may include a video camera and a storage device operably coupled thereto. Video encoder 106 may include any device configured to receive video data and generate a compliant bitstream representing the video data. A compliant bitstream may refer to a bitstream that a video decoder can receive and reproduce video data therefrom. Aspects of a compliant bitstream may be defined according to a video coding standard. When generating a compliant bitstream video encoder 106 may compress video data.

Compression may be lossy (discernible or indiscernible to a viewer) or lossless.

[0043] Referring again to FIG. 1, data encapsulator 107 may receive encoded video data and generate a compliant bitstream, e.g., a sequence of NAL units according to a defined data structure. A device receiving a compliant bitstream can reproduce video data therefrom. It should be noted that the term conforming bitstream may be used in place of the term compliant bitstream. It should be noted that data encapsulator 107 need not necessary be located in the same physical device as video encoder 106. For example, functions described as being performed by video encoder 106 and data encapsulator 107 may be distributed among devices illustrated in FIG. 7.

[0044] In one example, data encapsulator 107 may include a data encapsulator configured to receive one or more media components and generate media presentation based on DASH. FIG. 8 is a block diagram illustrating an example of a data encapsulator that may implement one or more techniques of this disclosure. Data encapsulator 500 may be configured to generate a media presentation according to the techniques described herein. In the example illustrated in FIG. 8, functional blocks of component encapsulator 500 correspond to functional blocks for generating a media presentation (e.g., a DASH media presentation). As illustrated in FIG. 8, component encapsulator 500 includes media presentation description generator 502, segment generator 504, and system memory 506. Each of media presentation description generator 502, segment generator 504, and system memory 506 may be interconnected (physically, communicatively, and/or operatively) for inter-component communications and may be implemented as any of a variety of suitable circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. It should be noted that although data encapsulator 500 is illustrated as having distinct functional blocks, such an illustration is for descriptive purposes and does not limit data encapsulator 500 to a particular hardware architecture. Functions of data encapsulator 500 may be realized using any combination of hardware, firmware and/or software implementations.

[0045] Media presentation description generator 502 may be configured to generate media presentation description fragments. Segment generator 504 may be configured to receive media components and generate one or more segments for inclusion in a media presentation. System memory 506 may be described as a non-transitory or tangible computer-readable storage medium. In some examples, system memory 506 may provide temporary and/or long-term storage. In some examples, system memory 506 or portions thereof may be described as non-volatile memory and in other examples portions of system memory 506 may be described as volatile memory. System memory 506 may be configured to store information that may be used by data encapsulator

during operation.

[0046] As described above, Wang fails to provide a mechanism for signaling the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation. In one example, according to the techniques described herein, data encapsulator 107 may be configured to signal the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation. In one example, data encapsulator 107 may be configured to signal the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation according to the following rules:

For collective association a timed metadata track, e.g., 'invo,' 'rcvp,' or 'dyol', may be encapsulated in a DASH representation.
When a timed metadata track of the track sample entry type 'invo' or 'rcvp' or 'dyol' encapsulated in a DASH representation is associated with a sub-picture composition, an association descriptor shall be present as a child element of the DASH Representation element.
In this case the association descriptor shall:

- Include one string in the association element of the type //AdaptationSet[SubPicCompositionId="aa"], where "aa" indicates the sub-picture composition identifier value.

It should be noted that whereas the above example includes the XPath query //AdaptationSet[SubPicCompositionId="aa"] which identifies all the Adaptation Sets which have the SubPicCompositionId element having a certain value (e.g. value "aa"), other equivalent XPath queries could instead be used to specify all (or one or more Adaptation Sets which have the same SubPicCompositionId value) and are intended to be covered by this requirement.

- Include 'cdtg' as the value of the association@associationKindList attribute of the association element.

In this case the timed metadata track in the encapsulated DASH representation which contain the above described association descriptor is associated with the referenced sub-picture composition signaled via association element string collectively.

Further, in one example, the collective association a timed metadata track may be encapsulated in a DASH representation, where the @associationId attribute of this metadata representation may contain one or more values of the @id attribute of the representation(s) containing the omnidirectional media carried by the media track(s) that are associated with the timed metadata track through a 'cdtg' track reference as specified above. The @associationType attribute of this metadata representation shall be equal to 'cdtg'. This describes timed metadata track association with each DASH representation indicated by @id attribute collectively.

Further in this case the individual association may be defined as follows:

For individual association a timed metadata track, e.g., of track sample entry type 'invo', 'rcvp', or 'dyol', may be encapsulated in a DASH representation. The @associationId attribute of this metadata representation shall contain one or more values of the @id attribute of the representation(s) containing the omnidirectional media carried by the media track(s) that are associated with the timed metadata track through a 'cdsc' track reference. The @associationType attribute of this metadata representation shall be equal to 'cdsc'. This describes timed metadata track association with each DASH representation indicated by @id attribute individually.

In another example, the above text for collective association may be only applied under an otherwise condition in combination with the rules above. In this case, the rules for association may be as follows:

For collective association a timed metadata track, e.g., 'invo,' 'rcvp,' or 'dyol', may be encapsulated in a DASH representation.
When a timed metadata track of the track sample entry type 'invo' or 'rcvp' or 'dyol' encapsulated in a DASH representation is associated with a sub-picture composition, an association descriptor shall be present as a child element of the DASH Representation element.
In this case the association descriptor shall:

- Include one string in the association element of the type //AdaptationSet[SubPicCompositionId="aa"], where "aa" indicates the sub-picture composition identifier value.

It should be noted that whereas the above example includes the XPath query //AdaptationSet[SubPicCompositionId="aa"] which identifies all the Adaptation Sets which have the SubPicCompositionId element having a certain value (e.g. value "aa"), other equivalent XPath queries could instead be used to specify all (or one or more Adaptation Sets which have the same SubPicCompositionId value) and are intended to be covered by this requirement.

Further, it should be noted that in one example, the type //AdaptationSet[SubPicCompositionId="aa"] may be modified as follows: //AdaptationSet[omaf2:SubPicCompositionId="aa"]

In this case omaf2: namespace in which SubPicCompositionId is defined is included as part of the string. The omaf2 namespace may correspond to XML namespace "urn:mpeg:mpegI:omaf:2018" and as such may be defined by the declaration:

xmlns:omaf2="urn:mpeg:mpegI:omaf:2018"

- Include 'cdtg' as the value of the association@associationKindList attribute of the association element.

In this case, the timed metadata track in the encapsulated DASH representation which contain the above described association descriptor is associated with the referenced sub-picture composition signaled via association element string collectively.

Otherwise (i.e., for collective association other than sub-picture composition association), the @associationId attribute of this metadata representation may contain one or more values of the @id attribute of the representation(s) containing the omnidirectional media carried by the media track(s) that are associated with the timed metadata track through a 'cdtg' track reference as specified above. The @associationType attribute of this metadata representation shall be equal to 'cdtg'. This describes timed metadata track association with each DASH representation indicated by @id attribute collectively.

In one example, according to the techniques described herein, a timed metadata representation may be collectively associated with all media representations of a viewpoint as follows: An association descriptor is present as a child element of the DASH Representation element of the timed metadata representation, and the association descriptor shall:

- Include one string in the Association element as follows: //AdaptationSet/Viewpoint[@schemeIdUri="urn:mpeg:mpegI:omaf:2018:vwpt" and @value="bb"]/..

  where "bb" indicates the viewpoint ID value of the viewpoint as a string.
- Include 'cdtg' as the value of the Association@associationKindList attribute of the Association element.

The above string in the Association element selects all adaptation sets which have an OMAF V2 DASH viewpoint descriptor and have a particular viewpoint ID value ("bb" in the example above) within that descriptor. The "and" operator in the string above requires that only the OMAF V2 DASH viewpoint descriptor which also have the particular viewpoint ID value are selected. The "/.." part of the string selects the parent element.

It should be noted that the portion of the string "urn:mpeg:mpegI:omaf:2018:vwpt" may be changed to match the @schemeIdUri which matches the OMAF V2 DASH viewpoint descriptor. For example this may be changed to "urn:mpeg:mpegI:omaf:2019:vwpt" or other similar name.

As described above with respect to Table 2, Wang provides where an association descriptor included inside an adaptation set/representation/preselection element indicates that the parent element of this element's descriptor is associated with one or more elements in the MPD indicated by the XPath query in the omaf2:association element and the association type signaled by omaf2:@associationKindList. According to the techniques herein, in one example, a list @associationKindList may be constrained such that each value in the list is a four character code for track reference types registered in a MP4 registration authority, where a MP4 registration authority refers to a central or decentral entity which coordinates assignment and/or registration of all such four character codes such that each code is unique and unambiguously registered and used. One example of such a MP4 registration authority is MP4RA. MP4 may stand for MPEG 4. In such an example, the semantics of elements and attributes of the association descriptor in Table 2 may be modified as provided in Table 4.

Elements and attributes for association descriptor	**Use**	**Data type**	**Description**
association	0..N	omaf2:AssociationType	Element which specifies a list of XPath query string(s) which are evaluated to determine the elements (including certain values for their attributes) that are associated with the parent element of this association element's descriptor. The XPath

			query shall evaluate to one or more elements.
association@associationKindList	M	omaf2: listOfAssociationValues	Values in this list specify the kind of association between the parent element of this association element's descriptor and the elements it is associated with. Each Value in this list is 4 character code for track reference types registered in MP4 registration authority. If this list includes a single entry then the parent element of this association element's descriptor is associated collectively with all the elements resulting from evaluation of all XPath queries signaled in this association element with the kind of association indicated

			by this attribute. If this list includes multiple entries then the number of entries in the list shall be equal to the number of entries in the list signaled in this attribute's association element. In this case the parent element of this association element's descriptor is associated with the element(s) specified by corresponding collocated XPath query in the association element individually (if the XPath query results in a single element) or collectively if the XPath query results in multiple elements with the type of association indicated by the collocated value
			in this attribute.

Table 4

Further, as described above, an overlay may be defined as a rendering of one or more of videos, images and text over 360-degree video content. Background media may be defined as a piece of visual media on which an overlay is superimposed. Background media may be referred to as background visual media. Further, an overlay may be defined as piece of visual media rendered over omnidirectional video or image item or over a viewport. A visual media may be defined as video, image item or timed text. A viewport may be defined as region of omnidirectional image or video suitable for display and viewing by the user.

[0047] In some cases, one or more overlays may be associated with background media. For example, a logo may be overlaid on a background image. Examples include:

- overlaying a logo (note: the logo may not be rectangular and may use transparency);
- overlaying a sign language interpreter over the 360-degree video
- overlaying a small equi-rectangular projection of the entire 360-degree video as preview-window on top of the current viewport to be used as a guiding mechanism;
- overlaying a thumbnail of the recommended Viewport over the current Viewport.

In all these cases the overlay is associated with corresponding background media on which it is overlayed. The association may indicate that the corresponding overlay and background media are intended to be presented together.

In one example, the following constraints may be imposed such that an adaption set containing an overlay may be associated with an adaption set containing background media.

When an adaptation set containing an overlay is associated with one or more adaptation sets containing background media, an association descriptor shall be present as a child element of the adaptation set element containing the overlay.

In this case the association descriptor shall:

- Include a XPath string in the association element which evaluates to one or more adaptation set element(s) containing background media.
- Either include:
  - 'cdsc' as the value of the association@associationKindList attribute of the association element if the overlay applies individually to the background media.
  - 'cdtg' as the value of the association@associationKindList attribute of the association element if the overlay applies collectively to the background media (e.g. if the background media is signaled via multiple adaptation sets with each adaptation set corresponding to a sub-picture).

There can be multiple such association elements present inside an adaptation set containing an overlay.

When an adaptation set containing an overlay is associated with one or more adaptation set(s) containing background media as described above, they are intended to be presented together.

In another example, the following constraints may be imposed:

When an adaptation set containing an overlay is associated with one or more adaptation sets containg background media, an association descriptor shall be present as a child element of the adaptation set element containing the overlay.

In this case the association descriptor shall:

- Include a XPath string in the association element which evaluates to one or more adaptation set element(s) containing background media.
- Either:
  - One or more if the overlay applies individually to the background media. In this case the number of 'ovbg' values in this list shall be equal to the number of element(s) the XPath string in the association element above evaluates to.
  - A single 'ovbg' entry as the value of the association@associationKindList attribute of the association element if the overlay applies collectively to the background media (e.g. if the background media is signaled via multiple adaptation sets with each adaptation set corresponding to a sub-picture).

In another example, the following constraints may be imposed:

When an adaptation set containing an overlay is associated with one or more adaptation sets containg background media, an association descriptor shall be present as a child element of the adaptation set element containing the overlay.

In this case the association descriptor shall:

- Include a XPath string in the association element which evaluates to one or more adaptation set element(s) containing background media.
- Shall include one or more 'ovbg' value(s) for association@associationKindList attribute of the association element:
    - o If association@associationKindList includes one 'ovbg' value and number of element(s) the XPath string in the association element above evaluates to is greater to 1, the overlay applies collectively to the background media (e.g. if the background media is signaled via multiple adaptation sets with each adaptation set corresponding to a sub-picture).
    - o If association@associationKindList includes more than one 'ovbg' value and number of element(s) the XPath string in the association element above evaluates to is greater to 1, the number of entries in the list association@associationKindList shall be equal to the number of element(s) the XPath string in the association element above evaluates to. In this case the overlay applies individually to each the background media element the XPath string in the association element above evaluates to.
    - o If association@associationKindList includes only one 'ovbg' value and number of element(s) the XPath string in the association element above evaluates to is equal to 1, the overlay applies individually to the background media.

It should be noted that with respect to the sphere region structure specifying a sphere region in Wang2, in a case where both azimuth_range and elevation_range are equal to 0, the sphere region specified by this structure is a point on a spherical surface. Further, it should be noted that the azimuth_range and elevation_range syntax elements of SphereRegionStruct are optionally signalled controlled by the input argument, range_included_flag. However, the last byte of SphereRegionStruct which includes a bit for indication of interpolate and seven reserved bits is always signaled, where the semantics of interpolate syntax element are defined by the semantics of the structure which contain the instance of the SphereRegionStruct. It is asserted that in some typical cases, where SphereRegionStruct is used for signaling information, the interpolate syntax element may not be meaningful. As such, in this case the version of SphereRegionStruct in Wang, which does not allow exclusion of the last byte, may be inefficient as it wastes a byte. In one example, according to the techniques herein, a new sphere region structure, SphereRegionStruct2, is be defined which allows inclusion or exclusion of the last byte. In one example, according to the techniques herein, the following definition, syntax and semantics may be used for a new sphere region structure specifying a sphere region.

Definition

The sphere region structure (SphereRegionStruct2) specifies a sphere region.

When centre_tilt is equal to 0, the sphere region specified by this structure is derived as follows:

–If both azimuth_range and elevation_range are equal to 0, the sphere region specified by this structure is a point on a spherical surface.

–Otherwise, the sphere region is defined using variables centreAzimuth, centreElevation, cAzimuth1, cAzimuth, cElevation1, and cElevation2 derived as follows:

centreAzimuth = centre_azimuth ÷ 65536

centreElevation = centre_elevation ÷ 65536

cAzimuth1 = (centre_azimuth – azimuth_range ÷ 2) ÷ 65536

cAzimuth2 = (centre_azimuth + azimuth_range ÷ 2) ÷ 65536

cElevation1 = (centre_elevation – elevation_range ÷ 2) ÷ 65536

cElevation2 = (centre_elevation + elevation_range ÷ 2) ÷ 65536

The sphere region is defined as follows with reference to the shape type value specified in the semantics of the structure containing this instance of SphereRegionStruct2:

–When the shape type value is equal to 0, the sphere region is specified by four great circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in FIG. 5A.

–When the shape type value is equal to 1, the sphere region is specified by two azimuth circles and two elevation circles defined by four points cAzimuth1, cAzimuth2, cElevation1, cElevation2 and the centre point defined by centreAzimuth and centreElevation and as shown in FIG. 5B.

When centre_tilt is not equal to 0, the sphere region is firstly derived as above and then a tilt rotation is applied along the axis originating from the sphere origin passing through the centre point of the sphere region, where the angle value increases clockwise when looking from the origin towards the positive end of the axis. The final sphere region is the one after applying the tilt rotation.

Shape type value equal to 0 specifies that the sphere region is specified by four great circles as illustrated in FIG. 5A.

Shape type value equal to 1 specifies that the sphere region is specified by two azimuth circles and two elevation circles as illustrated in FIG. 5B.

Shape type values greater than 1 are reserved.

Syntax

```
aligned(8) SphereRegionStruct2(range_included_flag, interpolate_included_flag) {
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        singed int(32) centre_tilt;
        if (range_included_flag) {
                unsigned int(32) azimuth_range;
                unsigned int(32) elevation_range;
        }
        if (interpolate_included_flag) {
        unsigned int(1) interpolate;
        bit(7) reserved = 0;
        }
}
```

Semantics

centre_azimuth and centre_elevation specify the centre of the sphere region. centre_azimuth shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive. centre_elevation shall be in the range of $-90 * 2^{16}$ to $90 * 2^{16}$, inclusive.

centre_tilt specifies the tilt angle of the sphere region. centre_tilt shall be in the range of $-180 * 2^{16}$ to $180 * 2^{16} - 1$, inclusive.

azimuth_range and elevation_range, when present, specify the azimuth and elevation ranges, respectively, of the sphere region specified by this structure in units of 2^{-16} degrees. azimuth_range and elevation_range specify the range through the centre point of the sphere region, as illustrated by FIG. 5A or FIG. 5B. When azimuth_range and elevation_range are not present in this instance of SphereRegionStruct2, they are inferred as specified in the semantics of the structure containing this instance of SphereRegionStruct2. azimuth_range shall be in the range of 0 to $360 * 2^{16}$, inclusive. elevation_range shall be in the range of 0 to $180 * 2^{16}$, inclusive.

The semantics of interpolate are specified by the semantics of the structure containing this instance of SphereRegionStruct. When interpolate is not present in this instance of SphereRegionStruct2, it is inferred as specified in the semantics of the syntax structure containing this instance of SphereRegionStruct2.

In an example, interpolate_included_flag may be called last_byte_included_flag or some other name. In an example, SphereRegionStruct2 may instead be called SphereRegionStruct and all occurences of SphereRegionStruct in Wang and Wang2 and other OMAF standards/ working drafts may be changed as follows:

- All occurences of SphereRegionStruct(0) may be or shall be changed to SphereRegionStruct(0,1)
- All occurences of SphereRegionStruct(1) may be or shall be changed to SphereRegionStruct(1,1)

Thus SphereRegionStruct may be defined as follows:

```
aligned(8) SphereRegionStruct(range_included_flag, interpolate_included_flag) {
        signed int(32) centre_azimuth;
        signed int(32) centre_elevation;
        signed int(32) centre_tilt;
        if (range_included_flag) {
                unsigned int(32) azimuth_range;
                unsigned int(32) elevation_range;
        }
        if (interpolate_included_flag) {
        unsigned int(1) interpolate;
        bit(7) reserved = 0;
        }
}
```

It should be noted in some cases in Wang and Wang2 when SphereRegionStruct is included in another structure, the semantics and value for interpolate are unspecified. According to the techniques herein, interpolate may be inferred as follows in unspecified cases:

-When SphereRegionStruct() is included in the OmafTimedTextConfigBox, the following applies:
For the SphereRegionStruct(0) included in the OmafTimedTextConfigBox, interpolate is inferred to be equal to 0;
or in another example,
For the SphereRegionStruct(0) included in the OmafTimedTextConfigBox, interpolate shall be equal to 0.

In another example:
-When SphereRegionStruct() is included in the OmafTimedTextConfigBox, the following applies:
For the SphereRegionStruct(0) included in the OmafTimedTextConfigBox, interpolate is inferred to be equal to 1;
or in another example,
For the SphereRegionStruct(0) included in the OmafTimedTextConfigBox, interpolate shall be equal to 1.

- When the SphereRegionStruct is present in SphereRelativeOmniOverlay() (i.e., region_indication_type is equal to 1) the following applies:
interpolate is inferred to be equal to 0.
or in another example,
interpolate is inferred to be equal to 1.
In another example:
- When the SphereRegionStruct is present in SphereRelativeOmniOverlay() (i.e., region_indication_type is equal to 1) the following applies:
interpolate shall be equal to 0.
or in another example,
interpolate shall be equal to 1.

- For the SphereRegionStruct(1)included in the SphereRelative2DOverlay, the following applies:
interpolate is inferred to be equal to 0.
or in another example,
interpolate is inferred to be equal to 1.

In another example:
- For the SphereRegionStruct(1) included in the SphereRelative2DOverlay, the following applies:
interpolate shall be equal to 0.
or in another example,

interpolate shall be equal to 1.

- For the SphereRegionStruct(1) included in the AssociatedSphereRegion, the following applies:
interpolate is inferred to be equal to 0.
or in another example,
interpolate is inferred to be equal to 1.

In another example:
- For the SphereRegionStruct(1) included in the AssociatedSphereRegion, the following applies:
interpolate shall be equal to 0.
or in another example,
interpolate shall be equal to 1.

- For the SphereRegionStruct(1) included in the guide_region(), the following applies:
interpolate is inferred to be equal to 0.
or in another example,
interpolate is inferred to be equal to 1.

In another example:
- For the SphereRegionStruct(1)included in the guide_region(), the following applies:
interpolate shall be equal to 0.
or in another example,
interpolate shall be equal to 1.

In another example, in one or more cases above, the value of interpolate when not present may be inferred to be equal to 1.

[0048] In another example, in one or more cases above, the value of interpolate when not present may be inferred to be equal to 0.

[0049] In another example, in one or more cases above, the value of interpolate when not present shall be equal to 1.

[0050] In another example, in one or more cases above, the value of interpolate when not present shall be equal to 0.

[0051] Further, according to the techniques herein, when SphereRegionSampleEntry() is included in another structure, the values of static_azimuth_range and static_elevation_range may be inferred. For example, in one example, when SphereRegionSampleEntry() is included in TTSphereLocationSampleEntry(), the values of static_azimuth_range, and static_elevation_range are inferred to be equal to 0.

[0052] In this manner, data encapsulator 107 represents an example of a device configured to signal the association between Adaptation Sets corresponding to a sub-picture composition and a timed metadata representation.

[0053] Referring again to FIG. 1, interface 108 may include any device configured to receive data generated by data encapsulator 107 and transmit and/or store the data to a

communications medium. Interface 108 may include a network interface card, such as an Ethernet card, and may include an optical transceiver, a radio frequency transceiver, or any other type of device that can send and/or receive information. Further, interface 108 may include a computer system interface that may enable a file to be stored on a storage device. For example, interface 108 may include a chipset supporting Peripheral Component Interconnect (PCI) and Peripheral Component Interconnect Express (PCIe) bus protocols, proprietary bus protocols, Universal Serial Bus (USB) protocols, I^2C, or any other logical and physical structure that may be used to interconnect peer devices.

[0054] Referring again to FIG. 1, destination device 120 includes interface 122, data decapsulator 123, video decoder 124, and display 126. Interface 122 may include any device configured to receive data from a communications medium. Interface 122 may include a network interface card, such as an Ethernet card, and may include an optical transceiver, a radio frequency transceiver, or any other type of device that can receive and/or send information. Further, interface 122 may include a computer system interface enabling a compliant video bitstream to be retrieved from a storage device. For example, interface 122 may include a chipset supporting PCI and PCIe bus protocols, proprietary bus protocols, USB protocols, I^2C, or any other logical and physical structure that may be used to interconnect peer devices. Data decapsulator 123 may be configured to receive a bitstream generated by data encaspulator 107 and perform sub-bitstream extraction according to one or more of the techniques described herein.

[0055] Video decoder 124 may include any device configured to receive a bitstream and/or acceptable variations thereof and reproduce video data therefrom. Display 126 may include any device configured to display video data. Display 126 may comprise one of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of display. Display 126 may include a High Definition display or an Ultra High Definition display. Display 126 may include a stereoscopic display. It should be noted that although in the example illustrated in FIG. 1, video decoder 124 is described as outputting data to display 126, video decoder 124 may be configured to output video data to various types of devices and/or sub-components thereof. For example, video decoder 124 may be configured to output video data to any communication medium, as described herein. Destination device 120 may include a receive device.

[0056] FIG. 9 is a block diagram illustrating an example of a receiver device that may implement one or more techniques of this disclosure. That is, receiver device 600 may be configured to parse a signal based on the semantics described above. Further, receiver device 600 may be configured to operate according to expected play behavior described herein. Further, receiver device 600 may be configured to perform

translation techniques described herein. Receiver device 600 is an example of a computing device that may be configured to receive data from a communications network and allow a user to access multimedia content, including a virtual reality application. In the example illustrated in FIG. 9, receiver device 600 is configured to receive data via a television network, such as, for example, television service network 404 described above. Further, in the example illustrated in FIG. 9, receiver device 600 is configured to send and receive data via a wide area network. It should be noted that in other examples, receiver device 600 may be configured to simply receive data through a television service network 404. The techniques described herein may be utilized by devices configured to communicate using any and all combinations of communications networks.

[0057] As illustrated in FIG. 9, receiver device 600 includes central processing unit(s) 602, system memory 604, system interface 610, data extractor 612, audio decoder 614, audio output system 616, video decoder 618, display system 620, I/O device(s) 622, and network interface 624. As illustrated in FIG. 9, system memory 604 includes operating system 606 and applications 608. Each of central processing unit(s) 602, system memory 604, system interface 610, data extractor 612, audio decoder 614, audio output system 616, video decoder 618, display system 620, I/O device(s) 622, and network interface 624 may be interconnected (physically, communicatively, and/or operatively) for inter-component communications and may be implemented as any of a variety of suitable circuitry, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic, software, hardware, firmware or any combinations thereof. It should be noted that although receiver device 600 is illustrated as having distinct functional blocks, such an illustration is for descriptive purposes and does not limit receiver device 600 to a particular hardware architecture. Functions of receiver device 600 may be realized using any combination of hardware, firmware and/or software implementations.

[0058] CPU(s) 602 may be configured to implement functionality and/or process instructions for execution in receiver device 600. CPU(s) 602 may include single and/or multi-core central processing units. CPU(s) 602 may be capable of retrieving and processing instructions, code, and/or data structures for implementing one or more of the techniques described herein. Instructions may be stored on a computer readable medium, such as system memory 604.

[0059] System memory 604 may be described as a non-transitory or tangible computer-readable storage medium. In some examples, system memory 604 may provide temporary and/or long-term storage. In some examples, system memory 604 or portions thereof may be described as non-volatile memory and in other examples

portions of system memory 604 may be described as volatile memory. System memory 604 may be configured to store information that may be used by receiver device 600 during operation. System memory 604 may be used to store program instructions for execution by CPU(s) 602 and may be used by programs running on receiver device 600 to temporarily store information during program execution. Further, in the example where receiver device 600 is included as part of a digital video recorder, system memory 604 may be configured to store numerous video files.

[0060] Applications 608 may include applications implemented within or executed by receiver device 600 and may be implemented or contained within, operable by, executed by, and/or be operatively/communicatively coupled to components of receiver device 600. Applications 608 may include instructions that may cause CPU(s) 602 of receiver device 600 to perform particular functions. Applications 608 may include algorithms which are expressed in computer programming statements, such as, for-loops, while-loops, if-statements, do-loops, etc. Applications 608 may be developed using a specified programming language. Examples of programming languages include, Java™, Jini™, C, C++, Objective C, Swift, Perl, Python, PhP, UNIX Shell, Visual Basic, and Visual Basic Script. In the example where receiver device 600 includes a smart television, applications may be developed by a television manufacturer or a broadcaster. As illustrated in FIG. 9, applications 608 may execute in conjunction with operating system 606. That is, operating system 606 may be configured to facilitate the interaction of applications 608 with CPUs(s) 602, and other hardware components of receiver device 600. Operating system 606 may be an operating system designed to be installed on set-top boxes, digital video recorders, televisions, and the like. It should be noted that techniques described herein may be utilized by devices configured to operate using any and all combinations of software architectures.

[0061] System interface 610 may be configured to enable communications between components of receiver device 600. In one example, system interface 610 comprises structures that enable data to be transferred from one peer device to another peer device or to a storage medium. For example, system interface 610 may include a chipset supporting Accelerated Graphics Port (AGP) based protocols, Peripheral Component Interconnect (PCI) bus based protocols, such as, for example, the PCI Express™ (PCIe) bus specification, which is maintained by the Peripheral Component Interconnect Special Interest Group, or any other form of structure that may be used to interconnect peer devices (e.g., proprietary bus protocols).

[0062] As described above, receiver device 600 is configured to receive and, optionally, send data via a television service network. As described above, a television service network may operate according to a telecommunications standard. A telecommu-

nications standard may define communication properties (e.g., protocol layers), such as, for example, physical signaling, addressing, channel access control, packet properties, and data processing. In the example illustrated in FIG. 9, data extractor 612 may be configured to extract video, audio, and data from a signal. A signal may be defined according to, for example, aspects DVB standards, ATSC standards, ISDB standards, DTMB standards, DMB standards, and DOCSIS standards.

[0063] Data extractor 612 may be configured to extract video, audio, and data, from a signal. That is, data extractor 612 may operate in a reciprocal manner to a service distribution engine. Further, data extractor 612 may be configured to parse link layer packets based on any combination of one or more of the structures described above.

[0064] Data packets may be processed by CPU(s) 602, audio decoder 614, and video decoder 618. Audio decoder 614 may be configured to receive and process audio packets. For example, audio decoder 614 may include a combination of hardware and software configured to implement aspects of an audio codec. That is, audio decoder 614 may be configured to receive audio packets and provide audio data to audio output system 616 for rendering. Audio data may be coded using multi-channel formats such as those developed by Dolby and Digital Theater Systems. Audio data may be coded using an audio compression format. Examples of audio compression formats include Motion Picture Experts Group (MPEG) formats, Advanced Audio Coding (AAC) formats, DTS-HD formats, and Dolby Digital (AC-3) formats. Audio output system 616 may be configured to render audio data. For example, audio output system 616 may include an audio processor, a digital-to-analog converter, an amplifier, and a speaker system. A speaker system may include any of a variety of speaker systems, such as headphones, an integrated stereo speaker system, a multi-speaker system, or a surround sound system.

[0065] Video decoder 618 may be configured to receive and process video packets. For example, video decoder 618 may include a combination of hardware and software used to implement aspects of a video codec. In one example, video decoder 618 may be configured to decode video data encoded according to any number of video compression standards, such as ITU-T H.262 or ISO/IEC MPEG-2 Visual, ISO/IEC MPEG-4 Visual, ITU-T H.264 (also known as ISO/IEC MPEG-4 Advanced video Coding (AVC)), and High-Efficiency Video Coding (HEVC). Display system 620 may be configured to retrieve and process video data for display. For example, display system 620 may receive pixel data from video decoder 618 and output data for visual presentation. Further, display system 620 may be configured to output graphics in conjunction with video data, e.g., graphical user interfaces. Display system 620 may comprise one of a variety of display devices such as a liquid crystal display (LCD), a plasma display, an organic light emitting diode (OLED) display, or another type of

display device capable of presenting video data to a user. A display device may be configured to display standard definition content, high definition content, or ultra-high definition content.

[0066] I/O device(s) 622 may be configured to receive input and provide output during operation of receiver device 600. That is, I/O device(s) 622 may enable a user to select multimedia content to be rendered. Input may be generated from an input device, such as, for example, a push-button remote control, a device including a touch-sensitive screen, a motion-based input device, an audio-based input device, or any other type of device configured to receive user input. I/O device(s) 622 may be operatively coupled to receiver device 600 using a standardized communication protocol, such as for example, Universal Serial Bus protocol (USB), Bluetooth, ZigBee or a proprietary communications protocol, such as, for example, a proprietary infrared communications protocol.

[0067] Network interface 624 may be configured to enable receiver device 600 to send and receive data via a local area network and/or a wide area network. Network interface 624 may include a network interface card, such as an Ethernet card, an optical transceiver, a radio frequency transceiver, or any other type of device configured to send and receive information. Network interface 624 may be configured to perform physical signaling, addressing, and channel access control according to the physical and Media Access Control (MAC) layers utilized in a network. Receiver device 600 may be configured to parse a signal generated according to any of the techniques described above with respect to FIG. 8. In this manner, receiver device 600 represents an example of a device configured to decapsulate a timed metadata track in a particular representation associated with a sub-picture composition and parse an association identifier of a timed metadata track, where the association identifier includes a value corresponding to omnidirectional media carried by a media track.

[0068] In one or more examples, the functions described may be implemented in hardware, software, firmware, or any combination thereof. If implemented in software, the functions may be stored on or transmitted over as one or more instructions or code on a computer-readable medium and executed by a hardware-based processing unit. Computer-readable media may include computer-readable storage media, which corresponds to a tangible medium such as data storage media, or communication media including any medium that facilitates transfer of a computer program from one place to another, e.g., according to a communication protocol. In this manner, computer-readable media generally may correspond to (1) tangible computer-readable storage media which is non-transitory or (2)a communication medium such as a signal or carrier wave. Data storage media may be any available media that can be accessed by one or more computers or one or more processors to retrieve instructions, code and/or

data structures for implementation of the techniques described in this disclosure. A computer program product may include a computer-readable medium.

[0069] By way of example, and not limitation, such computer-readable storage media can comprise RAM, ROM, EEPROM, CD-ROM or other optical disk storage, magnetic disk storage, or other magnetic storage devices, flash memory, or any other medium that can be used to store desired program code in the form of instructions or data structures and that can be accessed by a computer. Also, any connection is properly termed a computer-readable medium. For example, if instructions are transmitted from a website, server, or other remote source using a coaxial cable, fiber optic cable, twisted pair, digital subscriber line (DSL), or wireless technologies such as infrared, radio, and microwave, then the coaxial cable, fiber optic cable, twisted pair, DSL, or wireless technologies such as infrared, radio, and microwave are included in the definition of medium. It should be understood, however, that computer-readable storage media and data storage media do not include connections, carrier waves, signals, or other transitory media, but are instead directed to non-transitory, tangible storage media. Disk and disc, as used herein, includes compact disc (CD), laser disc, optical disc, digital versatile disc (DVD), floppy disk and Blu-ray disc where disks usually reproduce data magnetically, while discs reproduce data optically with lasers. Combinations of the above should also be included within the scope of computer-readable media.

[0070] Instructions may be executed by one or more processors, such as one or more digital signal processors (DSPs), general purpose microprocessors, application specific integrated circuits (ASICs), field programmable logic arrays (FPGAs), or other equivalent integrated or discrete logic circuitry. Accordingly, the term “processor,” as used herein may refer to any of the foregoing structure or any other structure suitable for implementation of the techniques described herein. In addition, in some aspects, the functionality described herein may be provided within dedicated hardware and/or software modules configured for encoding and decoding, or incorporated in a combined codec. Also, the techniques could be fully implemented in one or more circuits or logic elements.

[0071] The techniques of this disclosure may be implemented in a wide variety of devices or apparatuses, including a wireless handset, an integrated circuit (IC) or a set of ICs (e.g., a chip set). Various components, modules, or units are described in this disclosure to emphasize functional aspects of devices configured to perform the disclosed techniques, but do not necessarily require realization by different hardware units. Rather, as described above, various units may be combined in a codec hardware unit or provided by a collection of interoperative hardware units, including one or more processors as described above, in conjunction with suitable software and/or firmware.

[0072] Moreover, each functional block or various features of the base station device and the terminal device used in each of the aforementioned embodiments may be implemented or executed by a circuitry, which is typically an integrated circuit or a plurality of integrated circuits. The circuitry designed to execute the functions described in the present specification may comprise a general-purpose processor, a digital signal processor (DSP), an application specific or general application integrated circuit (ASIC), a field programmable gate array (FPGA), or other programmable logic devices, discrete gates or transistor logic, or a discrete hardware component, or a combination thereof. The general-purpose processor may be a microprocessor, or alternatively, the processor may be a conventional processor, a controller, a microcontroller or a state machine. The general-purpose processor or each circuit described above may be configured by a digital circuit or may be configured by an analogue circuit. Further, when a technology of making into an integrated circuit superseding integrated circuits at the present time appears due to advancement of a semiconductor technology, the integrated circuit by this technology is also able to be used.

[0073] Various examples have been described. These and other examples are within the scope of the following claims.

[0074] <Cross Reference>

This Nonprovisional application claims priority under 35 U.S.C. § 119 on provisional Application No. 62/725,236 on August 30, 2018, No. 62/742,904 on October 8, 2018 No. 62/785,436 on December 27, 2018,No. 62/815,229 on March 7, 2019, the entire contents of which are hereby incorporated by reference.

Claims

[Claim 1] A method of signaling information associated with an omnidirectional video, the method comprising:
encapsulating a timed metadata track associated with a particular representation; and
signaling an association descriptor of the particular representation of the timed metadata track,
wherein the association descriptor includes (i) a string in an association element of a type concerning a sub-picture composition identifier value and (ii) a constant of the association element.

[Claim 2] The method of claim 1, wherein the association descriptor is present as a child element of the particular representation.

[Claim 3] The method of claim, 1 wherein the constant included in the association descriptor indicates a value of an association attribute of the association element.

[Claim 4] A method of determining information associated with an omnidirectional video, the method comprising:
decapsulating a timed metadata track associated with a particular representation; and
receiving an association descriptor of the particular representation of the timed metadata track,
wherein the association descriptor includes (i) a string in an association element of a type concerning a sub-picture composition identifier value and (ii) a constant of the association element.

[Claim 5] A device comprising one or more processors configured to perform any and all combinations of the steps of claims 1-4.

[Claim 6] An apparatus comprising means for performing any and all combinations of the steps of claims 1-4.

[Claim 7] A non-transitory computer-readable storage medium comprising instructions stored thereon that, when executed, cause one or more processors of a device to perform any and all combinations of the steps of claims 1-4.

[Fig. 1]

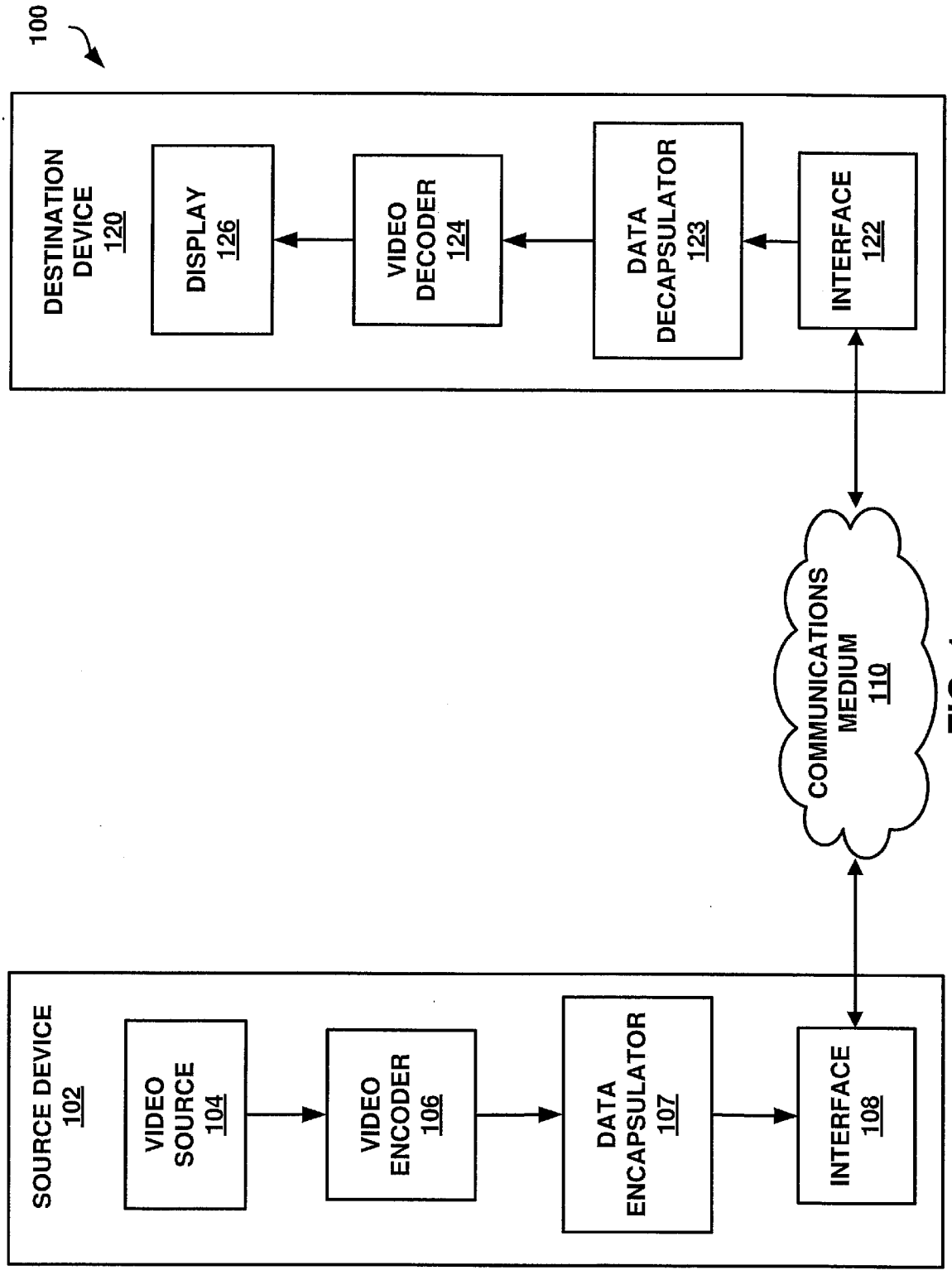

100
SOURCE DEVICE 102
VIDEO SOURCE 104
VIDEO ENCODER 106
DATA ENCAPSULATOR 107
INTERFACE 108
COMMUNICATIONS MEDIUM 110
DESTINATION DEVICE 120
DISPLAY 126
VIDEO DECODER 124
DATA DECAPSULATOR 123
INTERFACE 122

FIG. 1

[Fig. 2A]

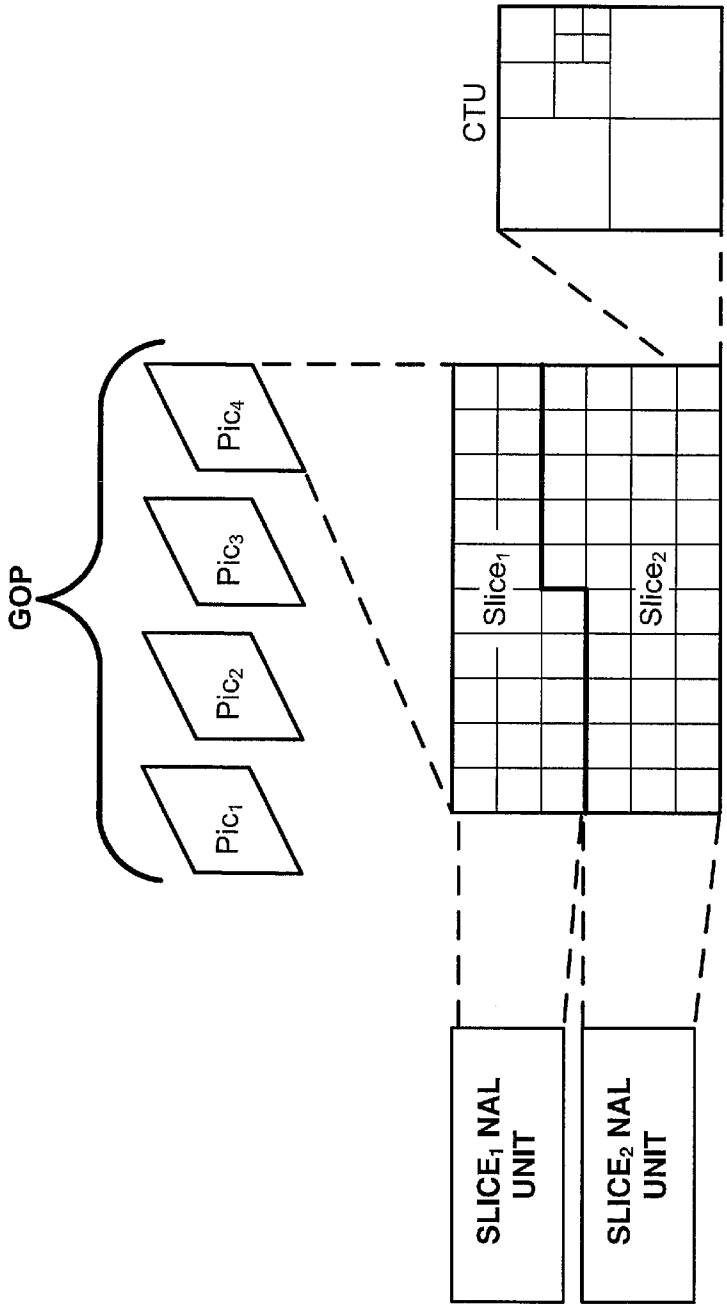

CTU
GOP
Pic4
Pic3
Pic2
Pic1
Slice1
Slice2
SLICE1 NAL UNIT
SLICE2 NAL UNIT

FIG. 2A

[Fig. 2B]

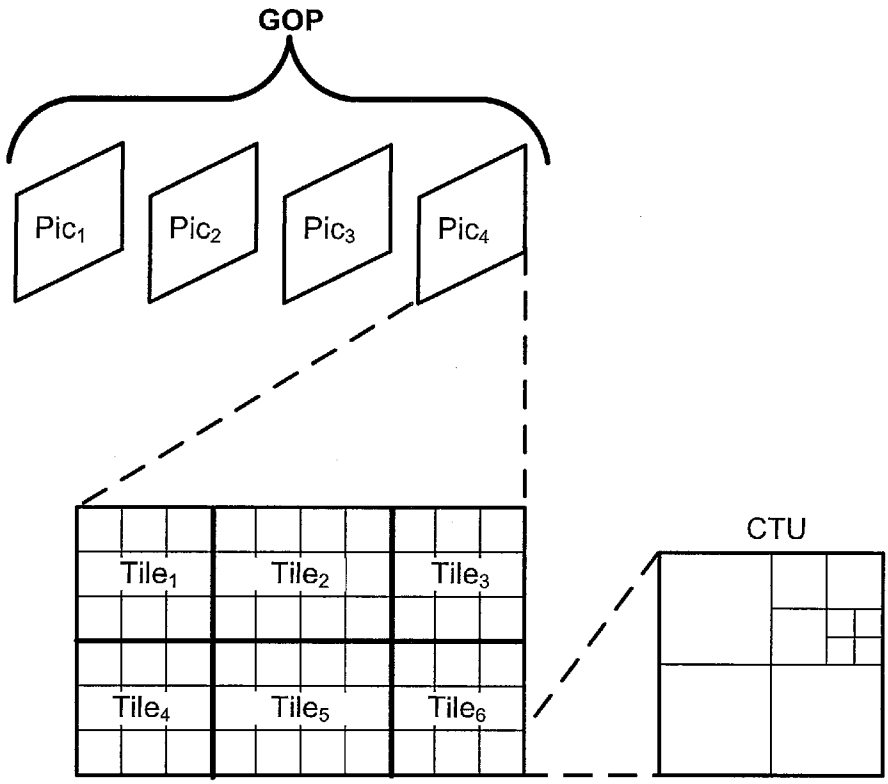

GOP
Pic1
Pic2
Pic3
Pic4
Tile1
Tile2
Tile3
Tile4
Tile5
Tile6
CTU

FIG. 2B

[Fig. 3]

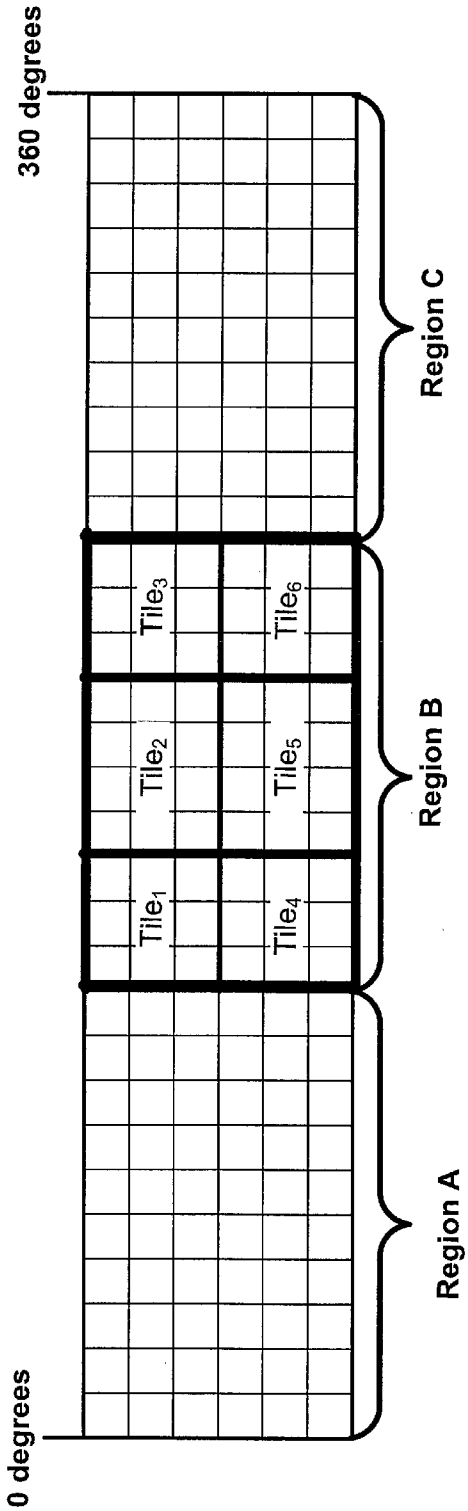

360 degrees
0 degrees
Region C
Region B
Region A
Tile1
Tile2
Tile3
Tile4
Tile5
Tile6

FIG. 3

[Fig. 4]

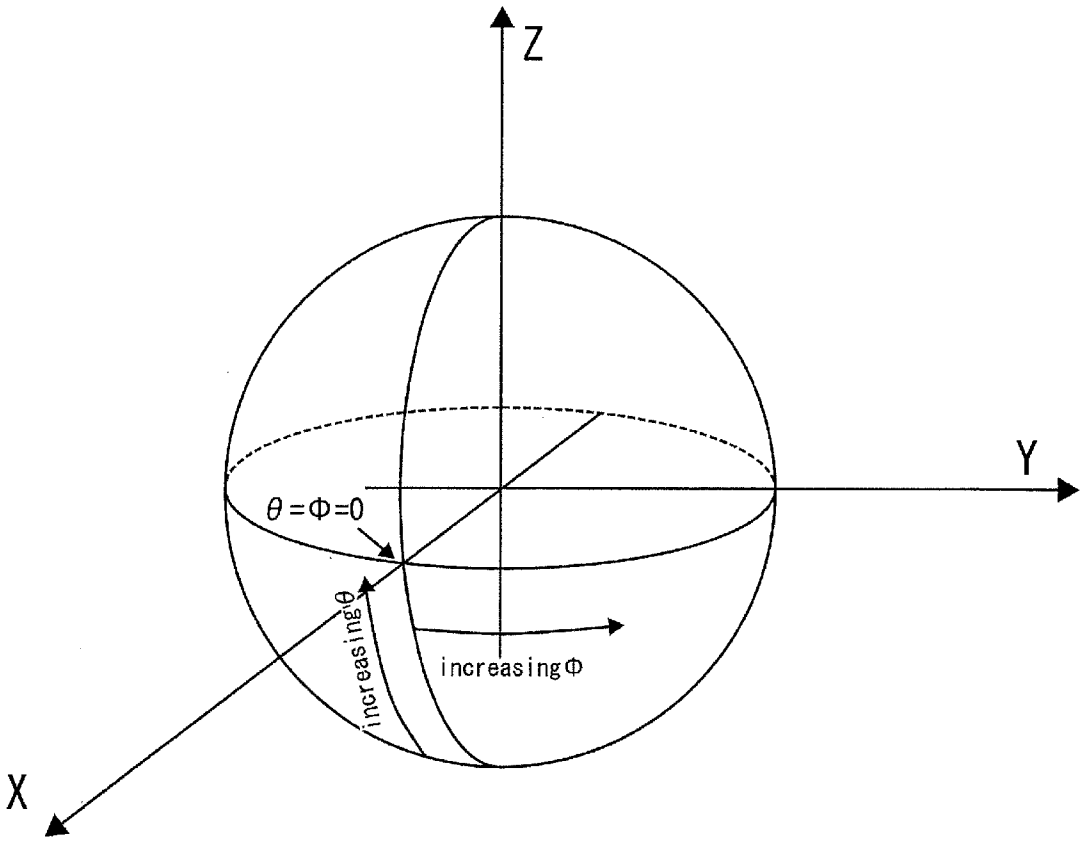


FIG. 4

[Fig. 5A]

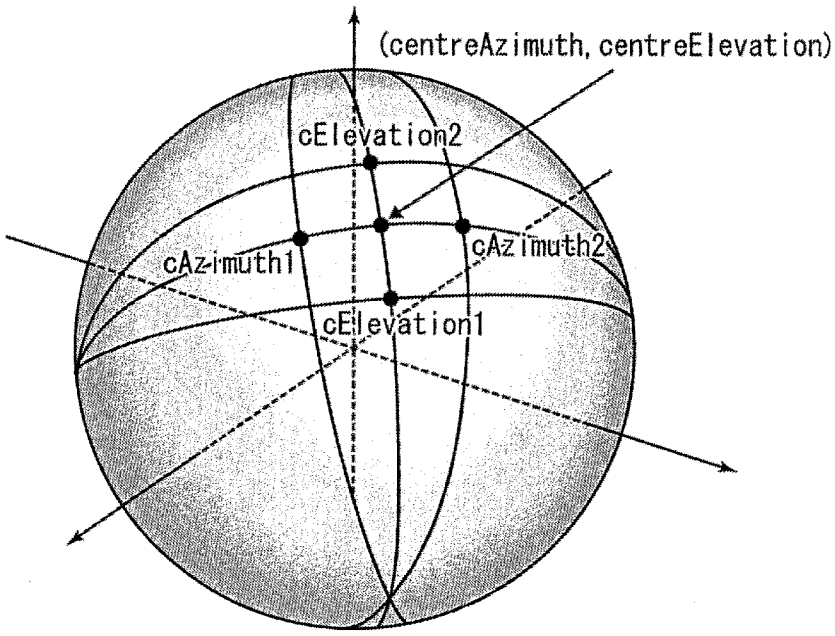


FIG. 5A

[Fig. 5B]

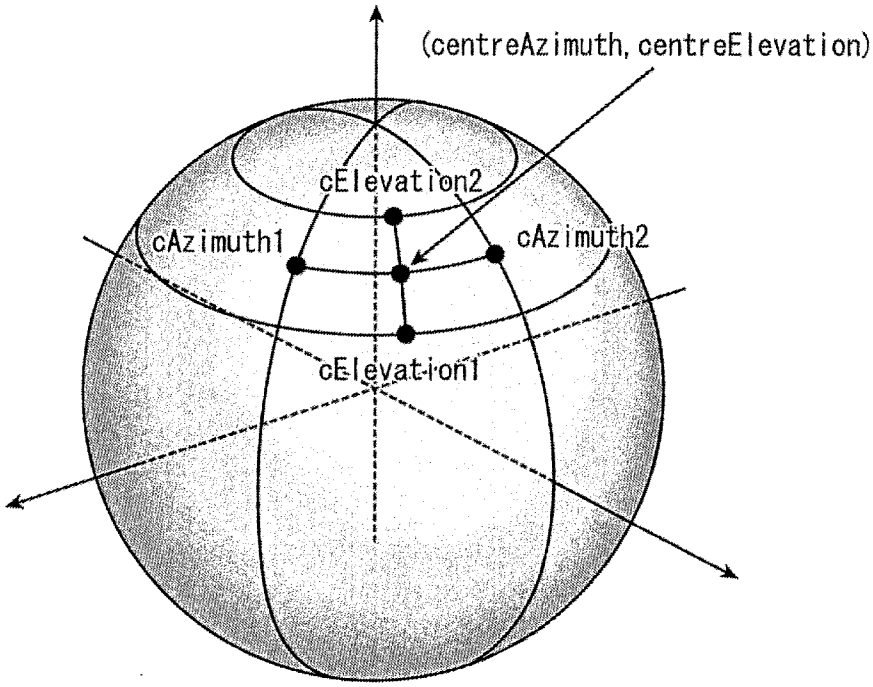

(centreAzimuth, centreElevation)
cElevation2
cAzimuth1
cAzimuth2
cElevation1

FIG. 5B

[Fig. 6]

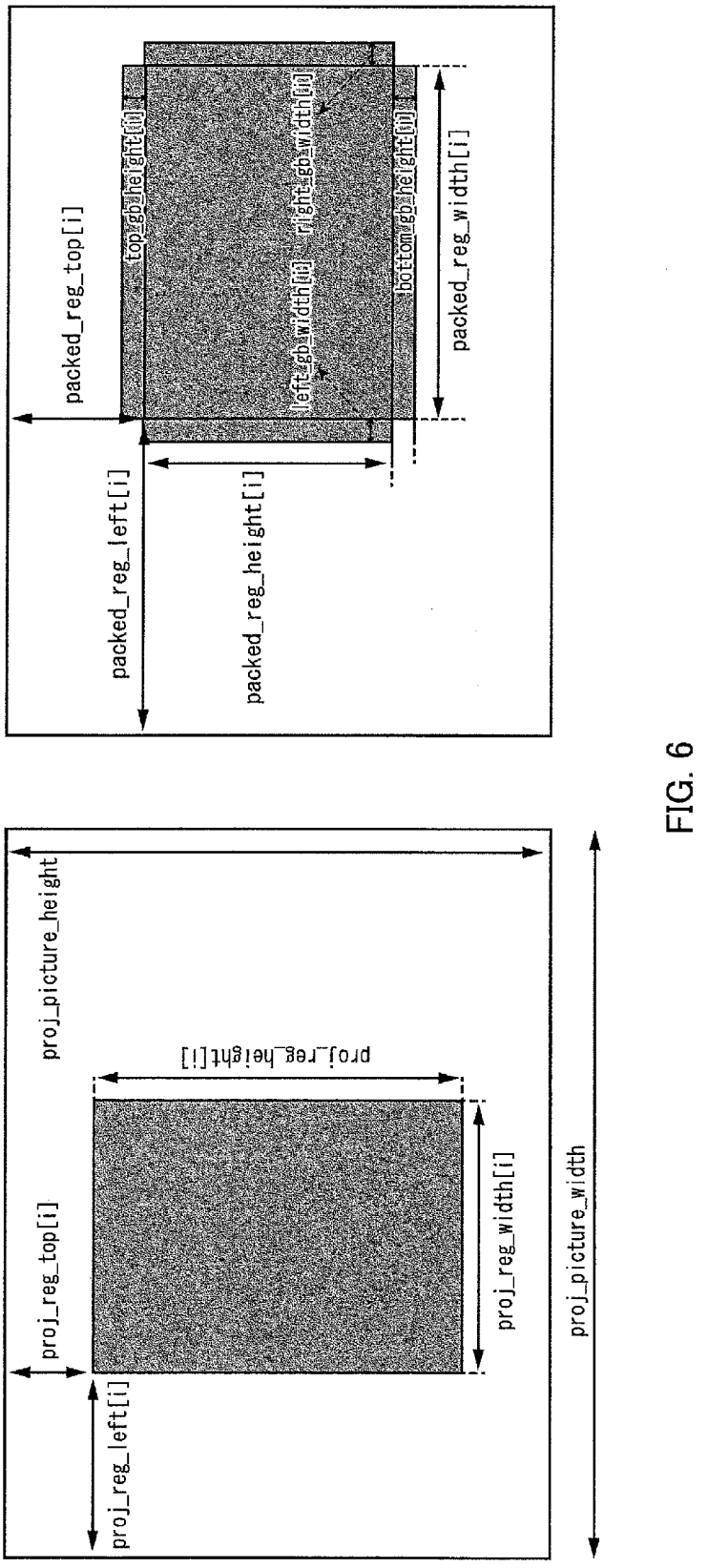

packed_reg_top[i]
top_gb_height[i]
left_gb_width[i]
right_gb_width[i]
bottom_gb_height[i]
packed_reg_width[i]
packed_reg_left[i]
packed_reg_height[i]
proj_picture_height
proj_reg_height[i]
proj_reg_top[i]
proj_reg_width[i]
proj_reg_left[i]
proj_picture_width

FIG. 6

[Fig. 7]

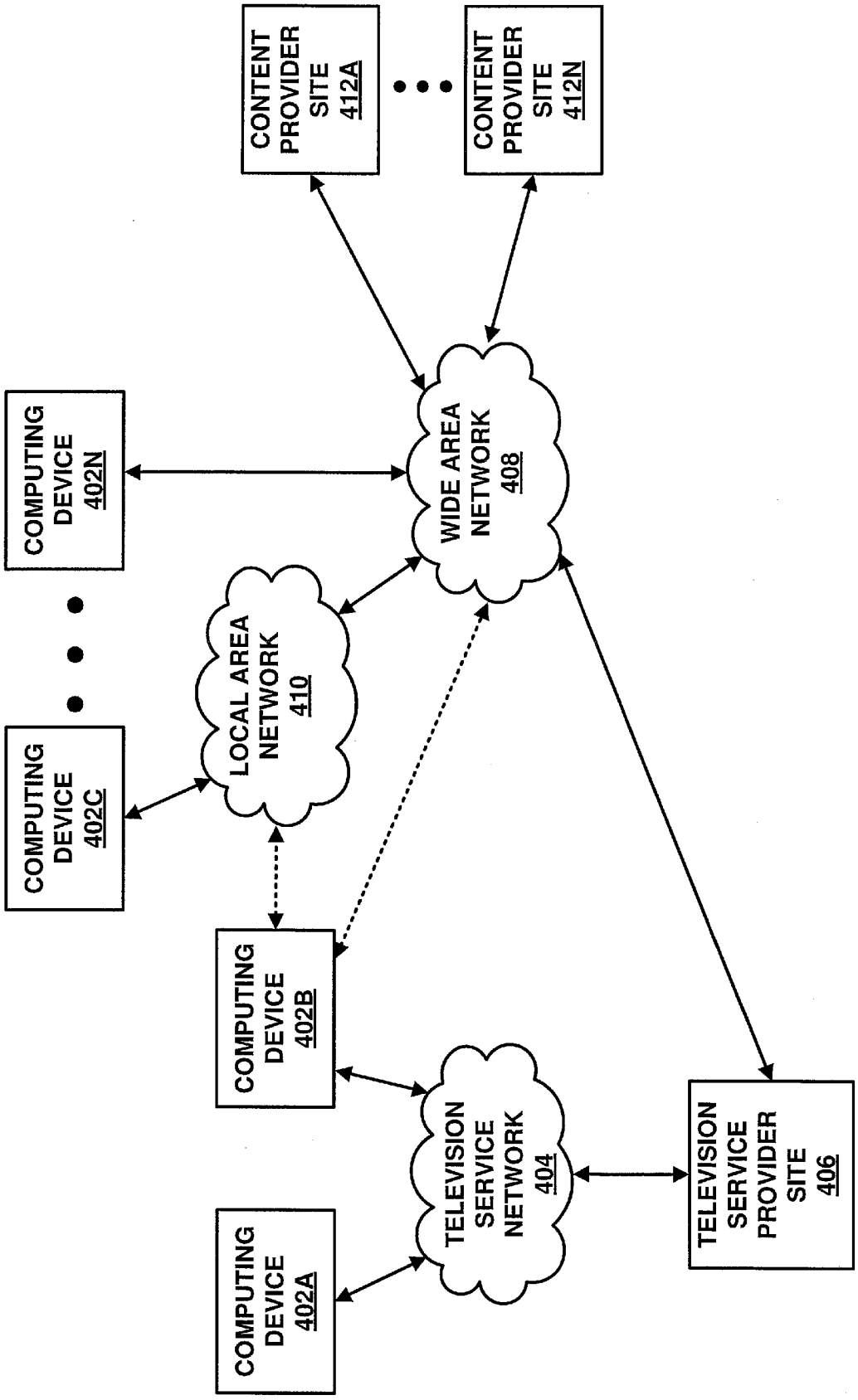

CONTENT PROVIDER SITE 412A
CONTENT PROVIDER SITE 412N
WIDE AREA NETWORK 408
COMPUTING DEVICE 402N
LOCAL AREA NETWORK 410
COMPUTING DEVICE 402C
COMPUTING DEVICE 402B
TELEVISION SERVICE NETWORK 404
TELEVISION SERVICE PROVIDER SITE 406
COMPUTING DEVICE 402A

FIG. 7

[Fig. 8]

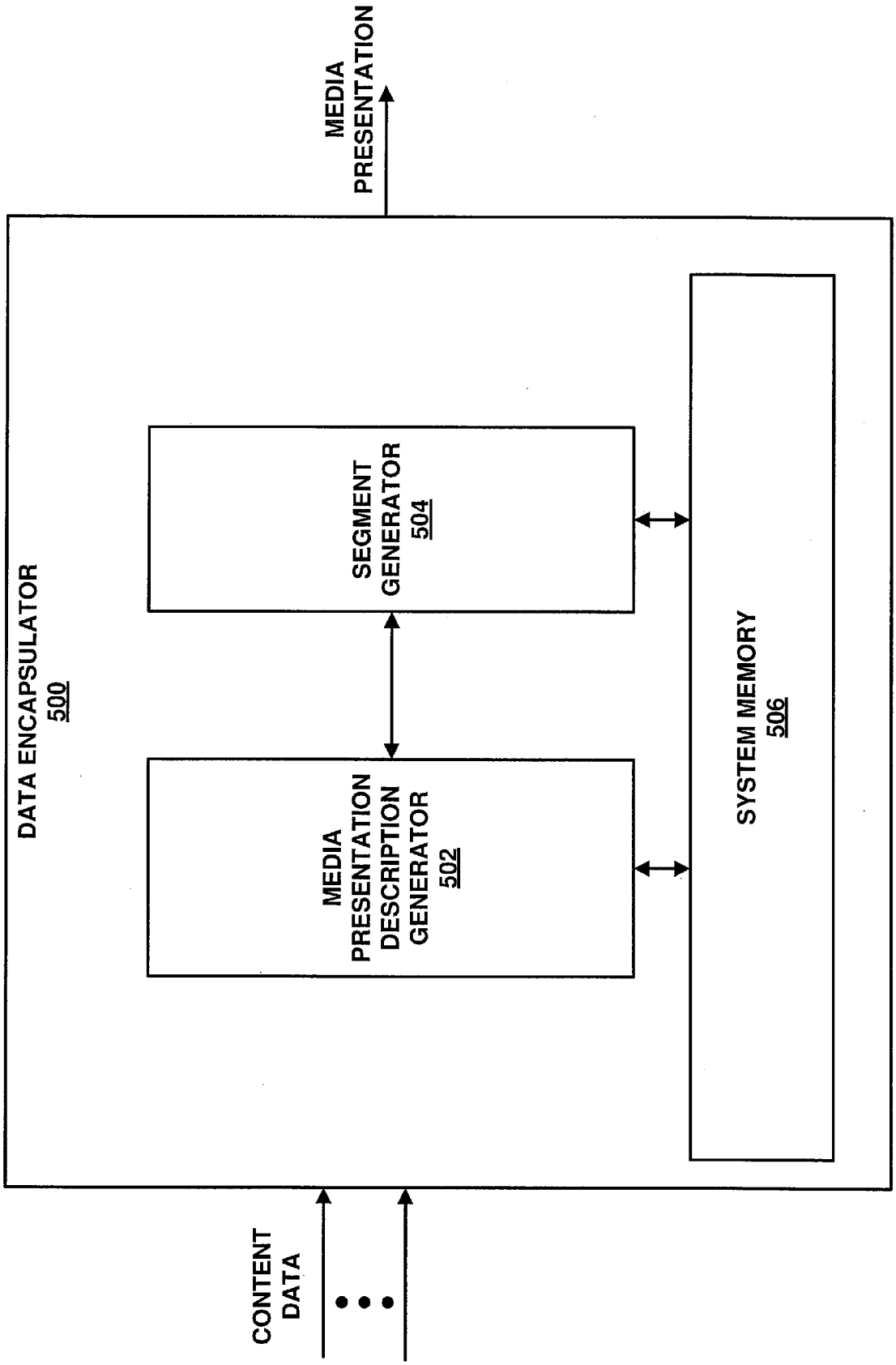

MEDIA PRESENTATION
DATA ENCAPSULATOR
500
SEGMENT GENERATOR
504
MEDIA PRESENTATION DESCRIPTION GENERATOR
502
SYSTEM MEMORY
506
CONTENT DATA

FIG. 8

[Fig. 9]

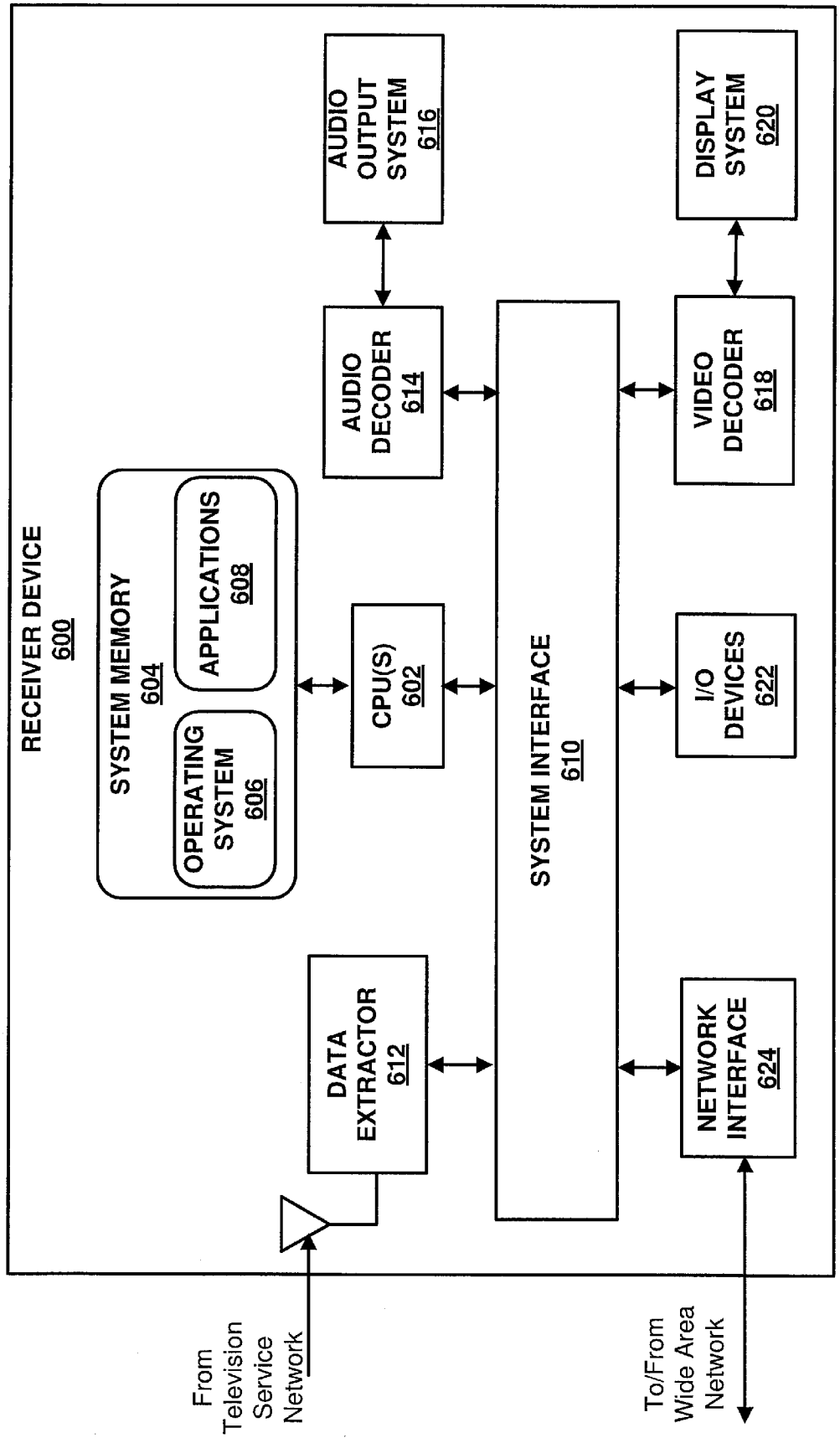

RECEIVER DEVICE
600
SYSTEM MEMORY
604
OPERATING SYSTEM
606
APPLICATIONS
608
CPU(S)
602
SYSTEM INTERFACE
610
AUDIO DECODER
614
AUDIO OUTPUT SYSTEM
616
VIDEO DECODER
618
DISPLAY SYSTEM
620
I/O DEVICES
622
DATA EXTRACTOR
612
NETWORK INTERFACE
624
From Television Service Network
To/From Wide Area Network

FIG. 9

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/034011

A. CLASSIFICATION OF SUBJECT MATTER

Int.Cl. H04N21/235(2011.01)i

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)

Int.Cl. H04N21/235

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

Published examined utility model applications of Japan 1922-1996
Published unexamined utility model applications of Japan 1971-2019
Registered utility model specifications of Japan 1996-2019
Published registered utility model applications of Japan 1994-2019

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

IEEE Xplore

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2018/0061002 A1 (LEE et al.) 2018.03.01, Paragraphs [0006],[0007],[0012],[0475], [0477]-[0480],[0483],[0484],[0486],[0488], [0489]; Figures 22 and 23. & JP 2019-525675 A & WO 2018/038523 A1 & CN 109691094 A & EP 3503547 A1 & BR 112019003605 A2 & KR 10-2019-075041 A	1-7
A	US 2018/0199042 A1 (WANG et al.) 2018.07.12, Paragraphs [0106],[0129]-[0131]; Figure 2A. & WO 2018/127123 A1 & TW 201832555 A & CN 110169076 A	1-7

☑ 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
08.11.2019	19.11.2019

Name and mailing address of the ISA/JP	Authorized officer
Japan Patent Office 3-4-3, Kasumigaseki, Chiyoda-ku, Tokyo 100-8915, Japan	KUBO, Mitsuhiro 5C 9189 Telephone No. +81-3-3581-1101 Ext. 3539

INTERNATIONAL SEARCH REPORT

International application No.
PCT/JP2019/034011

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	EP 3349467 A1 (NOKIA TECHNOLOGIES OY) 2018.07.18, Paragraphs [0047],[0185],[0188],[0211],[0246], [0280],[0288]; Figures 6A,7,8 and 10b. (Family: None)	1-7
A	LG ELECTRONICS, et al., "[OMAF] Proposed solution for sub-picture tracks", [online], ISO/IEC JTC1/SC29/WG11, MPEG2018/M42492, 2018.04, Pages 1-8, [retrieved on 2019.11.07], Retrieved from the Internet: <URL: http://mcsl.skku.edu/wordpress/wp-content/uploads/2018/05/m42492_OMAF_sub-picture_composition.docx>. Especially, Page 1, lines 1-7; Page 2, lines 6-8 and 27-35; Page 3, lines 1-3; Page 4, lines 9-13 and 17-33; Page 5, lines 1-10, 13-14 and 17-18; Fourth and eighth boxes in right-hand column in page 7.	1-7
A	LG ELECTRONICS INC., "pCR TS 26.118: Media Profiles for Video", [online], 2018.04.03, 3GPP TSG SA4 WG4 Meeting#98, S4-180387, Pages 1-4, [retrieved on 2019.11.07], Retrieved from the Internet: <URL: https://www.3gpp.org/ftp/tsg_sa/WG4_CODEC/TSGS4_98/Docs/S4-180387.zip>. Especially, Final paragraph of section 5.3.1.1.2 in page 2.	1-7
A	YE-KUI WANG, "A summary of OMAF DIS to SoDIS updates", [online], 2017.10.04, Pages 1-20, [retrieved on 2019.11.07], Retrieved from the Internet: <URL: https://www.3gpp.org/ftp/TSG_SA/WG4_CODEC/TSGS4_95/Docs/S4-170798.zip>. Especially,	1-7

C (Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
	Page 9, lines 1-5; Page 10, lines 1-4; Page 11, lines 1-4; Page 14, lines 1-10; Page 18, lines 1-5. [NOTE] This document is contained in the file "OMAF.pptx" which is included in the downloaded zip-file "S4-170798.zip".	