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

(19) World Intellectual Property Organization
International Bureau

(43) International Publication Date
19 January 2012 (19.01.2012)

PCT

(10) International Publication Number
WO 2012/009454 A1

(51) International Patent Classification:
H04N 7/26 (2006.01) *H04N 13/00* (2006.01)
H04N 7/46 (2006.01) *H04N 21/2343* (2011.01)
H04N 7/50 (2006.01) *H04N 21/44* (2011.01)

(21) International Application Number: PCT/US2011/043885

(22) International Filing Date: 13 July 2011 (13.07.2011)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:

61/363,884	13 July 2010 (13.07.2010)	US
61/366,436	21 July 2010 (21.07.2010)	US
61/374,222	16 August 2010 (16.08.2010)	US
61/433,110	14 January 2011 (14.01.2011)	US
13/031,067	18 February 2011 (18.02.2011)	US

(71) Applicant *(for all designated States except US)*: **QUALCOMM INCORPORATED** [US/US]; ATTN: International IP Administration, 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(72) Inventors; and

(75) Inventors/Applicants *(for US only)*: **CHEN, Ying** [CN/US]; 5775 Morehouse Drive, San Diego, California 92121-1714 (US). **KARCZEWICZ, Marta** [US/US]; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(74) Agents: **RICKENBRODE, John** et al.; 5775 Morehouse Drive, San Diego, California 92121-1714 (US).

(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, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, DK, DM, DO, DZ, EC, EE, EG, ES, FI, GB, GD, GE, GH, GM, GT, HN, HR, HU, ID, IL, IN, IS, JP, KE, KG, KM, KN, KP, KR, KZ, LA, LC, LK, LR, LS, LT, LU, LY, MA, MD, ME, MG, MK, MN, MW, MX, MY, MZ, NA, NG, NI, NO, NZ, OM, PE, PG, PH, PL, PT, RO, RS, RU, 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, SD, SL, SZ, TZ, UG, ZM, ZW), Eurasian (AM, AZ, BY, KG, KZ, MD, RU, TJ, TM), European (AL, AT, BE, BG, CH, CY, CZ, DE, DK, EE, ES, FI, FR, GB, GR, HR, HU, IE, IS, IT, LT, LU, LV, MC, MK, MT, NL, NO, PL, PT, RO, RS, SE, SI, SK, SM, TR), OAPI (BF, BJ, CF, CG, CI, CM, GA, GN, GQ, GW, 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))*

[Continued on next page]

(54) Title: VIDEO SWITCHING FOR STREAMING VIDEO DATA

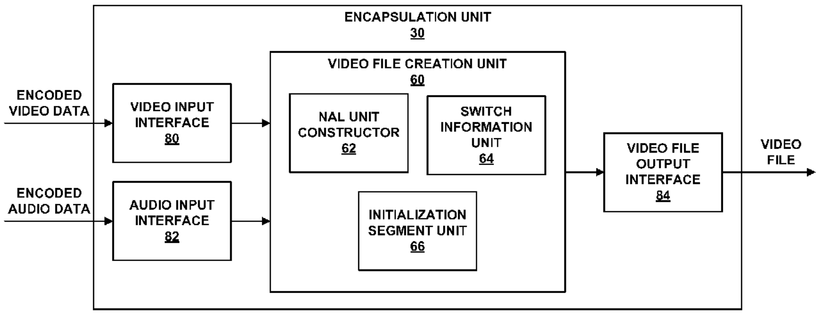

FIG. 2

(57) **Abstract**: In one example, an apparatus includes a processor configured to provide information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence, and to provide information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, in response to a first request from the client device to retrieve data from the first representation. In this manner, the client device may use the information to perform seamless switching between the representations.

Published:

- *with international search report (Art. 21(3))*
- *before the expiration of the time limit for amending the claims and to be republished in the event of receipt of amendments (Rule 48.2(h))*

VIDEO SWITCHING FOR STREAMING VIDEO DATA

[0001] This application claims the benefit of U.S. Provisional Application No. 61/363,884, filed July 13, 2010, U.S. Provisional Application No. 61/366,436, filed July 21, 2010, U.S. Provisional Application No. 61/374,222, filed August 16, 2010, and U.S. Provisional Application No. 61/433,110 filed January 14, 2011, each of which is hereby incorporated by reference in its entirety.

TECHNICAL FIELD

[0002] This disclosure relates to storage and transport of encoded video data.

BACKGROUND

[0003] Digital video capabilities can be incorporated into a wide range of devices, including digital televisions, digital direct broadcast systems, wireless broadcast systems, personal digital assistants (PDAs), laptop or desktop computers, digital cameras, digital recording devices, digital media players, video gaming devices, video game consoles, cellular or satellite radio telephones, video teleconferencing devices, and the like. Digital video devices implement video compression techniques, such as those described in the standards defined by MPEG-2, MPEG-4, ITU-T H.263 or ITU-T H.264/MPEG-4, Part 10, Advanced Video Coding (AVC), and extensions of such standards, to transmit and receive digital video information more efficiently.

[0004] Video compression techniques perform spatial prediction and/or temporal prediction to reduce or remove redundancy inherent in video sequences. For block-based video coding, a video frame or slice may be partitioned into macroblocks. Each macroblock can be further partitioned. Macroblocks in an intra-coded (I) frame or slice are encoded using spatial prediction with respect to neighboring macroblocks. Macroblocks in an inter-coded (P or B) frame or slice may use spatial prediction with respect to neighboring macroblocks in the same frame or slice or temporal prediction with respect to other reference frames.

[0005] After video data has been encoded, the video data may be packetized for transmission or storage. The video data may be assembled into a video file conforming to any of a variety of standards, such as the International Organization for Standardization (ISO) base media file format and extensions thereof, such as AVC.

[0006] Efforts have been made to develop new video coding standards based on H.264/AVC. One such standard is the scalable video coding (SVC) standard, which is the scalable extension to H.264/AVC. Another standard is the multi-view video coding (MVC), which has become the multiview extension to H.264/AVC. A joint draft of MVC is in described in JVT-AB204, "Joint Draft 8.0 on Multiview Video Coding," 28th JVT meeting, Hannover, Germany, July 2008, available at http://wftp3.itu.int/av-arch/jvt-site/2008_07_Hannover/JVT-AB204.zip. A version of the AVC standard is described in JVT-AD007, "Editors' draft revision to ITU-T Rec. H.264 | ISO/IEC 14496-10 Advanced Video Coding – in preparation for ITU-T SG 16 AAP Consent (in integrated form)," 30th JVT meeting, Geneva, CH, Feb. 2009," available from http://wftp3.itu.int/av-arch/jvt-site/2009_01_Geneva/JVT-AD007.zip. This document provides an integration of the SVC and MVC standards with the AVC specification.

SUMMARY

[0007] In general, this disclosure describes techniques for switching between representations of multimedia content while streaming the representations using a network streaming protocol. The same multimedia content may be encoded in a variety of ways, resulting in various representations of the multimedia content. The techniques of this disclosure may improve streaming of data for the multimedia content (e.g., according to a network streaming protocol such as HTTP version 1.1, also referred to as HTTP streaming) by improving a client device's ability to switch between representations without interrupting video playback at the client device.

[0008] In accordance with the techniques of this disclosure, a server device may signal (that is, provide in response to a request) a delay value representing decoding delay and picture reordering delay. Decoding delay may represent an amount of time required to decode one or more encoded video samples of a representation. Picture reordering delay may represent an amount of time required to reorder pictures for display purposes, as the pictures are not necessarily displayed in the same order that they are decoded. The client device may use the delay value to determine a point at which the client device may switch from one representation to another. That is, the client device may determine a point at which the client device stops requesting data of a first representation and instead begins requesting data for a second, different representation.

Taking account of the signaled delay value, the client may determine this point such that playback (following decoding and reordering) is not interrupted by the switch between representations.

[0009] In one example, a method of sending video data includes providing information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence; providing information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, in response to a first request from the client device to retrieve data from the first representation, sending the requested data from the first representation to the client device, and, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, sending the requested data from the second representation to the client device.

[0010] In another example, an apparatus for sending video data includes a processor configured to provide information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence, provide information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, in response to a first request from the client device to retrieve data from the first representation, send the requested data from the first representation to the client device, and, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, send the requested data from the second representation to the client device.

[0011] In another example, an apparatus for sending video data includes means for providing information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence, means for providing information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, means for, in response to a first request from the client device to retrieve data from the first representation, sending the requested data from the first representation to the client device, and means for, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, sending the requested data from the second representation to the client device.

[0012] In another example, a computer program product includes a computer-readable storage medium comprises instructions that, when executed, cause a processor of a source device for sending video data to provide information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence, provide information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, in response to a first request from the client device to retrieve data from the first representation, send the requested data from the first representation to the client device, and, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, send the requested data from the second representation to the client device.

[0013] In another example, a method of receiving video data includes receiving a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence, requesting data from the first representation of a video sequence in accordance with a streaming network protocol, in response to determining to switch from the first representation to the second representation, buffering an amount of data from the first representation based on the delay value, and, after buffering the amount of data from the first representation, requesting data from a second representation of the video sequence in accordance with the streaming network protocol.

[0014] In another example, an apparatus for receiving video data includes a processor configured to receive a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence, request data from the first representation of a video sequence in accordance with a streaming network protocol, in response to determining to switch from the first representation to the second representation, buffer an amount of data from the first representation based on the delay value, and, after buffering the amount of data from the first representation, request data from a second representation of the video sequence in accordance with the streaming network protocol.

[0015] In another example, an apparatus for receiving video data includes means for receiving a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence, means for requesting data from the first representation of a video sequence in accordance with a streaming network protocol, means for in response to determining to switch from the first representation to the second representation, buffering an amount of data from the first representation based on the delay value, and means for requesting, after buffering the amount of data from the first representation, data from a second representation of the video sequence in accordance with the streaming network protocol.

[0016] In another example, a computer program product includes a computer-readable storage medium comprises instructions that cause a processor of a destination device for receiving video data to receive a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence, request data from the first representation of a video sequence in accordance with a streaming network protocol, in response to determining to switch from the first representation to the second representation, buffer an amount of data from the first representation based on the delay value, and after buffering the amount of data from the first representation, request data from a second representation of the video sequence in accordance with the streaming network protocol.

[0017] 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.

BRIEF DESCRIPTION OF DRAWINGS

[0018] FIG. 1 is a block diagram illustrating an example system in which an audio/video (A/V) source device transfers audio and video data to an A/V destination device.

[0019] FIG. 2 is a block diagram illustrating components of an example encapsulation unit.

[0020] FIG. 3 is a conceptual diagram illustrating a sequence of coded video frames.

[0021] FIG. 4 is a conceptual diagram illustrating switchable representations.

[0022] FIG. 5A is a conceptual diagram illustrating an example of two representations that are timely aligned.

[0023] FIG. 5B is a conceptual diagram illustrating an example of two representations that are not timely aligned.

[0024] FIG. 6 is a flowchart illustrating an example method for performing seamless switching between two switchable representations.

[0025] FIG. 7 is a flowchart illustrating an example method for a server device to provide information describing switchable representations to a client device.

[0026] FIG. 8 is a flowchart illustrating an example method for a client device to switch from retrieving data of a first representation from a server device to retrieving data of a second representation from the server device.

DETAILED DESCRIPTION

[0027] In general, this disclosure describes techniques for supporting switching between representations of multimedia content in a network streaming context. There may be natural delay when switching between representations, caused by a variety of factors. As explained in greater detail below, the techniques of this disclosure include signaling (e.g., a server device providing data in response to a request from a client device) information indicative of a delay to account for when switching from one representation of multimedia content to another representation of the multimedia content. In this manner, the client device may buffer a sufficient amount of data from the first representation to allow for decoding of the data before receiving data from the second representation and beginning to decode the data from the second representation. Accordingly, the client device may be capable of switching between representations without interrupting playback of the multimedia content. In this manner, the techniques of this disclosure may improve a user's experience when viewing the content on the client device.

[0028] The signaled delay may represent both decoding delay and picture reordering delay. In particular, the signaled delay value may comprise a combination of a maximum decoding delay and a maximum reordering delay for switching from one representation to another. There may be different decoding delays associated with a sequence of pictures, e.g., based on an encoding mode, a bitrate, a resolution, or other factors of each frame. The signaled delay value may include a maximum decoding delay of any access unit of any representation, or any access unit of the two representations involved in the switch.

[0029] The picture reordering delay may correspond to a delay associated with reordering pictures from decoding to display order. In the example of H.264/AVC, a bitstream may include various types of coded pictures, e.g., intra-coded pictures (e.g., I-pictures) and inter-coded pictures (e.g., P-pictures and B-pictures). Data for an I-picture are generally coded relative only to other data of the I-picture, and thus, are intra-picture coded. Data for P- and B-pictures may be coded relative to one or more other previously coded pictures. A P-picture, for example, may be coded relative to one or more pictures preceding the P-picture in display order. A B-picture, as another example, may be coded relative to one or more pictures preceding the B-picture in display order and one or more pictures following the B-picture in display order. Pictures used for reference may be encoded and placed earlier in the bitstream than the pictures making reference to those pictures.

[0030] As one example, a B-picture may be encoded relative to both an I-picture and a P-picture, and the P-picture may be encoded relative to the I-picture. Of the three pictures, the P-picture may have the earliest display time, the I-picture may have the latest display time, and the B-picture may have a display time between that of the P-picture and the I-picture. Due to the prediction structure of these pictures, a video encoder may place the I-picture earliest of the three pictures in the bitstream, place the B-picture latest of the three pictures in the bitstream, and place the P-picture between the I-picture and the B-picture in the bitstream. Accordingly, a decoder may decode the bitstream in order, that is, decode the I-picture, then the P-picture, and then the B-picture. However, the decoder may reorder the pictures into display order such that the

P-picture is displayed before the B-picture, and the B-picture is displayed before the I-picture.

[0031] As discussed above, the signaled delay value may include a combination of a decoding delay and a picture reordering delay. The picture reordering delay may correspond to a maximum difference between a presentation time and a decoding time for any picture in any representation. In some examples, the picture reordering delay may correspond to the maximum difference between presentation time and decoding time for any picture in the two representations involved in the switch. Presentation time corresponds to a relative time at which a picture is displayed, while decoding time corresponds to a relative time at which the picture is decoded. As noted above, presentation time and decoding time are not necessarily equal, and therefore, following decoding, decoded pictures may be reordered into presentation time ordering.

[0032] A server device may signal to a client device a delay value for seamless switching. The delay value may comprise the sum of the maximum decoding delay of any picture in any representation and the maximum difference between presentation time and decoding time of any picture in any representation of the same multimedia content. The client device may use this signaled delay value to perform a seamless switch between two representations. For example, the client device may determine an amount of data to have buffered in the current representation before switching to a different representation, such that seamless switching can be effected. That is, the client device may have smooth playback and seamless decoding, assuming that the available bandwidth is sufficient for the new representation.

[0033] To effect seamless switching, a client device may need to provide both seamless decoding during a switch between representations, as well as seamless display of decoded pictures. Seamless decoding may ensure that a second decoder is not needed to decode pictures from the new representation (although a client device may nevertheless include more than one decoder). Seamless displaying may correspond to having pictures ready for continuous playback during a switch between the representations.

[0034] The techniques of this disclosure also include forming switch groups from a set of representations, and signaling that a set of representations belong to the same switch group. A representation may include an initialization segment, which includes data for initializing a decoder and/or other elements of a destination device. The initialization segment may describe, for example, the movie box of a ISO base media file, the samples containing the sequence parameter set and picture parameter set, of the H.264/AVC bitstreams, and possibly sequence level supplemental enhancement information (SEI) messages. In some cases, the same initialization segment may be used to initialize the destination device for more than one representation. A switch group may include a set of representations that may each be initialized using the same initialization segment. In this manner, a client device may retrieve an initialization segment for a first representation of a switch group, then switch to a second representation of the switch group without retrieving an initialization segment specific to the second representation. That is, the client device may use the initialization segment for the first representation to decode the second representation. A switch group may include the representations which have timely aligned segments thus switching from one representation to another, when the segments all start with a random access point picture. In this case, on timely overlapped segments of different representations need to be requested, if switching between such the group happens.

[0035] In general, an I-picture may be referred to as a random access point (RAP). That is, a decoder may begin decoding video data starting at the I-picture and be capable of decoding some, if not all, of the pictures in the bitstream following the I-picture. There may be some pictures that are not correctly decodable, however, such as B-pictures having a display order that is earlier than the I-picture but a decoding order that is later than the I-picture. These B-pictures may depend from an earlier I-picture in the bitstream. An I-picture for which all subsequent pictures in the bitstream can be properly decoded without retrieving earlier pictures in the bitstream may be referred to as an instantaneous decoding refresh (IDR) random access point. An I-picture for which at least one subsequent picture in the bitstream cannot be properly decoded without retrieving earlier pictures in the bitstream may be referred to as an open decoding refresh (ODR) random access point.

[0036] Stated another way, an IDR picture may contain only intra-coded slices and may cause all reference pictures except for the IDR picture to be marked as “unused for reference.” A coded video sequence may be defined as a sequence of consecutive access units in decoding order from an IDR access unit, inclusive, to the next IDR access unit, exclusive, or to the end of the bitstream, whichever appears earlier. Each picture in a group of pictures (GOP) including an IDR picture can be properly decoded regardless of whether any previous pictures were decoded. Such a GOP may be referred to as a closed GOP. That is, a closed GOP is a GOP in which all pictures can be properly decoded, assuming all pictures of the GOP are available for decoding. In the example of H.264/AVC, a closed GOP may start from an IDR access unit, that is, an access unit including an IDR picture. Moreover, each picture in a coded video sequence can be properly decoded, assuming that each picture of the coded video sequence is available for decoding (including the initial IDR picture).

[0037] An open GOP may be defined as a group of pictures in which pictures preceding an I-picture of the GOP in display order are not correctly decodable, even when all pictures of the open GOP are available for decoding. An encoder may include information in the bitstream indicating that a GOP is an open GOP, such as a recovery point supplemental enhancement information (SEI) message. An I-picture of an open GOP may be referred to as an ODR picture. Although pictures following the open GOP in decoding order may be correctly decodable, certain pictures in the open GOP may refer to a previous I-picture in decoding order, and thus not be properly decodable unless that previous I-picture is available for decoding.

[0038] Generally, the decoder may begin decoding a bitstream starting from an IDR or ODR picture, and thus treat both IDR and ODR pictures as random access points within the bitstream. However, because there are some pictures following an ODR picture in the bitstream that are not properly decodable if the bitstream is accessed starting at the ODR picture, the decoder may exert unnecessary effort in attempting to decode these pictures. In accordance with the techniques of this disclosure, a server may signal whether a random access point picture is an IDR or ODR picture. The server may also signal byte ranges for the random access points, such that a client device may submit one or more HTTP partial GET requests to access only the IDR and/or ODR pictures. In this manner, the client may retrieve the IDR and/or ODR pictures to enable a trick mode display of the multimedia content, such as fast forward or rewind.

[0039] The techniques of this disclosure may be applied to video files conforming to any of ISO base media file format, Scalable Video Coding (SVC) file format, Advanced Video Coding (AVC) file format, Third Generation Partnership Project (3GPP) file format, and/or Multiview Video Coding (MVC) file format, or other similar video file formats.

[0040] The ISO Base Media File Format is designed to contain timed media information for a presentation in a flexible, extensible format that facilitates interchange, management, editing, and presentation of the media. ISO Base Media File format (ISO/IEC 14496-12:2004) is specified in MPEG-4 Part-12, which defines a general structure for time-based media files. It is used as the basis for other file formats in the family such as AVC file format (ISO/IEC 14496-15) defined support for H.264/MPEG-4 AVC video compression, 3GPP file format, SVC file format, and MVC file format. 3GPP file format and MVC file format are extensions of the AVC file format. ISO base media file format contains the timing, structure, and media information for timed sequences of media data, such as audio-visual presentations. The file structure may be object-oriented. A file can be decomposed into basic objects very simply and the structure of the objects is implied from their type.

[0041] Files conforming to the ISO base media file format (and extensions thereof) may be formed as a series of objects, called “boxes.” Data in the ISO base media file format may be contained in boxes, such that no other data needs to be contained within the file and there need not be data outside of boxes within the file. This includes any initial signature required by the specific file format. A “box” may be an object-oriented building block defined by a unique type identifier and length. Typically, a presentation is contained in one file, and the media presentation is self-contained. The movie container (movie box) may contain the metadata of the media and the video and audio frames may be contained in the media data container and could be in other files.

[0042] A representation (motion sequence) may be contained in several files. Timing and framing (position and size) information is generally in the ISO base media file and the ancillary files may essentially use any format. This representation may be ‘local’ to the system containing the representation, or may be provided via a network or other stream delivery mechanism.

[0043] The files may have a logical structure, a time structure, and a physical structure, and these structures are not required to be coupled. The logical structure of the file may be of a movie or video clip (potentially including both video and audio data) that in turn contains a set of time-parallel tracks. The time structure of the file may be that the tracks contain sequences of samples in time, and those sequences are mapped into the timeline of the overall movie by optional edit lists. The physical structure of the file may separate the data needed for logical, time, and structural de-composition, from the media data samples themselves. This structural information may be concentrated in a movie box, possibly extended in time by movie fragment boxes. The movie box may document the logical and timing relationships of the samples, and may also contain pointers to where they are located. Those pointers may be into the same file or another one, e.g., referenced by a URL.

[0044] Each media stream may be contained in a track specialized for that media type (audio, video etc.), and may further be parameterized by a sample entry. The sample entry may contain the 'name' of the exact media type (the type of decoder needed to decode the stream) and any parameterization of that decoder needed. The name may also take the form of a four-character code, e.g., "moov," or "trak." There are defined sample entry formats not only for MPEG-4 media, but also for the media types used by other organizations using this file format family.

[0045] Support for meta-data generally takes two forms. First, timed meta-data may be stored in an appropriate track, and synchronized as desired with the media data it is describing. Secondly, there may be general support for non-timed meta-data attached to the movie or to an individual track. The structural support for meta-data is general, and allows the storage of meta-data resources elsewhere in the file or in another file, in a manner similar to the storage of the media data, that is, the coded video pictures. In addition, these resources may be named, and may be protected.

[0046] The term "progressive download" is used to describe the transfer of digital media files from a server to a client, typically using the HTTP protocol, e.g., version 1.1 of HTTP. One example of progressive download is HTTP streaming. When initiated from a computer, the computer may begin playback of the media before the download is complete. One difference between streaming media and progressive download is in how the digital media data is received and stored by the end user device that is accessing the digital media. A media player that is capable of progressive download playback relies on metadata located in the header of the file to be intact and a local buffer of the digital media file as it is downloaded from a web server. At the point at which a specified amount of buffered data becomes available to the local playback device, the device may begin to play the media. This specified amount of buffered data may be embedded into the file by the producer of the content in the encoder settings and may be reinforced by additional buffer settings imposed by the media player of the client computer.

[0047] In HTTP streaming, rather than providing all data for a representation, including encoded audio and video samples, in the movie (MOOV) box, movie fragments (MOOF boxes) are supported, which may contain samples separately from the movie box. Each movie fragment may contain samples for a certain period of time. In general, a client may quickly seek to a desired time using the movie fragments. A movie fragment may contain continuous bytes of a file, such that in HTTP streaming (for example), a client may issue a single partial GET request to retrieve a movie fragment. A partial GET request specifies an identifier (e.g., uniform resource locator (URL) or uniform resource name (URN)) of a file, as well as a byte range within the file, to be retrieved. In some examples, the encoded audio and video samples may be contained entirely in movie fragments, such that the movie box itself does not contain any encoded audio or video samples.

[0048] AVC and 3GPP are extensions of the ISO base media file format, while SVC and MVC are extensions of the AVC file format. Accordingly, the techniques of this disclosure may be applied with respect to video files conforming to the ISO base media file format, the AVC file format and extensions thereof, e.g., SVC and MVC, and/or the 3GPP file format. The techniques may further be applied to these and other extensions of these formats, and may further be applied to extend other file formats to provide information describing switchable representations, and delay values to allow seamless switching between two or more representations.

[0049] The Motion Pictures Experts Group (MPEG) has developed an AVC file format, as an extension of the ISO base media file format. In the AVC file format, parameter sets, including the sequence parameter set and the picture parameter sets can be associated with the video elementary stream, which is the in the video track. It is also possible to have the parameter sets in another track, called a parameter set track, which includes the parameter set elementary stream containing the samples that are the SPS or PPS network abstraction layer (NAL) units. The location of SEI message NAL units, however, is not restricted in the AVC file format, as long as any SEI message is presented in a sample with a decoding time prior to any picture that uses this SEI message.

[0050] AVC file format is extended by the SVC and MVC extensions, to support the storage of SVC and MVC, respectively. In SVC and MVC, multiple layers/views can be contained in one video track or multiple video tracks. There are tools defined in the SVC and MVC specifications to construct operation points when the layers/views are presented in various organizations.

[0051] With respect to 3GPP as another example, HTTP/TCP/IP transport is supported for 3GPP files for download and progressive download. Furthermore, using HTTP for video streaming may provide some advantages, and the video streaming services based on HTTP are becoming popular. HTTP streaming may provide certain advantages, including that existing Internet components and protocols may be used, such that new efforts are not needed to develop new techniques for transporting video data over a network. Other transport protocols, e.g., RTP payload format, require intermediate network devices, e.g., middle boxes, to be aware of the media format and the signaling context. Also, HTTP streaming can be client-driven, which may avoid control issues.

[0052] For example, to exploit features to obtain optimal performance, the server may keep track of the size and content of packets which are not yet acknowledged. The server may also analyze the file structure and reconstruct the state of the client buffer to make RD-optimal switching/thinning decisions. In addition, constraints on the bit stream variations may be satisfied in order to stay compliant with negotiated profiles. HTTP does not necessarily require new hardware or software implementations at a Web server that has HTTP 1.1 implemented. HTTP streaming also provides TCP-friendliness and firewall traversal.

[0053] The techniques of this disclosure may be applicable to network streaming protocols, such as HTTP streaming, e.g., in accordance with dynamic adaptive streaming over HTTP (DASH). In HTTP streaming, frequently used operations include GET and partial GET. The GET operation retrieves a whole file associated a given uniform resource locator (URL) or uniform resource name (URN). The partial GET operation receives a byte range as an input parameter and retrieves a continuous number of bytes of a file corresponding to the received byte range. Thus, movie fragments may be provided for HTTP streaming, because a partial GET operation can get one or more individual movie fragments. Note that, in a movie fragment, there can be several track fragments of different tracks. In HTTP streaming, a media representation may be a structured collection of data that is accessible to the client. The client may request and download media data information to present a streaming service to a user.

[0054] In the example of streaming 3GPP data using HTTP streaming, there may be multiple representations for video and/or audio data of multimedia content. The manifest of such representations may be defined in a Media Presentation Description (MPD) data structure. A media presentation may correspond to a structured collection of data that is accessible to an HTTP streaming client device. The HTTP streaming client device may request and download media data information to present a streaming service to a user of the client device. A media presentation may be described in the MPD data structure, which may include updates of the MPD.

[0055] A media presentation may contain a sequence of one or more periods. Periods may be defined by a *Period* element in the MPD. Each period may have an attribute *start* in the MPD. The MPD may include a *start* attribute and an *availableStartTime* attribute for each period. For live services, the sum of the *start* attribute of the period and the MPD attribute *availableStartTime* may specify the availability time of the period in UTC format, in particular the first Media Segment of each representation in the corresponding period. For on-demand services, the *start* attribute of the first period may be 0. For any other period, the *start* attribute may specify a time offset between the start time of the corresponding Period relative to the start time of the first Period. Each period may extend until the start of the next Period, or until the end of the media presentation in the case of the last period. Period start times may be precise. They may reflect the actual timing resulting from playing the media of all prior periods.

[0056] Each period may contain one or more representations for the same media content. A representation may be one of a number of alternative encoded versions of audio or video data. The representations may differ by encoding types, e.g., by bitrate, resolution, and/or codec for video data and bitrate, language, and/or codec for audio data. The term representation may be used to refer to a section of encoded audio or video data corresponding to a particular period of the multimedia content and encoded in a particular way.

[0057] Representations of a particular period may be assigned to a group indicated by a *group* attribute in the MPD. Representations in the same group are generally considered alternatives to each other. For example, each representation of video data for a particular period may be assigned to the same group, such that any of the representations may be selected for decoding to display video data of the multimedia content for the corresponding period. The media content within one period may be represented by either one representation from group 0, if present, or the combination of at most one representation from each non-zero group, in some examples. Timing data for each representation of a period may be expressed relative to the start time of the period.

[0058] A representation may include one or more segments. Each representation may include an initialization segment, or each segment of a representation may be self-initializing. When present, the initialization segment may contain initialization information for accessing the representation. In general, the initialization segment does not contain media data. A segment may uniquely referenced by an identifier, such as a uniform resource locator (URL), uniform resource name (URN), or uniform resource identifier (URI). The MPD may provide the identifiers for each segment. In some examples, the MPD may also provide byte ranges in the form of a *range* attribute, which may correspond to the data for a segment within a file accessible by the URL, URN, or URI.

[0059] Each representation may also include one or more media components, where each media component may correspond to an encoded version of one individual media type, such as audio, vide, or timed text (e.g., for closed captioning). Media components may be time-continuous across boundaries of consecutive media segments within one representation.

[0060] The MPD may also support partial representations, e.g., by providing a *group* attribute. A partial representation may correspond to a video or audio track of a multimedia file. The MPD may indicate that two partial representations have the same *group* attribute when the two partial representations are switchable to each other.

[0061] The server device may also signal whether two representations are efficiently switchable to each other. Two representations may be considered efficiently switchable if the initialization segment of either of the representations may be used when decoding the other representation. That is, a client device may receive an initialization segment for a current representation, then switch to a new representation without requesting an initialization segment for the new representation.

[0062] The server device may set a flag in an MPD data structure indicating whether a representation is a partial representation or not. For example, the MPD data structure may include a switch identifier for each representation. The server device may set the switch identifiers of two or more efficiently switchable representations to the same value. In this manner, a client device may determine a set of representations that are efficiently switchable to each other based on each of the representations having the same switch identifier. The server may further indicate, in the MPD, whether segments of the representations are timely aligned.

[0063] The MPD may also contain a *minBufferTime* attribute, which indicates a minimum amount of initially buffered media needed to ensure smooth playout, provided that each representation is delivered at or above the value of a *bandwidth* attribute of the MPD. In this manner, a client device may determine how much media to buffer before beginning playback, assuming a certain amount of available bandwidth. The client may further select representations based on the amount of available bandwidth, e.g., initially or when network conditions change, based on the *bandwidth* attribute. These attributes may provide a simple buffer model in a solution to enable continuous decoding and presentation of content under various network conditions.

[0064] In each period, representations may be switchable to each other. The MPD may provide a *bitstreamSwitchingFlag* attribute. When this flag is true, it indicates that the result of splicing on a bitstream level of any two time-sequential media segments within a period from any two different representations containing the same media types complies with the media segment format. When this flag is true, all bitstreams may be assumed to be timely aligned and to have the same profile and level indicators, and that many sequence parameters are the same (in the example of H.264/AVC coding).

[0065] During streaming and playback, a client device may switch between representations. For example, the client device may determine that network conditions have changed such that available bandwidth has either increased or decreased. When bandwidth has increased, the client device may switch to a higher quality representation, assuming the client device is capable of decoding and rendering the data of the higher quality representation. On the other hand, when bandwidth has decreased, the client device may switch to a lower quality representation that consumes less bandwidth. In general, based upon updated information during an ongoing media presentation, a client device may switch representations. Switching to a new representation may include procedures similar to beginning to retrieve or seeking to a particular position within a representation from the time point (e.g., the temporal location) where the previous representation was stopped.

[0066] After determining that a switch to a new representation is appropriate, the client device may begin streaming the new representation starting at a random access point. That is, the client device may seek to a random access point in the new representation at a desired presentation time that is later than, but close to, the current presentation time. By presenting data from the previous representation up to the random access point, then presenting data from the new representation at and following the random access point, the client device may achieve seamless switching. In accordance with the techniques of this disclosure, the client device may use a signaled delay time to determine a random access point of the new representation from which to begin streaming data of the new representation. The server device may signal the delay time in the MPD data structure for a media content. In some examples, a source device may align random access points across different representations, which may provide advantages in locating random access points in the representations.

[0067] In some cases, two switchable representations may not be time-aligned. For example, random access points in the two representations may occur at different temporal positions. In such a case, the server may set a flag or provide other information to indicate that the two representations are not aligned. In such a case, switching from one representation to the other may be difficult. For example, the end of the fragment in the current representation and the frames starting at the random access point in the new fragment of the switched-to representation may have a display time overlap. This disclosure also provides techniques for seamlessly switching between two non-aligned representations. In particular, a client device and/or a server device may introduce additional delay, to cause the client device to continue retrieving data from the current representation before switching to the new representation at a random access point. This delay may depend on the GOP structure of the current representation as well as the location of the picture in the GOP.

[0068] In accordance with the techniques of this disclosure, the server device may signal an overlap delay value representative of the difference between a temporal position of a non-random access point picture in one representation and a temporal position of a random access point picture in another representation. The server device may signal the overlap delay value in an MPD for the representations. In some examples, the server device may signal the maximum overlap delay value of each representation for a whole period, or even for a whole service. In some examples, the server device may signal a maximum delay value as the sum of overlap delay and the delay value discussed above for fragments that are aligned. Thus, the server device may signal a single delay value that may include the sum of a maximum decoding delay, a maximum picture reordering delay, and a maximum overlap delay for two or more switchable representations.

[0069] The server device may further signal a minimum delay value for a representation. Assuming that the representation is delivered at or above the value of its bandwidth attribute, the client device may be assured of smooth playout of the representation if sufficient data is buffered to cover the minimum delay. The server device may be configured to modify the calculation of a minimum buffer time value to cover the case of seamless switching. The minimum buffer time value (minBufferTime) may provide the minimum amount of initially buffered media that is needed to ensure smooth playout and seamless switching, provided that each Representation is delivered at or above the value of its bandwidth attribute. In this manner, the minimum buffer time value may correspond to a simple buffer model in a solution to enable continuous decoding and presentation of content under various network conditions.

[0070] In general, a user of a client device may pause or stop a media presentation during playback. When the user pauses or stops the presentation, the client device may stop requesting media segments or portions thereof, until the user resumes playback. To resume, the client device may request media segments starting with the fragment immediately following the last requested fragment. The MPD for any specific representation may provide a *TrickMode* element for the representation, in which case the representation can be assumed to be explicitly enabled for use with trick modes. Trick modes generally include fast forward and rewind modes, or other such modes in which playback is altered relative to the "normal" playback. The MPD may provide an *alternatePlayoutRate* attribute, in which case the client device may play the representation with speeds corresponding to the *alternatePlayoutRate*, with the same decoder profile and level as the normal playout rate. The client device may also use multiple representations to support trick mode behavior.

[0071] FIG. 1 is a block diagram illustrating an example system 10 in which audio/video (A/V) source device 20 transports audio and video data to A/V destination device 40. System 10 of FIG. 1 may correspond to a video teleconference system, a server/client system, a broadcaster/receiver system, or any other system in which video data is sent from a source device, such as A/V source device 20, to a destination device, such as A/V destination device 40. In some examples, A/V source device 20 and A/V destination device 40 may perform bidirectional information exchange. That is, A/V source device 20 and A/V destination device 40 may be capable of both encoding and decoding (and transmitting and receiving) audio and video data. In some examples, audio encoder 26 may comprise a voice encoder, also referred to as a vocoder.

[0072] A/V source device 20, in the example of FIG. 1, comprises audio source 22 and video source 24. Audio source 22 may comprise, for example, a microphone that produces electrical signals representative of captured audio data to be encoded by audio encoder 26. Alternatively, audio source 22 may comprise a storage medium storing previously recorded audio data, an audio data generator such as a computerized synthesizer, or any other source of audio data. Video source 24 may comprise a video camera that produces video data to be encoded by video encoder 28, a storage medium encoded with previously recorded video data, a video data generation unit, or any other source of video data.

[0073] In general, source device 20 may act as a server of encoded video content. Video encoder 28 may encode the same video content in different ways, e.g., using different encoding standards and/or encoding at different bit rates, frame rates, or versions (e.g., two-dimensional or three-dimensional). Thus, the same video content may be available in a variety of different representations. Source device 20 may advertise the availability of different representations of the video content. Destination device 40 may be capable of selecting one of the representations, e.g., based on characteristics of the representations and decoding and rendering capabilities of destination device 40, as well as current network conditions, e.g., available bandwidth.

[0074] Source device 20 may generally act as a server, e.g., in accordance with HTTP version 1.1. Thus, source device 20 may provide data for a representation to destination device 40 in accordance with streaming network protocols and network-based streaming standards, such as the DASH standard. In some cases, as discussed in greater detail below, representations may be switchable. That is, destination device 40 may begin retrieving data of one representation, then switch to a different representation mid-stream. In accordance with the techniques of this disclosure, destination device 40 may seamlessly switch between two representations, without needing to decode two representations in parallel and without encountering gaps in playback.

[0075] Audio source 22 and video source 24 may capture raw audio and video data. Raw audio and video data may comprise analog or digital data. Analog data may be digitized before being encoded by audio encoder 26 and/or video encoder 28. Audio source 22 may obtain audio data from a speaking participant while the speaking participant is speaking, and video source 24 may simultaneously obtain video data of the speaking participant. In other examples, audio source 22 may comprise a computer-readable storage medium comprising stored audio data, and video source 24 may comprise a computer-readable storage medium comprising stored video data. In this manner, the techniques described in this disclosure may be applied to live, streaming, real-time audio and video data or to archived, pre-recorded audio and video data.

[0076] Audio frames that correspond to video frames are generally audio frames containing audio data that was captured by audio source 22 contemporaneously with video data captured by video source 24 that is contained within the video frames. For example, while a speaking participant generally produces audio data by speaking, audio source 22 captures the audio data, and video source 24 captures video data of the speaking participant at the same time, that is, while audio source 22 is capturing the audio data. Hence, an audio frame may temporally correspond to one or more particular video frames. Accordingly, an audio frame corresponding to a video frame generally corresponds to a situation in which audio data and video data were captured at the same time and for which an audio frame and a video frame comprise, respectively, the audio data and the video data that was captured at the same time.

[0077] In some examples, audio encoder 26 may encode a timestamp in each encoded audio frame that represents a time at which the audio data for the encoded audio frame was recorded, and similarly, video encoder 28 may encode a timestamp in each encoded video frame that represents a time at which the video data for encoded video frame was recorded. In such examples, an audio frame corresponding to a video frame may comprise an audio frame comprising a timestamp and a video frame comprising the same timestamp. A/V source device 20 may include an internal clock from which audio encoder 26 and/or video encoder 28 may generate the timestamps, or that audio source 22 and video source 24 may use to associate audio and video data, respectively, with a timestamp.

[0078] In some examples, audio source 22 may send data to audio encoder 26 corresponding to a time at which audio data was recorded, and video source 24 may send data to video encoder 28 corresponding to a time at which video data was recorded. In some examples, audio encoder 26 may encode a sequence identifier in encoded audio data to indicate a relative temporal ordering of encoded audio data but without necessarily indicating an absolute time at which the audio data was recorded, and similarly, video encoder 28 may also use sequence identifiers to indicate a relative temporal ordering of encoded video data. Similarly, in some examples, a sequence identifier may be mapped or otherwise correlated with a timestamp.

[0079] The techniques of this disclosure are generally directed to the storage and transport of encoded multimedia (e.g., audio and video) data, and reception and subsequent interpretation and decoding of the transported multimedia data. As shown in the example of FIG. 1, video source 24 may provide a plurality of views of a scene to video encoder 28.

[0080] Each individual stream of data (whether audio or video) is referred to as an elementary stream. An elementary stream is a single, digitally coded (possibly compressed) component of a program. For example, the coded video or audio part of the program can be an elementary stream. An elementary stream may be converted into a packetized elementary stream (PES) before being encapsulated within a video file. Within the same program, a stream ID is used to distinguish the PES-packets belonging to one elementary stream from the other. The basic unit of data of an elementary stream is a packetized elementary stream (PES) packet. Thus, each view of MVC video data corresponds to respective elementary streams. Similarly, audio data corresponds to one or more respective elementary streams.

[0081] In the example of FIG. 1, encapsulation unit 30 receives elementary streams comprising video data from video encoder 28 and elementary streams comprising audio data from audio encoder 26. In some examples, video encoder 28 and audio encoder 26 may each include packetizers for forming PES packets from encoded data. In other examples, video encoder 28 and audio encoder 26 may each interface with respective packetizers for forming PES packets from encoded data. In still other examples, encapsulation unit 30 may include packetizers for forming PES packets from encoded audio and video data.

[0082] A representation, as that term is used in this disclosure, may include a combination of audio data and video data, e.g., an audio elementary stream and a video elementary stream delivered by A/V source device 20. Multimedia content may have one or more representations, e.g., when video data is encoded at different frame rates, different bit rates, or different resolutions. Similarly, audio data for multimedia content may correspond to various languages, e.g., English, French, and Chinese. A representation may correspond to a selection of an audio elementary stream and one or more video elementary streams (e.g., to support three-dimensional video data).

[0083] Encapsulation unit 30 receives PES packets for elementary streams of a program from audio encoder 26 and video encoder 28 and forms corresponding network abstraction layer (NAL) units from the PES packets. In the example of H.264/AVC (Advanced Video Coding), coded video segments are organized into NAL units, which provide a "network-friendly" video representation addressing applications such as video telephony, storage, broadcast, or streaming. NAL units can be categorized to Video Coding Layer (VCL) NAL units and non-VCL NAL units. VCL units may correspond to data generated by the core compression engine, and may include block, macroblock, and/or slice level data. Other NAL units may be non-VCL NAL units. In some examples, a coded picture in one time instance, normally presented as a primary coded picture, may be contained in an access unit, which may include one or more NAL units.

[0084] Non-VCL NAL units may include parameter set NAL units and SEI NAL units, among others. Parameter sets may contain sequence-level header information (in sequence parameter sets (SPS)) and the infrequently changing picture-level header information (in picture parameter sets (PPS)). With parameter sets (e.g., PPS and SPS), infrequently changing information need not to be repeated for each sequence or picture, hence coding efficiency may be improved. Furthermore, the use of parameter sets may enable out-of-band transmission of the important header information, avoiding the need for redundant transmissions for error resilience. In out-of-band transmission examples, parameter set NAL units may be transmitted on a different channel than other NAL units, such as SEI NAL units.

[0085] Supplemental Enhancement Information (SEI) may contain information that is not necessary for decoding the coded pictures samples from VCL NAL units, but may assist in processes related to decoding, display, error resilience, and other purposes. SEI messages may be contained in non-VCL NAL units. SEI messages are the normative part of some standard specifications, and thus are not always mandatory for standard compliant decoder implementation. SEI messages may be sequence level SEI messages or picture level SEI messages. Some sequence level information may be contained in SEI messages, such as scalability information SEI messages in the example of SVC and view scalability information SEI messages in MVC. These example SEI messages may convey information on, e.g., extraction of operation points and characteristics of the operation points.

[0086] In accordance with the techniques of this disclosure, encapsulation unit 30 may signal that two or more representations are efficiently switchable to each other. Encapsulation unit 30 may, for example, determine whether initialization segments of the two or more representations can be used interchangeably to decode and/or render encoded samples of the representations. Encapsulation unit 30 may generate data used to indicate whether representations are efficiently switchable to each other. Accordingly, destination device 40 may request the data, e.g., in order to switch from one representation to another during streaming.

[0087] Encapsulation unit 30 may also calculate a delay value representative of a maximum delay when switching between two efficiently switchable representations. That is, the delay value may generally represent an amount of delay that can be expected (assuming a constant bandwidth availability) when switching between two representations. Accordingly, encapsulation unit 30 may calculate the delay value as the sum of a maximum decoding delay and a maximum picture reordering delay, when the representations are timely aligned. When the representations are not timely aligned, e.g., when random access points occur at different temporal locations within the representations, encapsulation unit 30 may further calculate an overlap delay and include the overlap delay in the calculated delay value. Source device 20 may provide the delay value to destination device 40, e.g., in response to a request for the delay value from destination device 40.

[0088] Destination device 40 may be configured to retrieve and buffer a sufficient amount of data of a representation, based on the delay value, in order to perform seamless switching between two representations. Accordingly, destination device 40 may buffer a sufficient amount of data that destination device 40 is able to begin requesting data of a different representation, and begin decoding the data of the new representation, without suffering a display gap. That is, while retrieving data from the new representation, destination device 40 may decode and render data from the previous representation. Moreover, destination device 40 may be configured such that simultaneous decoding of two representations is not necessary. That is, after accessing a random access point of the new representation, destination device 40 need not decode data from two or more representations corresponding to the same temporal location in video playback. This may improve battery performance, in that video decoder 48 need not decode multiple samples for the same temporal location.

[0089] A NAL unit including video data in its payload may comprise various granularity levels of video data. For example, a NAL unit may comprise a block of video data, a macroblock, a plurality of macroblocks, a slice of video data, or an entire frame of video data. Encapsulation unit 30 may receive encoded video data from video encoder 28 in the form of PES packets of elementary streams. Encapsulation unit 30 may associate each elementary stream with a corresponding program.

[0090] Encapsulation unit 30 may also assemble access units from a plurality of NAL units. In general, an access unit may comprise one or more NAL units for representing a frame of video data, as well audio data corresponding to the frame when such audio data is available. An access unit generally includes all NAL units for one output time instance, e.g., all audio and video data for one time instance. For example, if each view has a frame rate of 20 frames per second (fps), then each time instance may correspond to a time interval of 0.05 second. During this time interval, the specific frames for all views of the same access unit (the same time instance) may be rendered simultaneously. In an example corresponding to H.264/AVC, an access unit may comprise a coded picture in one time instance, which may be presented as a primary coded picture. Accordingly, an access unit may comprise all audio and video frames of a common temporal instance, e.g., all views corresponding to time *X*. This disclosure also refers to an encoded picture of a particular view as a "view component." That is, a view component may comprise an encoded picture (or frame) for a particular view at a particular time. Accordingly, an access unit may be defined as comprising all view components of a common temporal instance. The decoding order of access units need not necessarily be the same as the output or display order.

[0091] As with most video coding standards, H.264/AVC defines the syntax, semantics, and decoding process for error-free bitstreams, any of which conform to a certain profile or level. H.264/AVC does not specify the encoder, but the encoder is tasked with guaranteeing that the generated bitstreams are standard-compliant for a decoder. In the context of video coding standard, a "profile" corresponds to a subset of algorithms, features, or tools and constraints that apply to them. As defined by the H.264 standard, for example, a "profile" is a subset of the entire bitstream syntax that is specified by the H.264 standard. A "level" corresponds to the limitations of the decoder resource consumption, such as, for example, decoder memory and computation, which are related to the resolution of the pictures, bit rate, and macroblock (MB) processing rate. A profile may be signaled with a profile_idc (profile indicator) value, while a level may be signaled with a level_idc (level indicator) value.

[0092] The H.264 standard, for example, recognizes that, within the bounds imposed by the syntax of a given profile, it is still possible to require a large variation in the performance of encoders and decoders depending upon the values taken by syntax elements in the bitstream such as the specified size of the decoded pictures. The H.264 standard further recognizes that, in many applications, it is neither practical nor economical to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile. Accordingly, the H.264 standard defines a "level" as a specified set of constraints imposed on values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively, these constraints may take the form of constraints on arithmetic combinations of values (e.g., picture width multiplied by picture height multiplied by number of pictures decoded per second). The H.264 standard further provides that individual implementations may support a different level for each supported profile.

[0093] A decoder conforming to a profile ordinarily supports all the features defined in the profile. For example, as a coding feature, B-picture coding is not supported in the baseline profile of H.264/AVC but is supported in other profiles of H.264/AVC. A decoder conforming to a level should be capable of decoding any bitstream that does not require resources beyond the limitations defined in the level. Definitions of profiles and levels may be helpful for interpretability. For example, during video transmission, a pair of profile and level definitions may be negotiated and agreed for a whole transmission session. More specifically, in H.264/AVC, a level may define, for example, limitations on the number of macroblocks that need to be processed, decoded picture buffer (DPB) size, coded picture buffer (CPB) size, vertical motion vector range, maximum number of motion vectors per two consecutive MBs, and whether a B-block can have sub-macroblock partitions less than 8x8 pixels. In this manner, a decoder may determine whether the decoder is capable of properly decoding the bitstream.

[0094] Parameter sets generally contain sequence-layer header information in sequence parameter sets (SPS) and the infrequently changing picture-layer header information in picture parameter sets (PPS). With parameter sets, this infrequently changing information need not be repeated for each sequence or picture; hence, coding efficiency may be improved. Furthermore, the use of parameter sets may enable out-of-band transmission of header information, avoiding the need for redundant transmissions to achieve error resilience. In out-of-band transmission, parameter set NAL units are transmitted on a different channel than the other NAL units.

[0095] The examples of SVC and MVC provide sequence parameter set (SPS) extensions for scalable video or multiview video. The SPS extension information may be contained in sub-SPS NAL units, which may have a different NAL unit type than the other NAL units. Profile and level information for the supported operation points may be included in NAL units including SPS extension information.

[0096] A media representation may include a media presentation description (MPD), which may contain descriptions of different alternative representations (e.g., video services with different qualities) and the description may include, e.g., codec information, a profile value, and a level value. Destination device 40 may retrieve the MPD of a media representation to determine how to access movie fragments of various presentations. Movie fragments may be located in movie fragment boxes (moof boxes) of video files.

[0097] Video compression standards such as ITU-T H.261, H.262, H.263, MPEG-1, MPEG-2 and H.264/MPEG-4 part 10 make use of motion compensated temporal prediction to reduce temporal redundancy. The encoder uses a motion compensated prediction from some previously encoded pictures (also referred to herein as frames) to predict the current coded pictures according to motion vectors. There are three major picture types in typical video coding. They are Intra coded picture ("I-pictures" or "I-frames"), Predicted pictures ("P-pictures" or "P-frames") and Bi-directional predicted pictures ("B-pictures" or "B-frames"). P-pictures use only the reference picture before the current picture in temporal order. In a B-picture, each block of the B-picture may be predicted from one or two reference pictures. These reference pictures could be located before or after the current picture in temporal order.

[0098] In accordance with the H.264 coding standard, as an example, B-pictures use two lists of previously-coded reference pictures, list 0 and list 1. These two lists can each contain past and/or future coded pictures in temporal order. Blocks in a B-picture may be predicted in one of several ways: motion-compensated prediction from a list 0 reference picture, motion-compensated prediction from a list 1 reference picture, or motion-compensated prediction from the combination of both list 0 and list 1 reference pictures. To get the combination of both list 0 and list 1 reference pictures, two motion compensated reference areas are obtained from list 0 and list 1 reference picture respectively. Their combination may be used to predict the current block.

[0099] The ITU-T H.264 standard supports intra prediction in various block sizes, such as 16 by 16, 8 by 8, or 4 by 4 for luma components, and 8x8 for chroma components, as well as inter prediction in various block sizes, such as 16x16, 16x8, 8x16, 8x8, 8x4, 4x8 and 4x4 for luma components and corresponding scaled sizes for chroma components. In this disclosure, "NxN" and "N by N" may be used interchangeably to refer to the pixel dimensions of the block in terms of vertical and horizontal dimensions, e.g., 16x16 pixels or 16 by 16 pixels. In general, a 16x16 block will have 16 pixels in a vertical direction (y = 16) and 16 pixels in a horizontal direction (x = 16). Likewise, an NxN block generally has N pixels in a vertical direction and N pixels in a horizontal direction, where N represents a nonnegative integer value. The pixels in a block may be arranged in rows and columns. Blocks may have different numbers of pixels in the horizontal and vertical dimensions. That is, blocks may include NxM pixels, where N is not necessarily equal to M.

[0100] Block sizes that are less than 16 by 16 may be referred to as partitions of a 16 by 16 macroblock. Video blocks may comprise blocks of pixel data in the pixel domain, or blocks of transform coefficients in the transform domain, e.g., following application of a transform such as a discrete cosine transform (DCT), an integer transform, a wavelet transform, or a conceptually similar transform to the residual video block data representing pixel differences between coded video blocks and predictive video blocks. In some cases, a video block may comprise blocks of quantized transform coefficients in the transform domain.

[0101] Smaller video blocks can provide better resolution, and may be used for locations of a video frame that include high levels of detail. In general, macroblocks and the various partitions, sometimes referred to as sub-blocks, may be considered video blocks. In addition, a slice may be considered to be a plurality of video blocks, such as macroblocks and/or sub-blocks. Each slice may be an independently decodable unit of a video frame. Alternatively, frames themselves may be decodable units, or other portions of a frame may be defined as decodable units. The term "coded unit" or "coding unit" may refer to any independently decodable unit of a video frame such as an entire frame, a slice of a frame, a group of pictures (GOP) also referred to as a sequence, or another independently decodable unit defined according to applicable coding techniques.

[0102] The term macroblock refers to a data structure for encoding picture and/or video data according to a two-dimensional pixel array that comprises 16x16 pixels. Each pixel comprises a chrominance component and a luminance component. Accordingly, the macroblock may define four luminance blocks, each comprising a two-dimensional array of 8x8 pixels, two chrominance blocks, each comprising a two-dimensional array of 16x16 pixels, and a header comprising syntax information, such as a coded block pattern (CBP), an encoding mode (e.g., intra- (I), or inter- (P or B) encoding modes), a partition size for partitions of an intra-encoded block (e.g., 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, or 4x4), or one or more motion vectors for an inter-encoded macroblock.

[0103] Video encoder 28, video decoder 48, audio encoder 26, audio decoder 46, encapsulation unit 30, and decapsulation unit 38 each may be implemented as any of a variety of suitable processing circuitry, as applicable, such as one or more microprocessors, digital signal processors (DSPs), application specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), discrete logic circuitry, software, hardware, firmware or any combinations thereof. Each of video encoder 28 and video decoder 48 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined video encoder/decoder (CODEC). Likewise, each of audio encoder 26 and audio decoder 46 may be included in one or more encoders or decoders, either of which may be integrated as part of a combined CODEC. An apparatus including video encoder 28, video decoder 48, audio encoder audio encoder 26, audio decoder 46, encapsulation unit 30, and/or decapsulation unit 38 may comprise an integrated circuit, a microprocessor, and/or a wireless communication device, such as a cellular telephone.

[0104] After encapsulation unit 30 has assembled NAL units and/or access units into a video file based on received data, encapsulation unit 30 passes the video file to output interface 32 for output. In some examples, encapsulation unit 30 may store the video file locally or send the video file to a remote server via output interface 32, rather than sending the video file directly to destination device 40. Output interface 32 may comprise, for example, a transmitter, a transceiver, a device for writing data to a computer-readable medium such as, for example, an optical drive, a magnetic media drive (e.g., floppy drive), a universal serial bus (USB) port, a network interface, or other output interface. Output interface 32 outputs the video file to a computer-readable medium 34, such as, for example, a transmission signal, a magnetic medium, an optical medium, a memory, a flash drive, or other computer-readable medium.

[0105] Ultimately, input interface 36 retrieves the data from computer-readable medium 34. Input interface 36 may comprise, for example, an optical drive, a magnetic media drive, a USB port, a receiver, a transceiver, or other computer-readable medium interface. Input interface 36 may provide the NAL unit or access unit to decapsulation unit 38. Decapsulation unit 38 may decapsulate a elements of a video file into constituent PES streams, depacketize the PES streams to retrieve encoded data, and send the encoded data to either audio decoder 46 or video decoder 48, depending on whether the encoded data is part of an audio or video stream, e.g., as indicated by PES packet headers of the stream. Audio decoder 46 decodes encoded audio data and sends the decoded audio data to audio output 42, while video decoder 48 decodes encoded video data and sends the decoded video data, which may include a plurality of views of a stream, to video output 44.

[0106] FIG. 2 is a block diagram illustrating components of an example encapsulation unit 30. In the example of FIG. 2, encapsulation unit 30 includes video input interface 80, audio input interface 82, video file creation unit 60, and video file output interface 84. Video file creation unit 60, in this example, includes NAL unit constructor 62, switch information unit 64, and initialization segment unit 66.

[0107] Video input interface 80 and audio input interface 82 receive encoded video and audio data, respectively. Video input interface 80 and audio input interface 82 may receive encoded video and audio data as the data is encoded, or may retrieve encoded video and audio data from a computer-readable medium. Upon receiving encoded video and audio data, video input interface 80 and audio input interface 82 pass the encoded video and audio data to video file creation unit 60 for assembly into a video file.

[0108] Video file creation unit 60 may correspond to a control unit including hardware, software, and/or firmware configured to perform the functions and procedures attributed thereto. The control unit may further perform the functions attributed to encapsulation unit 30 generally. For examples in which video file creation unit 60 is embodied in software and/or firmware, encapsulation unit 30 may include a computer-readable medium comprising instructions for video file creation unit 60 and a hardware-based processing unit to execute the instructions. Each of the sub-units of video file creation unit 60 (NAL unit constructor 62, switch information unit 64, and initialization segment unit 66, in this example) may be implemented as individual hardware units and/or software modules, and may be functionally integrated or further separated into additional sub-units.

[0109] Video file creation unit 60 may correspond to any suitable processing unit or processing circuitry, such as, for example, one or more microprocessors (which may correspond to multiple cores of a single integrated circuit), application-specific integrated circuits (ASICs), field programmable gate arrays (FPGAs), digital signal processors (DSPs), or any combination thereof. Video file creation unit 60 may further include a non-transitory computer-readable medium storing instructions for any or all of NAL unit constructor 62, switch information unit 64, and initialization segment unit 66, as well as a processor for executing the instructions.

[0110] In general, video file creation unit 60 may create a video file including the received audio and video data. NAL unit constructor 62 may form NAL units including encoded video and audio samples. Video file creation unit 60 may further be configured to assemble representations including encoded video and audio samples. In some examples, video file creation unit 60 may create multiple representations, which may be switchable to each other, in some examples.

[0111] Switch information unit 64 may be configured to generate information for switching between two or more of the representations. For example, switch information unit 64 may be configured to calculate a delay value based on the sum of a maximum decoding delay, a maximum picture reordering delay, and (when the representations are not timely aligned) a maximum overlap delay. Client devices may use the delay value to determine an amount of data of any representation to have buffered in order to seamlessly switch from one representation to another. Thus, the client devices may have a sufficient amount of data buffered to account for delay associated with decoding buffered pictures, reordering the pictures, and to ensure that pictures having a presentation time earlier than a random access point of the new representation can be decoded and rendered during the time that the client device retrieves, decodes, and renders data of the new representation.

[0112] Initialization segment unit 66 may generate initialization information for accessing a particular representation. In general, initialization information in an initialization segment does not contain media data. Initialization segment unit 66 may generate initialization segments for each of the representations. However, in some cases, the same initialization segments can be used to initialize multiple representations. When this is true, initialization segment unit 66 may generate information indicating that representations are efficiently switchable. That is, if a first representation and a second representation are efficiently switchable, a client device may retrieve an initialization segment of the first representation, retrieve and decode data of the first representation, then begin retrieving and decoding data of the second representation based on the initialization segment of the first representation, without retrieving an initialization segment of the second representation.

[0113] In some examples, video file creation unit 60 may generate a media presentation description (MPD) data structure that describes a media presentation. In accordance with the techniques of this disclosure, video file creation unit 60 may generate a modified MPD data structure that indicates whether two or more representations are efficiently switchable to each other, and a delay value for seamlessly switching between two representations. Table 1 below provides an example MPD data structure in accordance with the techniques of this disclosure. Table 1 is an example in which video file creation unit 60 signals information in the period level. In other examples, the information may be signaled at other levels.

[0114] The “Element or Attribute Name” column describes the name of the syntax object. The “Type” column describes whether the syntax object is an element or an attribute. The “Cardinality” column describes the cardinality of the syntax object, that is, the number of instances of the syntax object in an instance of a a data structure corresponding to Table 1. The “Optionality” column describes whether the syntax object is optional, in this example, where “M” indicates mandatory, “O” indicates optional, “OD” indicates optional with a default value, and “CM” indicates conditionally mandatory. The “description” column describes the semantics of the corresponding syntax object.

[0115] In this example, video file creation unit 60 may set a single flag (seamlessSwitchingFlag) to indicate whether representations are switchable. If the flag is set to true, a client device may assume that the representations are seamlessly switchable, given that an initial delay is equal to or larger than the minBufferTime signaled in the MPD. When the seamlessSwitchingFlag of this example is set to true, the client device may further assume that no playout gap and no parallel decoding of the representations are needed to perform seamless switching. The client may follow the techniques of this disclosure, e.g., as described with respect to FIG. 8 below, to perform seamless switching.

TABLE 1

Element or Attribute Name	**Type**	**Cardinality**	**Optionality**	**Description**
MPD	E	1	M	The root element that carries the Media Presentation Description for a Media Presentation.
type	A		OD default: OnDemand	"OnDemand" or "Live". Indicates the type of the Media Presentation. Currently, on-demand and live types are defined. If not present, the type of the presentation shall be inferred as OnDemand.
…				
minBufferTime	A		M	Provides the minimum amount of initially buffered media that is needed to ensure smooth playout provided that each Representation is delivered at or above the value of its *bandwidth* attribute.
minBufferSwitching	A		M	Provides the minimum amount of initially buffered media that is

Element or Attribute Name	**Type**	**Cardinality**	**Optionality**	**Description**
				needed to ensure seamless switching between Representations provided that each Representation is delivered at or above the value of its *bandwidth* attribute.
…	A			
Period	E	1…N	M	Provides the information of a Period
…	A			…
Representation	E	1..N	M	May be used to provide information about the original source (for example content provider) of the Media Presentation.
…	A			
efficientSwitchGroup	A		O Default: False	When present, indicate the representation is a mulitiview video plus depth representation.
reUseInitSegmentFlag	A	0 , 1	O Default: False	When True, indicate that the Representation utilizes the initialisation segment of the previous representation with the same repID and with a closest presentation time.

[0116] FIG. 3 is a conceptual diagram illustrating a sequence of coded video frames 100–132. The frames are shaded differently to indicate positions within a hierarchical prediction structure. For example, frames 100, 116, and 132 are shaded black to represent that frames 100, 116, 132 are at the top of the hierarchical prediction structure. Frames 100, 116, 132 may comprise, for example, intra-coded frames or inter-coded frames that are predicted from other frames in a single direction (e.g., P-frames). When intra-coded, frames 100, 116, 132 are predicted solely from data within the same frame. When inter-coded, frame 116, for example, may be coded relative to data of frame 100, as indicated by the dashed arrow from frame 116 to frame 100. Frames 116, 132 form key frames of groups of pictures (GOPs) 134, 136, respectively.

[0117] Frames 108, 124 are darkly shaded to indicate that they are next in the encoding hierarchy following frames 100, 116, and 132. Frames 108, 124 may comprise bi-directional, inter-mode prediction encoded frames. For example, frame 108 may be predicted from data of frames 100 and 116, while frame 124 may be predicted from frames 116 and 132. Frames 104, 112, 120, and 128 are lightly shaded to indicate that they are next in the encoding hierarchy following frames 108 and 124. Frames 104, 112, 120, and 128 may also comprise bi-directional, inter-mode prediction encoded frames. For example, frame 104 may be predicted from frames 100 and 108, frame 112 may be predicted from frames 108 and 116, frame 120 may be predicted from frame 116 and 124, and frame 128 may be predicted from frame 124 and 132.

[0118] Finally, frames 102, 106, 110, 114, 118, 122, 126, and 130 are shaded white to indicate that these frames are last in the encoding hierarchy. Frames 102, 106, 110, 114, 118, 122, 126, and 130 may be bi-directional, inter-mode prediction encoded frames. Frame 102 may be predicted from frames 100 and 104, frame 106 may be predicted from frames 104 and 108, frame 110 may be predicted from frames 108 and 112, frame 114 may be predicted from frames 112 and 116, frame 118 may be predicted from frame 116 and 120, frame 122 may be predicted from frames 120 and 124, frame 126 may be predicted from frames 124 and 128, and frame 130 may be predicted from frames 128 and 132.

[0119] Frames 100–132 are illustrated in display order. That is, following decoding, frame 100 is displayed before frame 102, frame 102 is displayed before frame 104, and so on. However, due to the encoding hierarchy, frames 100–132 may be decoded in a different order. Moreover, after being encoded, frames 100–132 may be arranged in decoding order in a bitstream including encoded data for frames 100–132. For example, frame 116 may be displayed last among frames of GOP 134. However, due to the encoding hierarchy, frame 116 may be decoded first of GOP 134. That is, in order to properly decode frame 108, for example, frame 116 may need to be decoded first, in order to act as a reference frame for frame 108. Likewise, frame 108 may act as a reference frame for frames 104, 106, 110, and 112, and therefore may need to be decoded before frames 104, 106, 110, and 112.

[0120] The time at which a frame is displayed may be referred to as presentation time, whereas the time at which the frame is decoded may be referred to as decoding time. Decoding times and presentation times generally provide indications of temporal ordering relative to other frames of the same sequence. The difference between a frame's decoding time and the frame's presentation time may be referred to as picture reordering delay for the frame. As discussed above, a source device (e.g., source device 20) may calculate a maximum picture reordering delay for a set of switchable representations to calculate a delay value to be signaled for the representations.

[0121] In this manner, a destination device, such as destination device 40, may use the signaled delay value to buffer a sufficient amount of data to provide seamless switching between two switchable representations. For example, suppose frames 100–132 are frames of a first representation. Destination device 40 may determine that network conditions have changed and that a second representation is better suited to the new network conditions. Destination device 40 may further determine that a random access point in the second representation exists having the same presentation time as frame 132. Destination device 40 may then retrieve each of frames 100–116 from the first representation, then retrieve frames with presentation times equal to those of frames 118–132 from the second representation. Destination device 40 may also retrieve a frame of the second representation having the same presentation time as frame 116 if the random access point is an open decoding refresh random access point, such that the frames of the second representation corresponding to frames 118–130 can be properly decoded.

[0122] FIG. 4 is a conceptual diagram illustrating switchable representations 154, 160. FIG. 4 illustrates a correspondence between raw video sequence 150 and fragments 158, 164 of representations 154, 160, respectively. In this example, representation 154 includes initialization segment 156 and fragments 158A–158N, and representation 160 includes initialization segment 162 and fragments 164A–164N.

[0123] Initialization segments 156, 162 may include similar or identical data. Accordingly, in some examples, a client device may retrieve initialization segment 156 and decode data of one or more of fragments 164 of representation 160 based on initialization segment 156 of representation 154, without retrieving initialization segment 162 of representation 160. The destination device may therefore switch between fragments 158 of representation 154 and fragments 164 of representation 160, without retrieving both of initialization segments 156, 162.

[0124] As an example, in 3GPP HTTP Streaming, initialization segments contain initialization information for accessing a corresponding representation, but the initialization segments do not contain media data. When streaming video coded with H.264/AVC, several parameters may be present prior to the first IDR picture of the video, such as a Sequence Parameter Set, Picture Parameter Set, and/or some sequence level Supplemental Enhancement Information (SEI) messages. Those parameters may be applicable for the entire representation, thus they may be included in the initialization segment, in accordance with the techniques of this disclosure, rather than including them in a media segment that contains the first IDR when the initialization segment is in use. Accordingly, initialization segments 156, 162 may include a sequence parameter set, a picture parameter set, and/or sequence level SEI messages for representations 154, 160, respectively.

[0125] The example of FIG. 4 illustrates fragments 158 and fragments 164 as having different sizes. This is to illustrate that the amount of memory occupied by fragments 158 is not necessarily the same as fragments 164. That is, fragment 158A, for example, is not the same size, in terms of memory consumption (and likewise bandwidth usage when transporting over a network) as fragment 164A. However, the dashed lines illustrate that fragment 158A and fragment 164A correspond to the same video segment 152A. Although fragments 158A and 164A are different sizes, they correspond to the same temporal region of video sequence 150. In other words, a starting presentation time and an ending presentation time for samples of fragment 158A correspond to starting and ending presentation times for samples of fragment 164.

[0126] In this manner, fragments 158 and 164 may be said to be timely aligned, assuming that random access points occur in the same points in representations 154 and

160. That is, when a random access point in one of fragments 158, the corresponding one of fragments 164 also has a random access point, and the random access points share a decoding order value. Timely alignment of representations may simplify the process of switching between two representations.

[0127] FIG. 5A is a conceptual diagram illustrating an example of two representations 180, 190 that are timely aligned. Representation 180 includes random access points (RAPs) 182A, 182B and samples 184A–184D. Random access points are samples with the additional property that a decoder can decode the random access point without having received any data preceding the random access point in the representation. Accordingly, random access points may generally be pictures that are intra-mode encoded.

[0128] Random access points 182 and samples 184 of representation 180 are illustrated in decoding time order, e.g., as the samples would be arranged in a bitstream sent from a server device to a client device. It should be understood that presentation times of random access points 182 and samples 184 do not necessarily match the decoding times.

[0129] Representation 190 of FIG. 5A includes random access points 192A, 192B and samples 194A–194D. Again, random access points 192 and samples 194 are illustrated in decoding time order, in this example. As can be seen, random access points 182 have corresponding random access points 192. That is, random access point 182A has the same decoding time as random access point 192A, and random access point 182B has the same decoding time as random access point 192B.

[0130] To switch from representation 180 to representation 190, a client device may buffer a certain number of samples (which may include random access points) of representation 180, based on a delay value received from a server device. The client device may then determine when the next random access point of representation 180 occurs, e.g., by analyzing signaled data from the server device that indicates locations, decoding times, and/or presentation times of random access points in representation 180. Suppose, for example, that the next random access point of representation 180 is random access point 182B. The client device may then request data of representation 190 starting at the random access point of representation 190 having the same decoding time as the determined random access point of representation 180. In this example, the client device would begin requesting data of representation 190 starting with random access point 192B, which has the same decoding time as random access point 182B.

[0131] In this manner, the client device may continually decode the data of representation 180 and begin retrieving data of representation 190 while decoding and reordering the buffered data of representation 180. After decoding all of the buffered data of representation 180, the client device may immediately begin decoding and reordering the retrieved data of representation 190. In this manner, the client device may perform seamless switching between two representations. That is, the client device may decode only one representation at a time (thus not requiring two or more decoders to effect the switch), and may also avoid gaps in playback, which may otherwise be caused by not having sufficient data buffered to perform the switch.

[0132] In the example above, representations 180 and 190 are timely aligned, enabling the client device to simply determine a random access point following the buffered data of representation 180, then request data of representation 190 at the random access point having the same decoding time as the determined random access point of representation 180. In other examples, two representations are not timely aligned, yet the techniques of this disclosure may still be used to switch between the representations.

[0133] FIG. 5B is a conceptual diagram illustrating an example of two representations 200, 210 that are not timely aligned. Representation 200 includes random access points 202A, 202B and samples 204A–204D, while representation 210 includes random access points 212A, 212B and samples 214A–214D. Again, random access points and samples in the example of FIG. 5B are illustrated in decoding time order.

[0134] In the example of FIG. 5B, representations 200 and 210 are not timely aligned. Specifically, in this example, random access point 202B does not have a corresponding random access point (that is, a random access point with the same decoding time) in representation 210. Likewise, random access point 212B does not have a corresponding random access point in representation 200. In general, two representations need not have the same number of random access points, and the random access points need not be timely aligned. Nevertheless, the techniques of this disclosure may still be applied to perform switching between two representations that are not timely aligned.

[0135] To switch between representation 200 and representation 210, for example, a client device may buffer a certain number of samples (which may include random access points) of representation 200, based on a delay value received from a server device. The server device may calculate the delay value as the sum of a maximum decoding delay, a maximum picture reordering delay, and (because representations 200 and 210 are not timely aligned), an overlap delay. Therefore, by buffering data based on the delay value, the client device may buffer a sufficient amount of data to ensure seamless switching between representations 200 and 210, which may account for decoding delay, picture reordering delay, and delay associated with the representations not being timely aligned.

[0136] The buffered data of representation 200 may include a fragment overlapping random access point 212B of representation 210. For example, the fragment may include random access point 202B and samples 204C and 204D (and potentially subsequent samples of representation 200 not illustrated in FIG. 5B). The fragment may "overlap" random access point 212B in the sense that the earliest presentation time of the samples in the fragment may be earlier than the presentation time of random access point 212B, but the latest presentation time of the samples in the fragment may be later than the presentation time of random access point 212B. Such a fragment is said to overlap the random access point. This disclosure may also refer to such a fragment as "spanning" the random access point. The fragment may correspond to the last buffered fragment of representation 200. That is, the client device may request data from representation 200 up to and including the fragment that overlaps the random access point of representation 210.

[0137] The client device may also locate a fragment of representation 210 including random access point 212B. The client device may request data from representation 210 starting at the fragment including random access point 212B. The client device may decode the data retrieved from representation 200 up to a group of pictures in the fragment overlapping random access point 212B. Upon reaching this group of pictures, the client device may begin decoding the data of representation 210. The client device may determine whether there are any pictures in representation 200 that have a presentation time earlier than the presentation time of random access point 212B, and decode any such pictures before decoding random access point 212B. In this manner, the client device may seamlessly switch between representations 200 and 210. That is, the client device may avoid gaps in video playback, while also avoiding parallel decoding, that is, operating two or more decoders at once to decode the representations.

[0138] Although in this example a certain amount of data may be retrieved that is not ultimately decoded, this retrieval of a small amount of extra data is outweighed by the benefits of not needing to perform parallel decoding of representations and also allows for seamless playback. Accordingly, the cost of retrieving the extra data may be considered negligible.

[0139] FIG. 6 is a flowchart illustrating an example method for performing seamless switching between two switchable representations. Although described with respect to source device 20 and destination device 40 of FIG. 1, it should be understood that the techniques of FIG. 6 may be performed by any server and client devices. Moreover, the steps in the method need not necessarily occur in the precise order illustrated in FIG. 6, and additional steps may be added or some steps may be removed without departing from the techniques of this disclosure.

[0140] Initially, source device 20 may provide data describing two or more switchable representations (250) to destination device 40. This data may describe, for example, minimum network bandwidth requirements for smooth playback, profile and/or level requirements of a decoder device to decode and/or render encoded video data of the representations, locations (e.g., fragments and/or byte ranges) of random access points, decoding times of random access points, presentation times of random access points, decoding and/or presentation times of fragments of the representations, byte ranges of the fragments, uniform resource locators including the random access points, or other such data for decoding and/or rendering encoded video data.

[0141] The data may further indicate whether the two or more representations (or some subset thereof) form an efficient switch group, and whether any of the two or more representations are timely aligned. Furthermore, the data may include a delay value for switching between switchable representations. The delay value may correspond to the sum of a maximum decoding delay and a maximum picture reordering delay for the switchable representations. In addition, for representations that are not timely aligned, the delay value may account for an overlap delay.

[0142] Destination device 40 may receive the data describing the two or more representations (252). Destination device 40 may then determine current network conditions (254), e.g., available bandwidth, to initially select one of the representations. In addition, destination device 40 may determine its own decoding and/or rendering capabilities to select between the representations. That is, of the representations that destination device 40 is capable of decoding and rendering, destination device 40 may select one of the representations based on the current network conditions. For example, destination device 40 may select a first representation having a bandwidth requirement closest to but not exceeding the current available network bandwidth.

[0143] Destination device 40 may then request data from the first representation (256). For example, destination device 40 may generate an HTTP GET or partial GET request specifying a fragment of the first representation, and potentially a byte range of the fragment. In response to the request, source device 20 may send encoded samples of the first representation requested by destination device 40 to destination device 40 (258). Destination device 40 may decode and display the received samples (260). Assuming no change in the network conditions, destination device 40 may continue to request samples of the first representation, then decode and display the samples after they are received.

[0144] In the example of FIG. 6, destination device 40 may determine a change in network conditions (262). For example, destination device 40 may determine that available network bandwidth has increased or decreased. In response to determining a bandwidth increase, destination device 40 may select a second representation having a higher bandwidth requirement and a corresponding higher quality. On the other hand, in response to determining a bandwidth decrease, destination device 40 may select a second representation having a lower bandwidth requirement, albeit with a lower quality.

[0145] After determining that the network conditions have changed, destination device 40 may buffer an amount of data of the first representation based on the delay value received from source device 20 (264). For example, destination device 40 may determine a number of samples or a number of fragments to retrieve from the first representation based on the delay value to allow for seamless switching from the first representation to the second representation. Buffering the data may include generating one or more requests for data from the first representation and sending the requests to source device 20. After buffering this data from the first representation, destination device 40 may begin requesting data from the selected second representation (266).

[0146] In response to the request, source device 20 may send encoded samples of the second representation to destination device 40 (268). During this time, destination device 40 may continue decoding the buffered data of the first representation. After receiving the requested data from the second representation, and after decoding samples of the first representation having a presentation time before a random access point of the second representation, destination device 40 may begin decoding and displaying received samples of the second representation (270). In some examples, destination device 40 may request data from the second representation without requesting an initialization segment for the second representation. Accordingly, destination device 40 may use an initialization segment retrieved from the first representation to decode and display the samples of the second representation.

[0147] FIG. 7 is a flowchart illustrating an example method for a server device, such as source device 20, to provide information describing switchable representations to a client device, such as destination device 40. Although described with respect to source device 20 and destination device 40 of FIG. 1, it should be understood that the techniques of FIG. 7 may be performed by any server and client devices. Moreover, the steps in the method need not necessarily occur in the precise order illustrated in FIG. 7, and additional steps may be added or some steps may be removed without departing from the techniques of this disclosure.

[0148] Initially, source device 20 may receive encoded video samples of two or more switchable representations (280). In some examples, source device 20 may receive raw video data and encode two or more switchable representations. In other examples, source device 20 may receive encoded representations from an intermediate device including an encoder, or multiple encoders, that produce the switchable representations. The representations may be timely aligned, in some examples, or in other examples, may not be timely aligned.

[0149] Source device 20 may determine a maximum decoding delay of the received samples of the switchable representations (282). For example, source device 20 may analyze the samples of each representation to determine a maximum decoding delay for each representation. Then, based on these determined maxima, source device 20 may calculate the maximum decoding delay of all of the representations. For example, suppose source device 20 receives samples for three representations. Source device 20 may determine a first maximum decoding delay for the first representation, a second maximum decoding delay for the second representation, and a third maximum decoding delay for the third representation. Source device 20 may then calculate the maximum of the first, second, and third maximum decoding delays as the maximum decoding delay of the switchable representations. The samples may comprise access units of the representations, such that the maximum decoding delay corresponds to the maximum delay associated with decoding any access unit of any of the switchable representations.

[0150] Source device 20 may also determine a maximum reordering delay of the samples (284). Again, source device 20 may analyze the samples for the switchable representations to calculate maximum reordering delays for each representation, then calculate the maximum of these to determine the maximum reordering delay associated with the switchable representations. In general, a picture reordering delay is the difference between the presentation time and decoding time for a frame of a representation. Accordingly, the maximum picture reordering delay may correspond to the maximum difference between the presentation time and decoding time of any frame of any of the switchable representations.

[0151] Source device 20 may then calculate a delay value associated with seamless switching between the switchable representations (286), e.g., as the sum of the maximum decoding delay and the maximum picture reordering delay. Source device 20 may further determine whether the representations are timely aligned (288). For example, source device 20 may determine that the representations are timely aligned only when, for any random access point in one of the representations, each of the other representations have a random access point having the same decoding times as the random access point of the one of the representations. Source device 20 may receive an indication from a decoding device as to whether the representations are timely aligned, or when source device 20 encodes the representations, may keep track of whether the representations are generated to be timely aligned.

[0152] If the representations are timely aligned (“YES” branch of 288), source device 20 may simply store the calculated delay value and encoded video samples (294). On the other hand, if the representations are not timely aligned (“NO” branch of 288), source device 20 may treat the delay value calculated at 286 as an intermediate delay value. Source device 20 may then calculate a maximum overlap delay for the switchable representations (290). The overlap delay of a picture may be based on a GOP structure of the current representation, as well as the location of the picture in the GOP. Source device 20 may calculate the maximum overlap delay for the switchable representations, then add the maximum overlap delay to the delay value calculated at step 286 (292). Source device 20 may then store this calculated delay value along with the video samples of the switchable representations (294). Source device 20 may, for example, generate signaling information that source device 20 may provide to destination device 40 in response to a request for such signaling information.

[0153] FIG. 8 is a flowchart illustrating an example method for a client device, such as destination device 40, to switch from retrieving data of a first representation from a server device, such as source device 20, to retrieving data of a second representation from the server device. Although described with respect to source device 20 and destination device 40 of FIG. 1, it should be understood that the techniques of FIG. 8 may be performed by any server and client devices. Moreover, the steps in the method need not necessarily occur in the precise order illustrated in FIG. 8, and additional steps may be added or some steps may be removed without departing from the techniques of this disclosure. Moreover, the example of FIG. 8 presumes that the two representations are not timely aligned.

[0154] Initially, destination device 40 may receive a delay value for two or more switchable representations (300), e.g., from source device 20. Destination device 40 may receive additional information describing the switchable representations, such as locations of random access points, whether the representations are timely aligned, profile and/or level indicators that describe decoding and/or rendering requirements of client devices to successfully decode/render the representations, bandwidth limitations of the representations, or other such information.

[0155] Destination device 40 may receive, decode, and display data of a first representation (302). For example, destination device 40 may select the first representation based on current network conditions such as available bandwidth and decoding and rendering capabilities of destination device 40. Destination device 40 may then submit one or more requests for data from the first representation and begin decoding and displaying the data of the first representation after buffering a sufficient amount of data, based on the received delay value.

[0156] At some point, destination device 40 may determine that there has been a change in the network conditions (304), e.g., that available bandwidth has either increased or decreased. Accordingly, based on the new network conditions, destination device 40 may select a second representation that is switchable to the first representation (306). As noted above, it is assumed that the first representation and the second representation are not timely aligned in this example.

[0157] Destination device 40 may determine a fragment, labeled "fragment M" in this example, that includes a next random access point (RAP) in the second representation (308). That is, destination device 40 may determine a random access point of the second representation that has a presentation time later than the presentation times of the buffered data for the first representation. Destination device 40 may then determine fragment M, which includes the random access point.

[0158] Destination device 40 may then locate a fragment, labeled "fragment N" in this example, in the first representation that spans the random access point of the second representation (310). That is, destination device 40 may identify fragment N having an earliest presentation time that is earlier than the presentation time of the random access point of the second representation, and having a latest presentation time that is later than the presentation time of the random access point of the second representation.

[0159] Destination device 40 may request data from the first representation up to and including fragment N (312). After requesting fragment N, destination device 40 may begin requesting data from the second representation, starting with fragment M (314). Destination device 40 may decode all samples of the first representation up to a group of pictures that spans the presentation time of the random access point of fragment M (316).

[0160] Destination device 40 may ensure that all pictures having a presentation time earlier than the presentation time of the random access point in fragment M have been decoded and displayed, which may include pictures within the group of pictures mentioned above. After this, destination device 40 may begin decoding pictures of the second representation starting at the random access point (318). Destination device 40 may further display the decoded pictures of the second representation, which may require some amount of picture reordering.

[0161] In addition, destination device 40 may request and decode fragments of the second representation without requesting an initialization segment specific to the second representation. For example, when the received information indicates that the first and second representations are efficiently switchable, destination device 40 may decode and render the samples of the second representation based on an initialization segment received for the first representation. Accordingly, the method of FIG. 8 may further include requesting and receiving an initialization segment of the first representation prior to receiving, decoding, and displaying data of the first representation (e.g., prior to step 302).

[0162] 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 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. The phrase "one or more processors" may include one or more cores of a multi-core processor, each of which may be included within a common chip (e.g., a single integrated circuit). A computer program product may include a computer-readable medium.

[0163] 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 transient media, but are instead directed to non-transient, 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.

[0164] 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.

[0165] 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.

[0166] Various examples have been described. These and other examples are within the scope of the following claims.

CLAIMS:

1. A method of sending video data, the method comprising:

providing information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence;

providing information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch;

in response to a first request from the client device to retrieve data from the first representation, sending the requested data from the first representation to the client device; and

in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, sending the requested data from the second representation to the client device.

2. The method of claim 1, further comprising:

determining a maximum decoding delay for coded video samples of the first and second representations, wherein a decoding delay of a coded video sample comprises delay for decoding the coded video sample;

determining a maximum picture reordering delay for the coded video samples of the first and second representations, wherein a picture reordering delay for a coded video picture comprises the difference between a decoding time for the coded video picture and a presentation time for the coded video picture; and

calculating a delay value comprising the sum of the maximum decoding delay and the maximum picture reordering delay,

wherein providing the information indicating that the client device can switch to the second representation comprises sending, to the client device, the delay value for switching from the first representation to the second representation.

3. The method of claim 2, further comprising:

determining whether the coded video samples of the first and second representations are time-aligned; and

when the coded video samples of the first and second representations are not time-aligned, wherein calculating the delay value further comprises adding a time-alignment delay value to the sum of the maximum decoding delay and the maximum picture reordering delay, wherein the time-alignment delay value comprises a temporal difference between a last coded video sample of a last requested fragment of the first representation and a first coded video sample of a first fragment of the second representation.

4. The method of claim 1, further comprising:

determining a maximum size value of a random access point picture for coded video samples of the representations; and

sending the maximum size value to the client device.

5. The method of claim 1, further comprising:

receiving an initialization segment for the first representation;

determining that the second representation can be decoded based on the initialization segment of the first representation;

assigning a group identifier value to the first representation and to the second representation; and

sending the group identifier value to the client device, wherein sending the requested data from the first representation comprises sending the initialization segment to the client device; and

wherein sending the requested data of the second representation to the client device comprises sending the requested data of the second representation to the client device without sending an initialization segment of the second representation to the client device.

6. An apparatus for transferring video data, the apparatus comprising a processor configured to provide information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence, provide information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch, in response to a first request from the client device to retrieve data from the first representation, send the requested data from the first representation to the client device, and, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, send the requested data from the second representation to the client device.

7. The apparatus of claim 6, wherein the processor is configured to determine a maximum decoding delay for the coded video samples of the first and second representations, wherein a decoding delay of a coded video sample comprises delay for decoding the coded video sample, determine a maximum picture reordering delay for the coded video samples of the first and second representations, wherein a picture reordering delay for a coded video picture comprises the difference between a decoding time for the coded video picture and a presentation time for the coded video picture, calculate a delay value comprising the sum of the maximum decoding delay and the maximum picture reordering delay, and send, to the client device, the delay value for switching from the first representation to the second representation.

8. The apparatus of claim 7, wherein the processor is configured to determine whether the coded video samples of the first and second representations are time-aligned, and, when the coded video samples of the first and second representations are not time-aligned, wherein calculating the delay value further comprises adding a time-alignment delay value to the sum of the maximum decoding delay and the maximum picture reordering delay, wherein the time-alignment delay value comprises a temporal difference between a last coded video sample of a last requested fragment of the first representation and a first coded video sample of a first fragment of the second representation.

9. The apparatus of claim 6, wherein the processor is configured to determine a maximum size value of a random access point picture for coded video samples of the representations, and send the maximum size value to the client device.

10. The apparatus of claim 6, wherein the processor is configured to receive an initialization segment for the first representation, determine that the second representation can be decoded based on the initialization segment of the first representation, assign a group identifier value to the first representation and to the second representation, and send the group identifier value to the client device, send the initialization segment to the client device with the requested data from the first representation, and send the requested data of the second representation to the client device without sending an initialization segment of the second representation to the client device.

11. The apparatus of claim 6, wherein the apparatus comprises at least one of:
an integrated circuit;
a microprocessor; and
a wireless communication device that includes the processor.

12. An apparatus for providing video data, the apparatus comprising:

means for providing information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence;

means for providing information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch;

means for, in response to a first request from the client device to retrieve data from the first representation, sending the requested data from the first representation to the client device; and means for, in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, sending the requested data from the second representation to the client device.

13. The apparatus of claim 12, further comprising:

means for determining a maximum decoding delay for the coded video samples of the first and second representations, wherein a decoding delay of a coded video sample comprises delay for decoding the coded video sample;

means for determining a maximum picture reordering delay for the coded video samples of the first and second representations, wherein a picture reordering delay for a coded video picture comprises the difference between a decoding time for the coded video picture and a presentation time for the coded video picture;

means for calculating a delay value comprising the sum of the maximum decoding delay and the maximum picture reordering delay; and

means for sending, to the client device, the delay value for switching from the first representation to the second representation.

14. The apparatus of claim 13, further comprising:

means for determining whether the coded video samples of the first and second representations are time-aligned; and

wherein the means for calculating the delay value further comprises means for adding a time-alignment delay value to the sum of the maximum decoding delay and the maximum picture reordering delay when the coded video samples of the first and second representations are not time-aligned, wherein the time-alignment delay value comprises a temporal difference between a last coded video sample of a last requested fragment of the first representation and a first coded video sample of a first fragment of the second representation.

15. The apparatus of claim 12, further comprising:

means for determining a maximum size value of a random access point picture for coded video samples of the representations; and

means for sending the maximum size value to the client device.

16. The apparatus of claim 12, further comprising:

means for receiving an initialization segment for the first representation;

means for determining that the second representation can be decoded based on the initialization segment of the first representation;

means for assigning a group identifier value to the first representation and to the second representation; and

means for sending the group identifier value to the client device,

wherein the means for sending the requested data from the first representation comprises means for sending the initialization segment to the client device, and

wherein the means for sending the requested data from the second representation comprises means for sending data of the second representation to the client device without sending an initialization segment of the second representation to the client device.

17. A computer program product comprising a computer-readable storage medium having stored thereon instructions that, when executed, cause a processor of a device for providing video data to:

provide information to a client device for accessing data for a first representation of a video sequence and a second representation of the video sequence;

provide information to the client device indicating that the client device can, during transmission of the data for the first representation, switch to the second representation without experiencing a presentation gap when displaying the video sequence and without simultaneously executing more than one decoder to decode the data for the first representation and the second representation during the switch;

in response to a first request from the client device to retrieve data from the first representation, send the requested data from the first representation to the client device; and

in response to a second request from the client device to retrieve data from the second representation after sending the requested data from the first representation, send the requested data from the second representation to the client device.

18. The computer program product of claim 17, further comprising instructions that cause the processor to:

determine a maximum decoding delay for coded video samples of the first and second representations, wherein a decoding delay of a coded video sample comprises delay for decoding the coded video sample;

determine a maximum picture reordering delay for the coded video samples of the first and second representations, wherein a picture reordering delay for a coded video picture comprises the difference between a decoding time for the coded video picture and a presentation time for the coded video picture;

calculate a delay value comprising the sum of the maximum decoding delay and the maximum picture reordering delay; and

send, to the client device, the delay value for switching from the first representation to the second representation.

19. The computer program product of claim 18, further comprising instructions that cause the processor to:

determine whether the coded video samples of the first and second representations are time-aligned; and

wherein the instructions that cause the processor to calculate the delay value comprise that cause the processor to, when the coded video samples of the first and second representations are not time-aligned, add a time-alignment delay value to the sum of the maximum decoding delay and the maximum picture reordering delay, wherein the time-alignment delay value comprises a temporal difference between a last coded video sample of a last requested fragment of the first representation and a first coded video sample of a first fragment of the second representation.

20. The computer program product of claim 17, further comprising instructions that cause the processor to:

determine a maximum size value of a random access point picture for coded video samples of the representations; and

send the maximum size value to the client device.

21. The computer program product of claim 17, further comprising instructions that cause the processor to:

receive an initialization segment for the first representation;

determine that the second representation can be decoded based on the initialization segment of the first representation;

assign a group identifier value to the first representation and to the second representation; and

send the group identifier value to the client device,

wherein the instructions that cause the processor to send the data from the first representation comprise instructions that cause the processor to send the initialization segment to the client device, and

wherein the instructions that cause the processor to send the data from the second representation comprises instructions that cause the processor to send data from the second representation to the client device without sending an initialization segment of the second representation to the client device.

22. A method of receiving video data, the method comprising:

receiving a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence;

requesting data from the first representation of a video sequence in accordance with a streaming network protocol;

in response to determining to switch from the first representation to the second representation, buffering an amount of data from the first representation based on the delay value; and

after buffering the amount of data from the first representation, requesting data from a second representation of the video sequence in accordance with the streaming network protocol.

23. The method of claim 22, further comprising:

requesting an initialization segment of the first representation prior to requesting the data of the first representation,

wherein requesting the data from the second representation comprises requesting the data from the second representation without requesting an initialization segment of the second representation.

24. The method of claim 22, further comprising:

detecting a change in network conditions; and

based on the change in the network conditions, determining to switch from the first representation to the second network representation.

25. The method of claim 22, further comprising:

receiving information indicating whether the first representation and the second representation are timely aligned; and

selecting a random access point of the second representation based on the information.

26. The method of claim 25, wherein when the first representation and the second representation are timely aligned, selecting the random access point comprises:

locating a first random access point of the first representation having an earliest presentation time value following presentation time values of samples of the buffered data of the first representation; and

selecting a second random access point of the second representation having the same presentation time value as the first random access point,

wherein requesting the data from the second representation comprises requesting a fragment of the second representation comprising the second random access point.

27. The method of claim 25, wherein when the first representation and the second representation are not timely aligned, the method comprises:

selecting the random access point comprises selecting a random access point of the second representation having a presentation time value later than a presentation time value of at least one sample of the buffered data of the first representation,

wherein buffering the amount of data comprises requesting and storing data of the first representation up to a first fragment having a starting presentation time value earlier than the presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and

wherein requesting data from the second representation comprises requesting data of the second representation starting at a second fragment of the second representation comprising the selected random access point.

28. The method of claim 27, further comprising:

decoding the data of the first representation until reaching a group of pictures of the first fragment, the group of pictures comprising a first picture having a starting presentation time value earlier than a presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and

after reaching the group of pictures of the first fragment, decoding the data of the second representation starting at the selected random access point.

29. An apparatus for receiving video data, the apparatus comprising a processor configured to receive a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence, request data from the first representation of a video sequence in accordance with a streaming network protocol, in response to determining to switch from the first representation to the second representation, buffer an amount of data from the first representation based on the delay value, and, after buffering the amount of data from the first representation, request data from a second representation of the video sequence in accordance with the streaming network protocol.

30. The apparatus of claim 29, wherein the processor is configured to request an initialization segment of the first representation prior to requesting the data of the first representation, wherein the processor is configured to request the data from the second representation without requesting an initialization segment of the second representation.

31. The apparatus of claim 29, wherein the processor is configured to detect a change in network conditions, and, based on the change in the network conditions, determine to switch from the first representation to the second network representation.

32. The apparatus of claim 29, wherein the processor is configured to receive information indicating whether the first representation and the second representation are timely aligned, and select a random access point of the second representation based on the information.

33. The apparatus of claim 32, wherein to select the random access point when the first representation and the second representation are timely aligned, the processor is configured to locate a first random access point of the first representation having an earliest presentation time value following presentation time values of samples of the buffered data of the first representation, and select a second random access point of the second representation having the same presentation time value as the first random access point, wherein to request the data from the second representation, the processor is configured to request a fragment of the second representation comprising the second random access point.

34. The apparatus of claim 32, wherein the processor is configured to, when the first representation and the second representation are not timely aligned, select the random access point comprises selecting a random access point of the second representation having a presentation time value later than a presentation time value of at least one sample of the buffered data of the first representation, wherein to buffer the amount of data, the processor is configured to request and store data of the first representation up to a first fragment having a starting presentation time value earlier than the presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and to request data from the second representation, the processor is configured to request data of the second representation starting at a second fragment of the second representation comprising the selected random access point.

35. The apparatus of claim 34, further comprising a video decoder configured to decode the data of the first representation until reaching a group of pictures of the first fragment, the group of pictures comprising a first picture having a starting presentation time value earlier than a presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and, after reaching the group of pictures of the first fragment, decode the data of the second representation starting at the selected random access point.

36. The apparatus of claim 29, wherein the apparatus comprises at least one of:

an integrated circuit;

a microprocessor; and

a wireless communication device that includes the processor.

37. An apparatus for receiving video data, the apparatus comprising:

means for receiving a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence;

means for requesting data from the first representation of a video sequence in accordance with a streaming network protocol;

means for in response to determining to switch from the first representation to the second representation, buffering an amount of data from the first representation based on the delay value; and

means for requesting, after buffering the amount of data from the first representation, data from a second representation of the video sequence in accordance with the streaming network protocol.

38. The apparatus of claim 37, further comprising:

means for requesting an initialization segment of the first representation prior to requesting the data of the first representation,

wherein the means for requesting the data from the second representation comprises means for requesting the data from the second representation without requesting an initialization segment of the second representation.

39. The apparatus of claim 37, further comprising:

means for detecting a change in network conditions; and

means for determining, based on the change in the network conditions, to switch from the first representation to the second network representation.

40. The apparatus of claim 37, further comprising:

means for receiving information indicating whether the first representation and the second representation are timely aligned; and means for selecting a random access point of the second representation based on the information.

41. The apparatus of claim 40, wherein when the first representation and the second representation are timely aligned, the means for selecting the random access point comprises:

means for locating a first random access point of the first representation having an earliest presentation time value following presentation time values of samples of the buffered data of the first representation; and

means for selecting a second random access point of the second representation having the same presentation time value as the first random access point,

wherein the means for requesting the data from the second representation comprises means for requesting a fragment of the second representation comprising the second random access point.

42. The apparatus of claim 40, wherein when the first representation and the second representation are not timely aligned:

the means for selecting the random access point comprises means for selecting a random access point of the second representation having a presentation time value later than a presentation time value of at least one sample of the buffered data of the first representation,

the means for buffering the amount of data comprises means for requesting and storing data of the first representation up to a first fragment having a starting presentation time value earlier than the presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and

the means for requesting data from the second representation comprises means for requesting data of the second representation starting at a second fragment of the second representation comprising the selected random access point.

43. The apparatus of claim 42, further comprising:

means for decoding the data of the first representation until reaching a group of pictures of the first fragment, the group of pictures comprising a first picture having a starting presentation time value earlier than a presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and

means for decoding, after reaching the group of pictures of the first fragment, the data of the second representation starting at the selected random access point.

44. A computer program product comprising a computer-readable storage medium having stored thereon instructions that, when executed, cause a processor of a device for receiving video data to:

receive a delay value indicative of a delay for switching between a first representation of a video sequence and a second representation of the video sequence;

request data from the first representation of a video sequence in accordance with a streaming network protocol;

in response to determining to switch from the first representation to the second representation, buffer an amount of data from the first representation based on the delay value; and

after buffering the amount of data from the first representation, request data from a second representation of the video sequence in accordance with the streaming network protocol.

45. The computer program product of claim 44, further comprising instructions that cause the processor to request an initialization segment of the first representation prior to requesting the data of the first representation,

wherein the means for requesting the data from the second representation comprises means for requesting the data from the second representation without requesting an initialization segment of the second representation.

46. The computer program product of claim 44, further comprising instructions that cause the processor to:

detect a change in network conditions; and based on the change in the network conditions, determine to switch from the first representation to the second network representation.

47. The computer program product of claim 44, further comprising instructions that cause the processor to:

receive information indicating whether the first representation and the second representation are timely aligned; and

select a random access point of the second representation based on the information.

48. The computer program product of claim 47, wherein when the first representation and the second representation are timely aligned, the instructions that cause the processor to selecting the random access point comprises:

locate a first random access point of the first representation having an earliest presentation time value following presentation time values of samples of the buffered data of the first representation; and

selecting a second random access point of the second representation having the same presentation time value as the first random access point,

wherein requesting the data from the second representation comprises requesting a fragment of the second representation comprising the second random access point.

49. The computer program product of claim 47, further comprising that cause the processor to, when the first representation and the second representation are not timely aligned:

select the random access point comprises selecting a random access point of the second representation having a presentation time value later than a presentation time value of at least one sample of the buffered data of the first representation,

wherein the instructions that cause the processor to buffer the amount of data comprise that cause the processor to request and store data of the first representation up to a first fragment having a starting presentation time value earlier than the presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point, and

wherein the instructions that cause the processor to request data from the second representation comprise instructions that cause the processor to request data of the second representation starting at a second fragment of the second representation comprising the selected random access point.

50. The computer program product of claim 49, further comprising instructions that cause the processor to:

decode the data of the first representation until reaching a group of pictures of the first fragment, the group of pictures comprising a first picture having a starting presentation time value earlier than a presentation time value of the selected random access point and an ending presentation time value later than the presentation time value of the selected random access point; and

after reaching the group of pictures of the first fragment, decode the data of the second representation starting at the selected random access point.

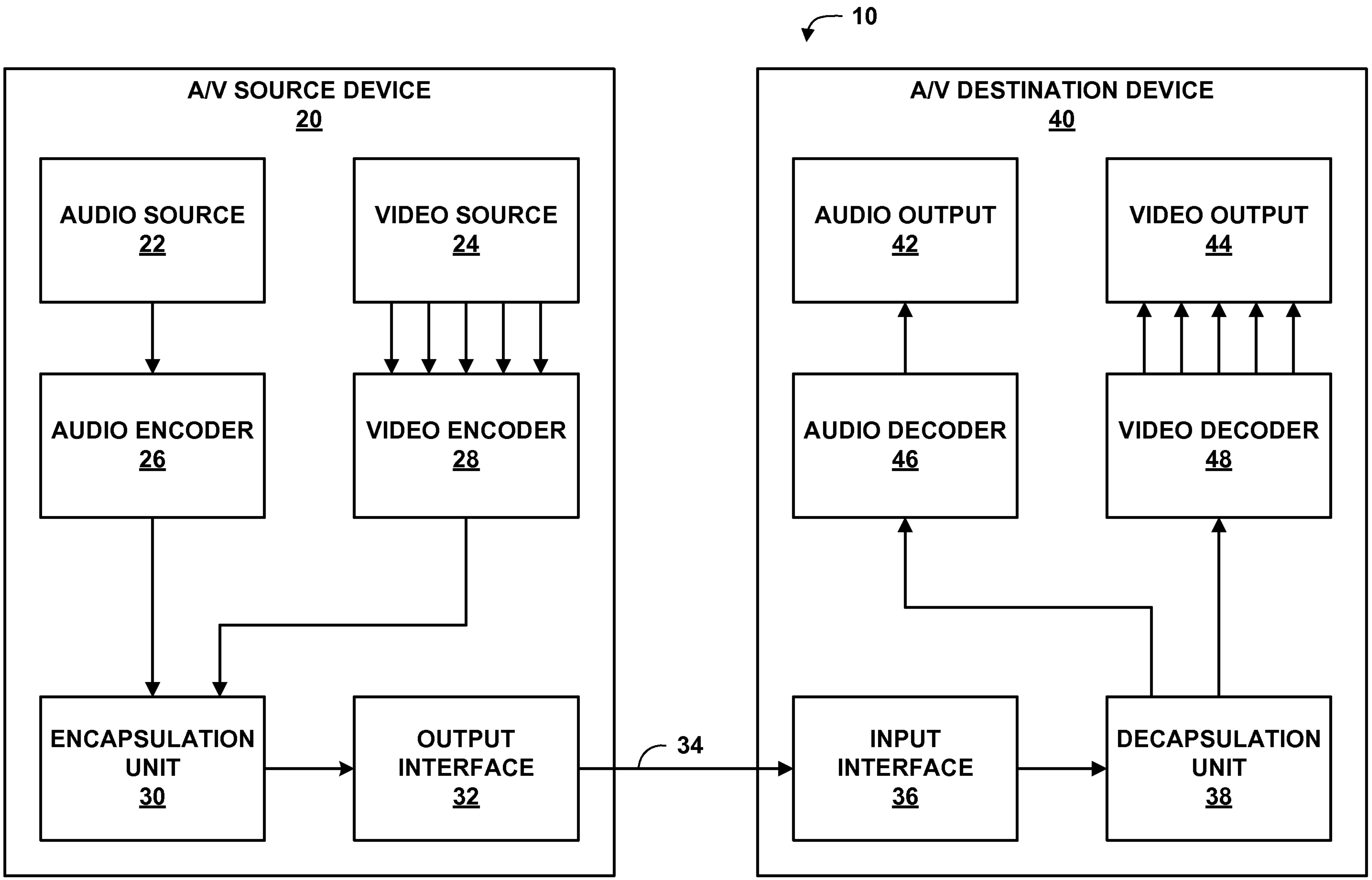
10
A/V SOURCE DEVICE
20
AUDIO SOURCE
22
VIDEO SOURCE
24
AUDIO ENCODER
26
VIDEO ENCODER
28
ENCAPSULATION UNIT
30
OUTPUT INTERFACE
32
34
A/V DESTINATION DEVICE
40
AUDIO OUTPUT
42
VIDEO OUTPUT
44
AUDIO DECODER
46
VIDEO DECODER
48
INPUT INTERFACE
36
DECAPSULATION UNIT
38

FIG. 1

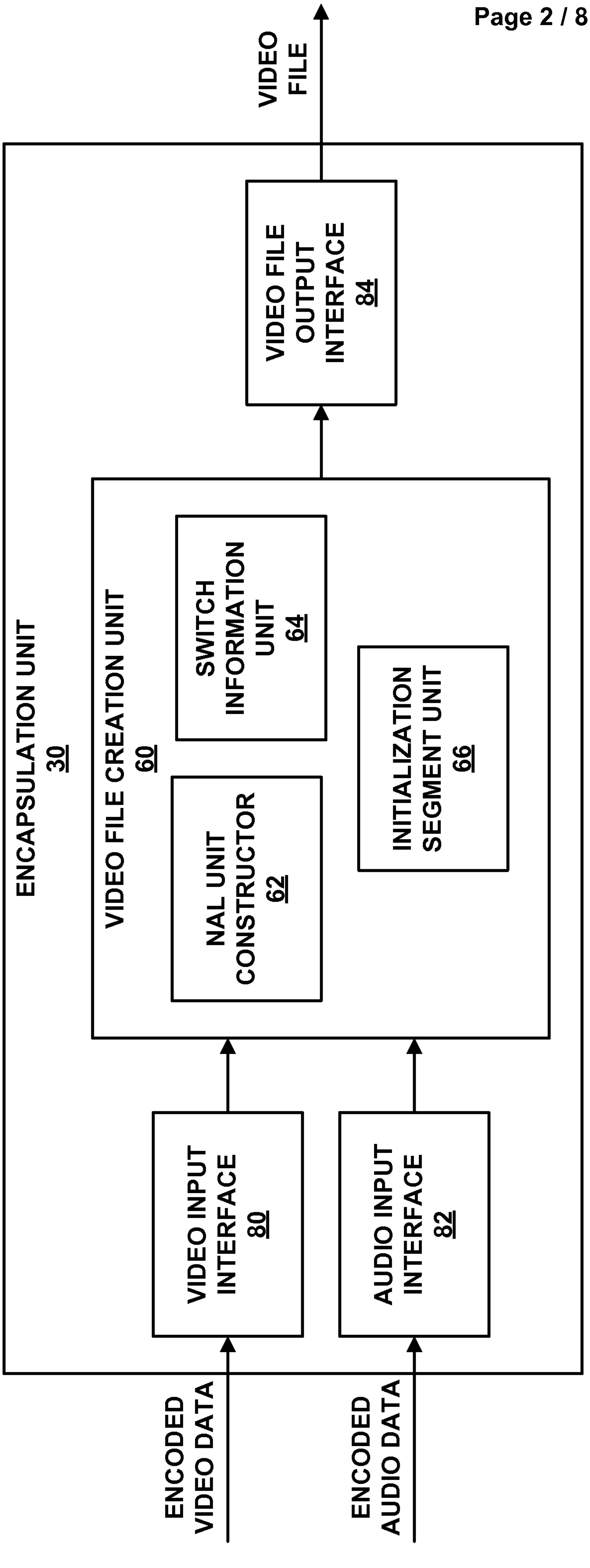
ENCODED VIDEO DATA
ENCODED AUDIO DATA
ENCAPSULATION UNIT 30
VIDEO INPUT INTERFACE 80
AUDIO INPUT INTERFACE 82
VIDEO FILE CREATION UNIT 60
NAL UNIT CONSTRUCTOR 62
SWITCH INFORMATION UNIT 64
INITIALIZATION SEGMENT UNIT 66
VIDEO FILE OUTPUT INTERFACE 84
VIDEO FILE

FIG. 2

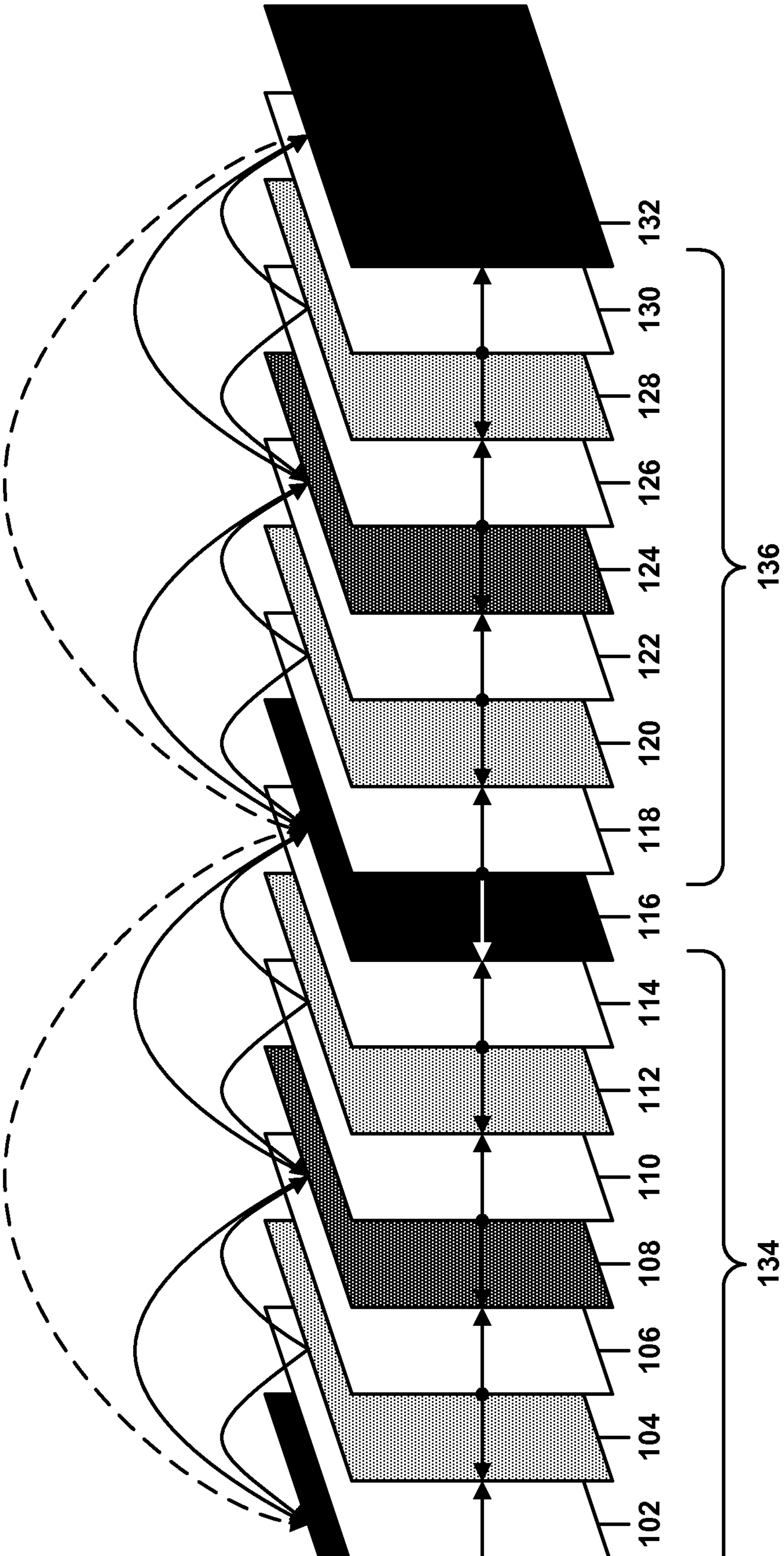
100
102
104
106
108
110
112
114
116
118
120
122
124
126
128
130
132
134
136

FIG. 3

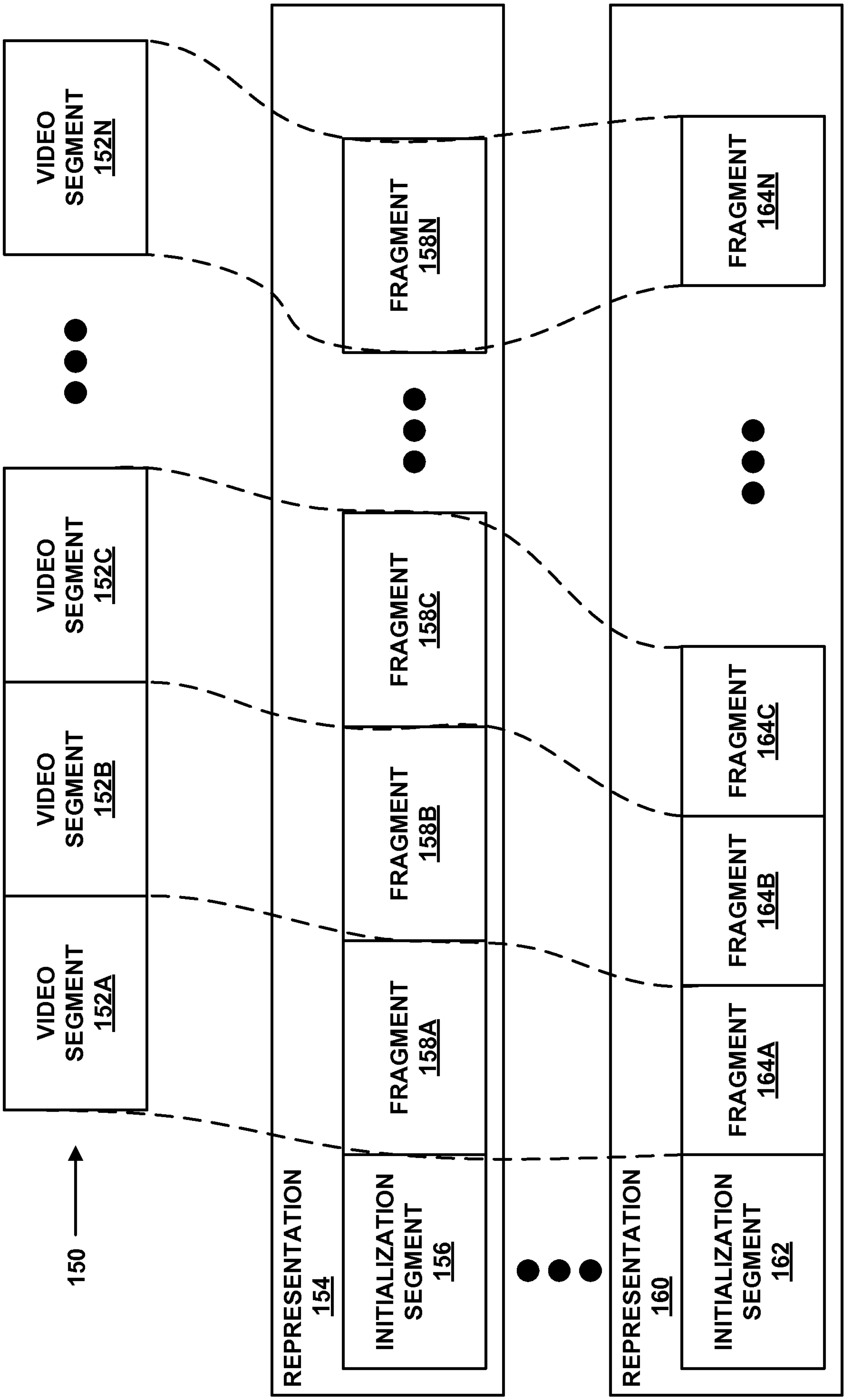
150
VIDEO SEGMENT 152A
VIDEO SEGMENT 152B
VIDEO SEGMENT 152C
VIDEO SEGMENT 152N
REPRESENTATION 154
INITIALIZATION SEGMENT 156
FRAGMENT 158A
FRAGMENT 158B
FRAGMENT 158C
FRAGMENT 158N
REPRESENTATION 160
INITIALIZATION SEGMENT 162
FRAGMENT 164A
FRAGMENT 164B
FRAGMENT 164C
FRAGMENT 164N

FIG. 4

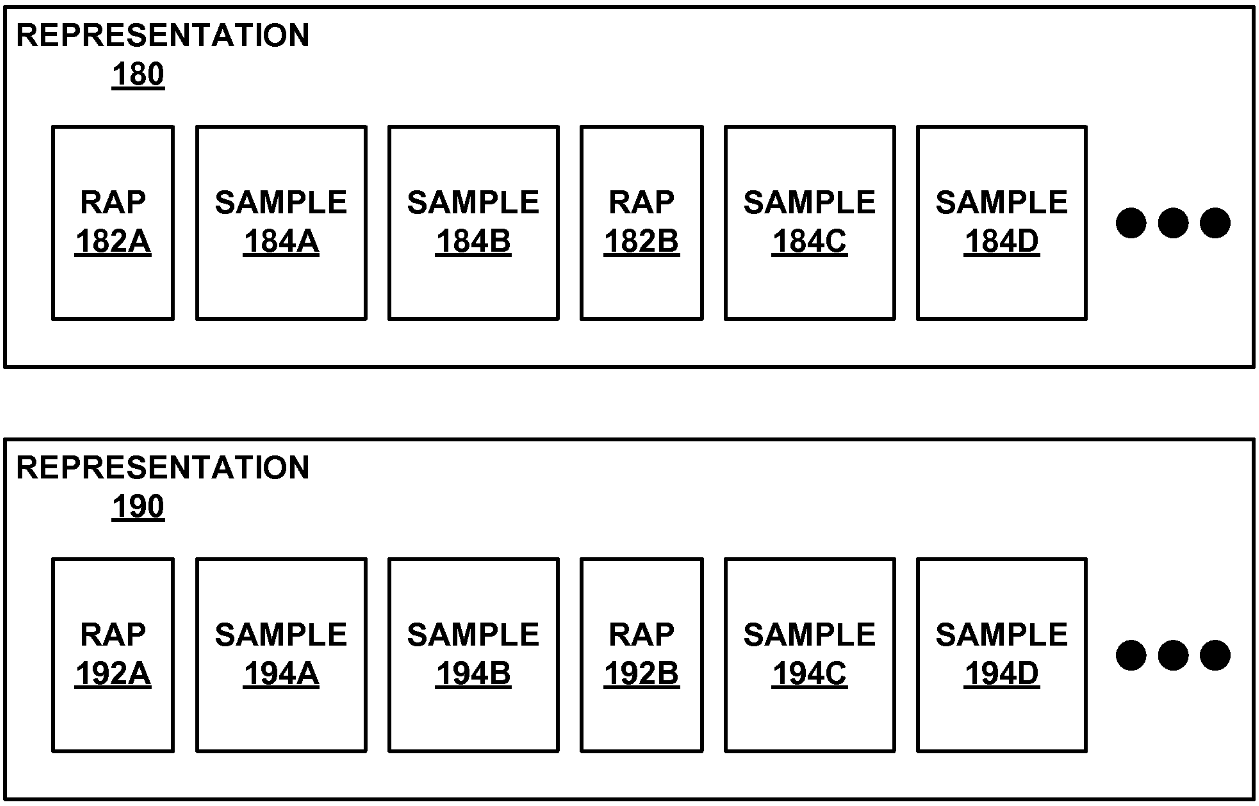
REPRESENTATION 180
RAP 182A
SAMPLE 184A
SAMPLE 184B
RAP 182B
SAMPLE 184C
SAMPLE 184D
REPRESENTATION 190
RAP 192A
SAMPLE 194A
SAMPLE 194B
RAP 192B
SAMPLE 194C
SAMPLE 194D

FIG. 5A

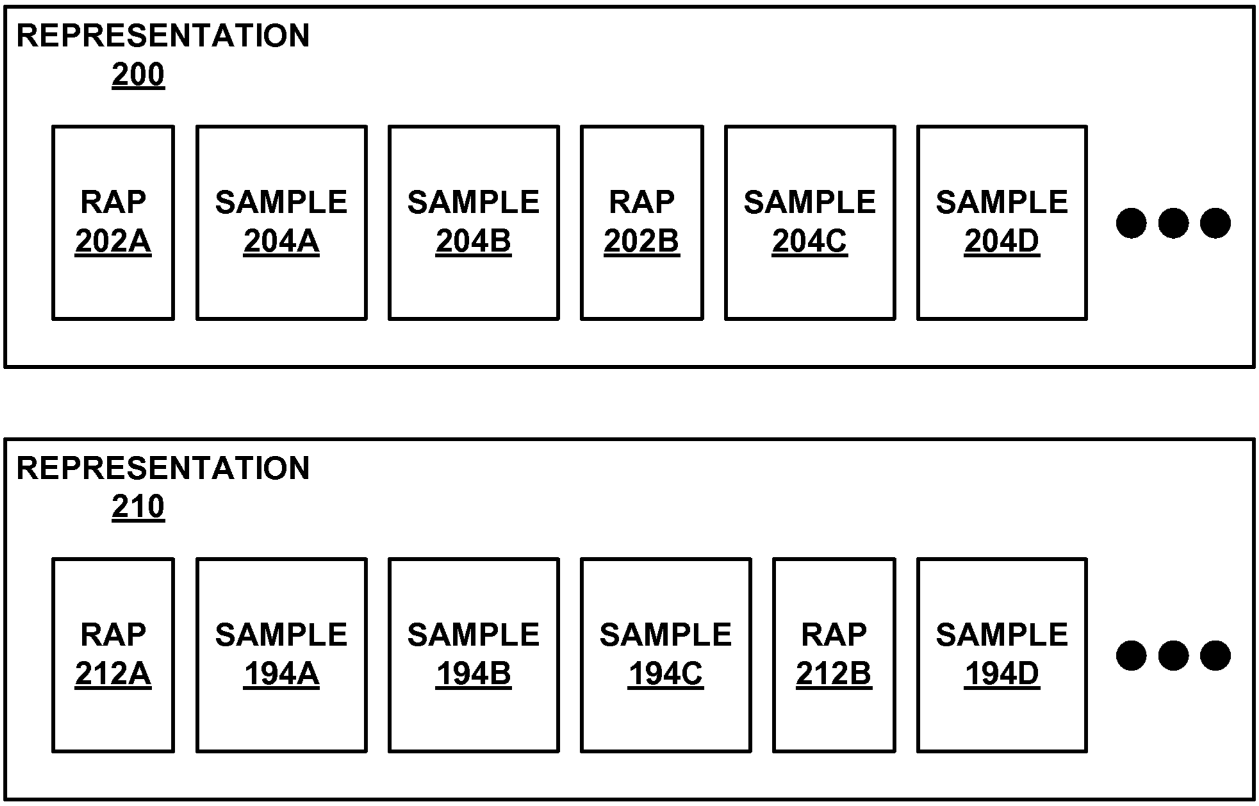
REPRESENTATION 200
RAP 202A
SAMPLE 204A
SAMPLE 204B
RAP 202B
SAMPLE 204C
SAMPLE 204D
REPRESENTATION 210
RAP 212A
SAMPLE 194A
SAMPLE 194B
SAMPLE 194C
RAP 212B
SAMPLE 194D

FIG. 5B

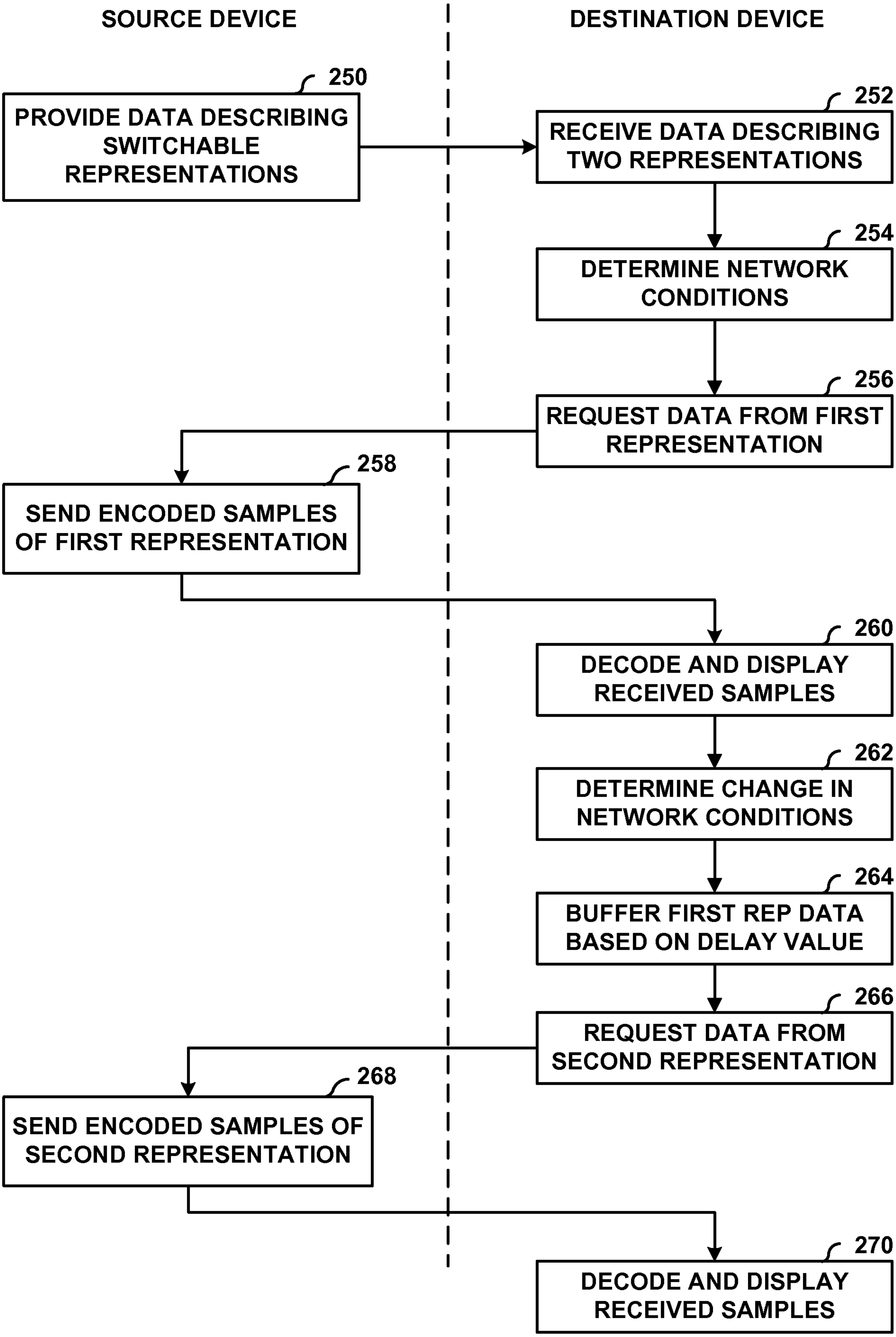
SOURCE DEVICE
DESTINATION DEVICE
250
PROVIDE DATA DESCRIBING SWITCHABLE REPRESENTATIONS
252
RECEIVE DATA DESCRIBING TWO REPRESENTATIONS
254
DETERMINE NETWORK CONDITIONS
256
REQUEST DATA FROM FIRST REPRESENTATION
258
SEND ENCODED SAMPLES OF FIRST REPRESENTATION
260
DECODE AND DISPLAY RECEIVED SAMPLES
262
DETERMINE CHANGE IN NETWORK CONDITIONS
264
BUFFER FIRST REP DATA BASED ON DELAY VALUE
266
REQUEST DATA FROM SECOND REPRESENTATION
268
SEND ENCODED SAMPLES OF SECOND REPRESENTATION
270
DECODE AND DISPLAY RECEIVED SAMPLES

FIG. 6

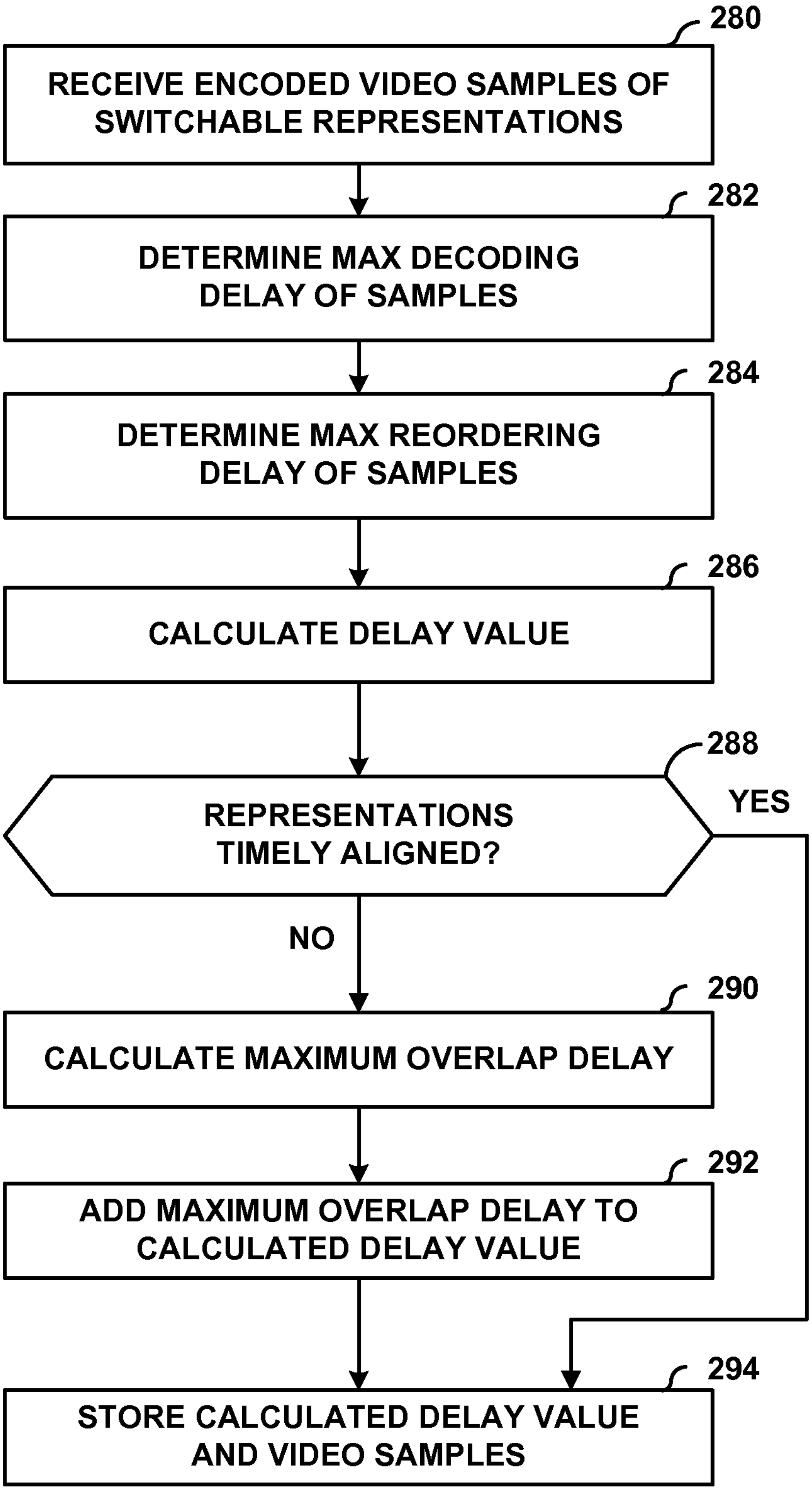
280
RECEIVE ENCODED VIDEO SAMPLES OF SWITCHABLE REPRESENTATIONS
282
DETERMINE MAX DECODING DELAY OF SAMPLES
284
DETERMINE MAX REORDERING DELAY OF SAMPLES
286
CALCULATE DELAY VALUE
288
REPRESENTATIONS TIMELY ALIGNED?
YES
NO
290
CALCULATE MAXIMUM OVERLAP DELAY
292
ADD MAXIMUM OVERLAP DELAY TO CALCULATED DELAY VALUE
294
STORE CALCULATED DELAY VALUE AND VIDEO SAMPLES

FIG. 7

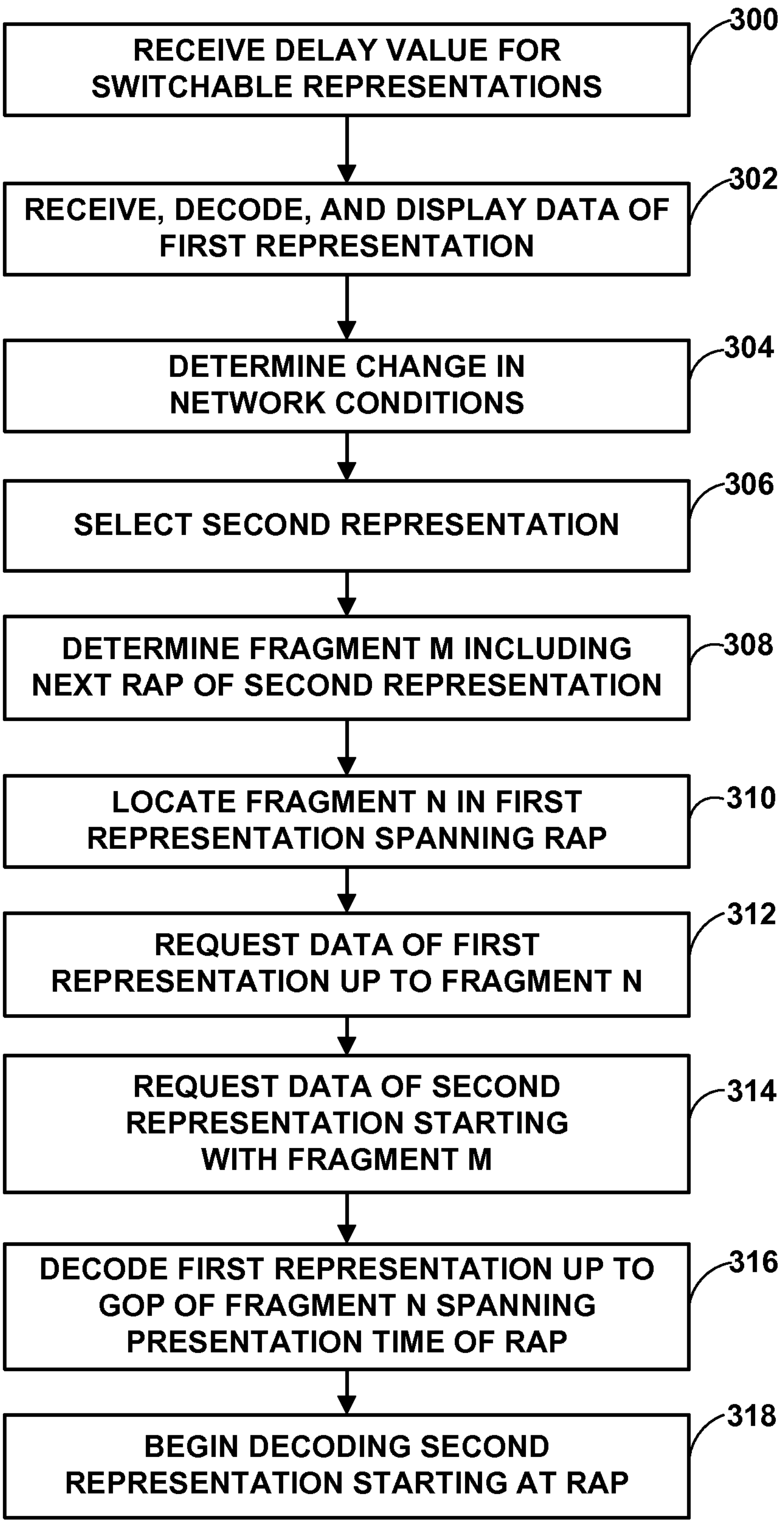
RECEIVE DELAY VALUE FOR SWITCHABLE REPRESENTATIONS
300
RECEIVE, DECODE, AND DISPLAY DATA OF FIRST REPRESENTATION
302
DETERMINE CHANGE IN NETWORK CONDITIONS
304
SELECT SECOND REPRESENTATION
306
DETERMINE FRAGMENT M INCLUDING NEXT RAP OF SECOND REPRESENTATION
308
LOCATE FRAGMENT N IN FIRST REPRESENTATION SPANNING RAP
310
REQUEST DATA OF FIRST REPRESENTATION UP TO FRAGMENT N
312
REQUEST DATA OF SECOND REPRESENTATION STARTING WITH FRAGMENT M
314
DECODE FIRST REPRESENTATION UP TO GOP OF FRAGMENT N SPANNING PRESENTATION TIME OF RAP
316
BEGIN DECODING SECOND REPRESENTATION STARTING AT RAP
318

FIG. 8

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/043885

A. CLASSIFICATION OF SUBJECT MATTER
INV. H04N7/26 H04N7/46 H04N7/50 H04N13/00 H04N21/2343
H04N21/44
ADD.

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)
H04N H04H

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

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

EPO-Internal

C. DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	US 2009/150557 A1 (WORMLEY NICHOLAS [US] ET AL) 11 June 2009 (2009-06-11)	1,6,11, 12,17, 22,24, 25,29, 31,32, 36,37, 39,40, 44,46,47
Y	abstract paragraphs [0005] - [0007], [0010], [0016] - [0026], [0029] - [0031], [0036], [0038] - [0041], [0043], [0046], [0048] - [0057] paragraphs [0065] - [0070], [0072] - [0081] claims 1-3, 5-25 figures 1-4 ----- -/--	23, 26-28, 30, 33-35, 38, 41-43, 45,48-50

☒ 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 document 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 1 December 2011	Date of mailing of the international search report 09/12/2011
Name and mailing address of the ISA/ European Patent Office, P.B. 5818 Patentlaan 2 NL - 2280 HV Rijswijk Tel. (+31-70) 340-2040, Fax: (+31-70) 340-3016	Authorized officer Beaudet, J

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
X	WO 2006/084503 A1 (ERICSSON TELEFON AB L M [SE]; HORN UWE [DE]; LOHMAR THORSTEN [DE]) 17 August 2006 (2006-08-17) the whole document -----	1,6,11, 12,17, 22,29, 36,37,44
X	US 6 415 326 B1 (GUPTA ANOOP [US] ET AL) 2 July 2002 (2002-07-02)	1,6,11, 12,17
Y	abstract figures 3-5 column 1, line 65 - column 2, line 52 column 4, lines 11-27 column 6, line 30 - column 7, line 5 column 7, line 61 - column 10, line 39 claims 1,2,5,6,9,10,13-25 -----	26-28, 33-35, 41-43, 48-50
X	"Universal Mobile Telecommunications System (UMTS); LTE; Transparent end-to-end Packet-switched Streaming Service (PSS); Protocols and codecs (3GPP TS 26.234 version 9.3.0 Release 9)", TECHNICAL SPECIFICATION, EUROPEAN TELECOMMUNICATIONS STANDARDS INSTITUTE (ETSI), 650, ROUTE DES LUCIOLES ; F-06921 SOPHIA-ANTIPOLIS ; FRANCE, vol. 3GPP SA, no. V9.3.0, 1 June 2010 (2010-06-01), XP014047290,	1,5,6, 10-12, 16,17,21
Y	paragraphs [5.5.4.2], [5.5.4.3], [5.5.4.4], [5.4.5], [5.5.4.6] paragraphs [10.2.3], [11.2.7], [12.2.3], [12.4.2], [12.6.2] paragraphs [12.6.3], [12.6.3.1], [12.6.4], [12.6.6] -----	23,30, 38,45
X	EP 1 298 931 A2 (OPLAYO OY [FI]) 2 April 2003 (2003-04-02) abstract paragraphs [0003], [0004], [0006] - [0017], [0019] - [0028], [0031], [0033] claims 1-19 -----	1,6,11, 12,17
X	US 2009/003439 A1 (WANG YE-KUI [FI] ET AL) 1 January 2009 (2009-01-01) abstract figures 7-9 paragraphs [0002], [0011], [0023] - [0034], [0044], [0046], [0049], [0059], [0069], [0071], [0078] claims 1,2,4,5,9,10,12,16,18,19,21,26,27,29,33-36,38,43 claims 44,46,50-54 -----	1,6,11, 12,17

-/--

INTERNATIONAL SEARCH REPORT

International application No
PCT/US2011/043885

C(Continuation). DOCUMENTS CONSIDERED TO BE RELEVANT

Category*	Citation of document, with indication, where appropriate, of the relevant passages	Relevant to claim No.
A	US 2002/133247 A1 (SMITH ROBERT D [US] ET AL) 19 September 2002 (2002-09-19) abstract figures 4,5 paragraphs [0009] - [0013], [0025], [0030] - [0032], [0034], [0037] - [0050], [0053] - [0062] claims 1,3-12,14-37 -----	1,6,11, 12,17, 22,29, 36,37,44

INTERNATIONAL SEARCH REPORT

Information on patent family members

International application No

PCT/US2011/043885

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 2009150557 A1	11-06-2009	US 2009150557 A1	11-06-2009
		WO 2009075766 A2	18-06-2009
WO 2006084503 A1	17-08-2006	CN 101116306 A	30-01-2008
		EP 1847087 A1	24-10-2007
		JP 2008530835 A	07-08-2008
		US 2008151885 A1	26-06-2008
		WO 2006084503 A1	17-08-2006
US 6415326 B1	02-07-2002	NONE	
EP 1298931 A2	02-04-2003	EP 1298931 A2	02-04-2003
		FI 20011856 A	21-03-2003
		US 2003055995 A1	20-03-2003
US 2009003439 A1	01-01-2009	CN 101690229 A	31-03-2010
		EP 2174502 A2	14-04-2010
		KR 20100030648 A	18-03-2010
		RU 2010102823 A	10-08-2011
		US 2009003439 A1	01-01-2009
		WO 2009001313 A2	31-12-2008
US 2002133247 A1	19-09-2002	NONE	