United States Patent Application Publication	20250260784
Kind Code	A1
Publication Date	August 14, 2025
Inventor(s)	Wang; Ye-Kui

Enhanced Signalling Of Preselection In A Media File

Abstract

A mechanism for processing video data is disclosed. The mechanism determines information from a Preselection Group Box. A segment_order field is omitted from the Preselection Group Box and sub-boxes thereof. A conversion is performed between a visual media data and the media data file based on the Preselection Group Box.

Inventors: **Wang; Ye-Kui (San Diego, CA)**

Applicant: **Bytedance Inc.** (Los Angeles, CA)

Family ID: **1000008574194**

Appl. No.: **19/098595**

Filed: **April 02, 2025**

Related U.S. Application Data

parent US continuation PCT/US2023/034015 20230928 PENDING child US 19098595
us-provisional-application US 63412632 20221003

Publication Classification

Int. Cl.: **H04N7/01** (20060101); **H04N19/70** (20140101)

U.S. Cl.:

CPC **H04N7/01** (20130101); **H04N19/70** (20141101);

Background/Summary

CROSS-REFERENCE TO RELATED PATENT APPLICATIONS [0001] This application is a continuation of International Patent Application No. PCT/US2023/034015, filed on Sep. 28, 2023, which claims the benefit of U.S. Provisional Patent Application No. 63/412,632, filed Oct. 3, 2022, the teachings and disclosure of which are hereby incorporated in their entireties by reference thereto.

TECHNICAL FIELD
[0002] This patent document relates to generation, storage, and consumption of digital audio video media information in a file format.
BACKGROUND
[0003] Digital video accounts for the largest bandwidth used on the Internet and other digital communication networks. As the number of connected user devices capable of receiving and displaying video increases, the bandwidth demand for digital video usage is likely to continue to grow.
SUMMARY
[0004] A first aspect relates to a method for processing video data comprising: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof; and performing a conversion between a visual media data and the media data file based on the PreselectionGroupBox.
[0005] A second aspect relates to an apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform any of the preceding aspects.
[0006] A third aspect relates to non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of the preceding aspects.
[0007] A fourth aspect relates to a non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof; and generating a bitstream based on the determining.
[0008] A fifth aspect relates to a method for storing bitstream of a video comprising: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof; generating a bitstream based on the determining; and storing the bitstream in a non-transitory computer-readable recording medium.
[0009] For the purpose of clarity, any one of the foregoing embodiments may be combined with any one or more of the other foregoing embodiments to create a new embodiment within the scope of the present disclosure.
[0010] These and other features will be more clearly understood from the following detailed description taken in conjunction with the accompanying drawings and claims.

Description

BRIEF DESCRIPTION OF THE DRAWINGS
[0011] For a more complete understanding of this disclosure, reference is now made to the following brief description, taken in connection with the accompanying drawings and detailed description, wherein like reference numerals represent like parts.
[0012] FIG. **1** is a block diagram showing an example video processing system.

[0013] FIG. **2** is a block diagram of an example video processing apparatus.
[0014] FIG. **3** is a flowchart for an example method of video processing.
[0015] FIG. **4** is a block diagram that illustrates an example video coding system.
[0016] FIG. **5** is a block diagram that illustrates an example encoder.
[0017] FIG. **6** is a block diagram that illustrates an example decoder.
[0018] FIG. **7** is a schematic diagram of an example encoder.
[0019] FIG. **8** is a flowchart for an example method of video processing.

DETAILED DESCRIPTION

[0020] It should be understood at the outset that although an illustrative implementation of one or more embodiments are provided below, the disclosed systems and/or methods may be implemented using any number of techniques, whether currently known or yet to be developed. The disclosure should in no way be limited to the illustrative implementations, drawings, and techniques illustrated below, including the exemplary designs and implementations illustrated and described herein, but may be modified within the scope of the appended claims along with their full scope of equivalents.

[0021] Section headings are used in the present document for ease of understanding and do not limit the applicability of techniques and embodiments disclosed in each section only to that section. Furthermore, H.266 terminology is used in some description only for ease of understanding and not for limiting scope of the disclosed techniques. As such, the techniques described herein are applicable to other video codec protocols and designs also. In the present document, editing changes are shown to text by bold italics indicating cancelled text and bold underline indicating added text, with respect to a draft of the Versatile Video Coding (VVC) specification or ISO base media file format (ISOBMFF) file format specification.

1. INITIAL DISCUSSION

[0022] This document is related to media file formats. Specifically, it is related to signaling of preselection in a media file, wherein a preselection is a set of one or more tracks representing one version of the media content for simultaneous decoding or presentation. The ideas may be applied individually or in various combination, to media files according to any media file formats, e.g., the ISOBMFF and file format derived from the ISOBMFF.

2. VIDEO CODING INTRODUCTION

2.1 Video Coding Standards

[0023] Video coding standards have evolved primarily through the development of the International Telecommunication Union (ITU) telecommunication standardization sector (ITU-T) and International Organization for Standardization (ISO)/International Electrotechnical Commission (IEC) standards. The ITU-T produced H.261 and H.263, ISO/IEC produced motion picture experts group (MPEG)-1 and MPEG-4 Visual, and the two organizations jointly produced the H.262/MPEG-2 Video and H.264/MPEG-4 Advanced Video Coding (AVC) and H.265/high efficiency video coding (HEVC) [1] standards. Since H.262, the video coding standards are based on the hybrid video coding structure wherein temporal prediction plus transform coding are utilized. The Versatile Video Coding (VVC) standard (ITU-T H.266 | ISO/IEC 23090-3) [2] and the associated Versatile Supplemental Enhancement Information (VSEI) standard (ITU-T H.274 | ISO/IEC 23002-7) [3] are designed for use in a maximally broad range of applications, including both the traditional uses such as television broadcast, video conferencing, or playback from storage media, and also newer and more advanced use cases such as adaptive bit rate streaming, video region extraction, composition and merging of content from multiple coded video bitstreams, multiview video, scalable layered coding, and viewport-adaptive 360° immersive media. The Essential Video Coding (EVC) standard (ISO/IEC 23094-1) is another video coding standard that is developed by MPEG.

2.2 File Format Standards

[0024] Media streaming applications are based on the internet protocol (IP), Transmission Control

Protocol (TCP), and HyperText Transfer Protocol (HTTP) transport methods, and rely on a file format such as the ISO base media file format (ISOBMFF) [4]. One such streaming system is dynamic adaptive streaming over HTTP (DASH) [5]. For using a video format with ISOBMFF and DASH, a file format specification specific to the video format, also referred to as network abstraction layer file format (NALFF) [6], which includes the file format specifications for all NAL units based video codecs such as AVC, HEVC, VVC, and their extensions, would be needed for encapsulation of the video content in ISOBMFF tracks and in DASH representations and segments. Important information about the video bitstreams, e.g., the profile, tier, and level, and many others, would need to be exposed as file format level metadata and/or DASH media presentation description (MPD) for content selection purposes, e.g., for selection of appropriate media segments both for initialization at the beginning of a streaming session and for stream adaptation during the streaming session. Similarly, for using an image format with ISOBMFF, a file format specification specific to the image format, such as the AVC image file format and the HEVC image file format in [7], would be needed.

2.3 Preselection Signaling in the ISOBMFF

[0025] An edition of the ISOBMFF standard is being developed. A draft specification of this edition is included in [8], which includes the specification of some features for preselection signalling, as follows.

. . .

3.1 Defections and Terms

. . .

Preselection

[0026] Set of one or more tracks representing one version of the media presentation for simultaneous decoding or presentation.

. . .

8.3.4.4.3 Preselection Group Box

8.3.4.4.3.1 Definition

[0027] The presence of a TrackGroupTypeBox with track_group_type equal to ‘pres’, which is also referred to as a PreselectionGroupBox, in a track indicates that this track contributes to a preselection.

[0028] All the tracks that have a track group with track_group_type equal to ‘pres’ and a particular value of track_group_id are part of the same preselection. The particular value of track_group_id is also referred to as the identifier (ID) of the preselection.

[0029] NOTE 1. This means that a preselection is uniquely identified by the track_group_id of the track group.

[0030] When multiple tracks contribute to a preselection, the optionally present PreselectionProcessingBox provides information on how to process the track containing this box in the context of the preselection and relative to other tracks. Consequently, the content of the PreselectionProcessingBox may differ for each track within a preselection.

[0031] NOTE 2. Preselections made from only one track do not require any track-related processing. In this case, the PreselectionProcessingBox is typically not present in the PreselectionGroupBox.

8.3.4.4.3.2 Syntax

TABLE-US-00001 aligned(8) class PreselectionGroupBox extends TrackGroupTypeBox(‘pres’) {
PreselectionProcessingBox preselection_processing; // optional }

8.3.4.4.3.3 Semantics

[0032] preselection_processing is an instance of the PreselectionProcessingBox, providing information needed for processing the containing track in the context of the preselection.

8.3.4.4.3.4 Preselection Processing Box

8.3.4.4.3.4.1 Definition

[0033] Box Type: ‘prsp’ [0034] Container: PreselectionGroupBox [0035] Mandatory: No [0036] Quantity: Zero or one
[0037] This box contains information about how the tracks contributing to the preselection can be processed. Media type specific boxes may be used to describe further processing.
8.3.4.4.3.4.2 Syntax
TABLE-US-00002 aligned(8) class PreselectionProcessingBox extends FullBox(‘prsp’, version=0, flags){ unsigned int(8) track_order; unsigned int(1) sample_merge_flag; unsigned int(7) reserved; // further attributes and Boxes defining additional processing of // the track contributing to the preselection }
8.3.4.4.3.4.3 Semantics
[0038] track_order defines the order of this track relative to other tracks in the preselection as described below.
[0039] sample_merge_flag equal to 1 indicates that this track is enabled to be merged with another track as described below.
[0040] Sample entry specific specifications might require the tracks for a preselection to be provided to the respective decoder instances in a specific order. Since other means, such as the track_id, are not reliable for this purpose, the track_order is used to order tracks in a preselection relative to each other. A lower number indicates that at a given time the sample of the containing track is provided to the decoder before the sample with the same given of other tracks with higher number. If two tracks in a preselection have their track_order set to the same value or if the preselection processing box is absent for at least one of the tracks, the order of these tracks is not relevant for the preselection, and samples can be provided to the decoder in any order.
[0041] A merge group is defined as a group of tracks, sorted according to track_order, where one track with the sample_merge_flag set to 0 is followed by a group of consecutive tracks with the sample_merge_flag set to 1. All tracks of a merge group shall be of the same media type and shall have all samples time-aligned.
[0042] If the sample entry type is associated with a codec-specific process to merge samples of a preselection, this process shall be used. [0043] NOTE 1. If the tracks in the merge group are all of sample entry type of “mhm2” (MPEG-H 3D Audio), the merging process is defined in ISO/IEC 23008-3:2019, subclause 14.6. [0044] NOTE 2. Tracks in a merge group may have different sample entry types.
[0045] If the sample entry type is not associated with a codec-specific process to merge samples of a preselection, the following process shall be used:
[0046] Merging within the merge group shall proceed by forming tuples of track samples with the same time stamp across contributing tracks. The ordering of samples within the tuple shall be determined by track_order. These tuples shall be formed by byte-wise concatenation of the samples resulting in a single sample with having the respective time stamp assigned. If generation of new tracks is targeted, each merge group shall result in a separate output track conformant to a media type derived from the media types of the merged tracks.
[0047] For tracks not part of a merge group, a merging process is not specified by the present document.
8.3.5 Track Group Description Box
8.3.5.1 Definitions
[0048] Box Type: ‘tkgd’ [0049] Container: MovieBox [0050] Mandatory: No [0051] Quantity: Zero or One
[0052] The TrackGroupDescriptionBox provides an array of TrackGroupEntryBoxes, where each TrackGroupEntryBox provides detailed characteristics of a particular track group. The syntax of the TrackGroupEntryBox is determined by track_group_entry_type. TrackGroupEntryBox is mapped to the track group by a unique track_group_entry_type that is associated with a track_group_type defined in subclause 8.3.4.3. More than one TrackGroupEntryBox with the same

track_group_entry_type may be present in TrackGroupDescriptionBox, in that case TrackGroupEntryBoxes shall have different track_group_id.
8.3.5.2 Syntax
TABLE-US-00003 aligned(8) class TrackGroupDescriptionBox extends Box('tkgd') { Box boxes[ ]; } aligned(8) abstract class TrackGroupEntryBox (unsigned int(32) track_group_entry_type, unsigned int(8) version, unsigned int(24) flags) extends FullBox(track_group_entry_type, version, flags) { unsigned int(32) track_group_id; // the remaining data may be specified // for a particular track_group_entry_type }
8.5.5.5 Semantics
[0053] The TrackGroupDescriptionBox contains TrackGroupEntryBoxes.
[0054] track_group_entry_type indicates a four character code (4CC) that is associated with a track_group_type. The pair of track_group_id and track_group_entry_type identifies the track group that the TrackGroupEntryBox describes.
8.3.5.4 Track Group Entry Definitions
8.3.5.4.1 Preselection Track Group Entry Box
8.3.5.4.1.1 Definition
[0055] Box Type: 'prse' [0056] Container: TrackGroupDescriptionBox [0057] Mandatory: No [0058] Quantity: Zero or More
[0059] Preselections can be qualified, for example, by language, kind or media specific attributes like audio rendering indications, audio interactivity or channel layouts. Attributes signalled in a PreselectionTrackGroupEntryBox take precedence over attributes signalled in contributing tracks.
[0060] Preselection TrackGroupEntryBox shall describe only track groups identified by track_group_type equal to 'prse'.
[0061] All preselections with at least one contributing track having the track_in_movie flag set to 1 shall be qualified by PreselectionTrackGroupEntryBoxes. Otherwise, the presence of the Preselection TrackGroupEntryBoxes is optional.
[0062] All attributes uniquely qualifying a preselection shall be present in PreselectionTrackGroupEntryBox of the preselection.
8.3.5.4.1.2 Syntax
TABLE-US-00004 aligned(8) class PreselectionTrackGroupEntryBox extends TrackGroupEntryBox('prse', version=0, flags) { unsigned int(8) num_tracks; utf8string preselection_tag; if (flags & 1) { unsigned int(8) selection_priority; } if (flags & 2) { unsigned int(8) segment_order; } // Boxes describing the preselection }
8.3.5.4.1.3 Semantics
[0063] This box contains information on what experience is available when this preselection is selected.
[0064] Boxes suitable to describe a preselection include but are not limited to the following list of boxes defined in this document: [0065] AudioElementBox (subclause 12.2.9) [0066] AudioElementSelectionBox (subclause 12.2.13) [0067] ExtendedLanguageBox (subclause 8.4.6) [0068] UserDataBox (subclause 8.10.1) [0069] KindBox (subclause 8.10.4) [0070] LabelBox (subclause 8.10.5) [0071] AudioRenderingIndicationBox (subclause 12.2.8) [0072] ChannelLayout (subclause 12.2.4)
[0073] If a UserDataBox is contained in a PreselectionTrackGroupEntryBox, then it shall not carry any of the above boxes.
[0074] num_tracks specifies the number of non-alternative tracks grouped by this preselection track group.
[0075] A track grouped by this preselection track group is a track that has the 'pres' track group with track_group_id equal to the ID of this preselection.
[0076] The number of non-alternative tracks grouped by this preselection track group is the sum of the following: [0077] the number of tracks that have alternate_group equal to 0 and are grouped by

this preselection track group, [0078] the number of unique non-zero alternate_group values in all tracks that are grouped by this preselection track group.
[0079] The value of num_tracks shall be greater than or equal to the number of non-alternative tracks grouped by this preselection track group in this file. A value equal to 0 indicates that the number of tracks grouped by this track group is unknown or not essential for processing the track group. [0080] NOTE 1. The value of num_tracks can be greater than the number of non-alternative tracks containing a PreselectionGroupBox with the same track_group_id in this file when the preselection is split into multiple files. [0081] NOTE 2. When a player has access to fewer non-alternative tracks grouped by this preselection track group than indicated by num_tracks, the player might need to omit the tracks grouped by this preselection track group.
[0082] preselection_tag is a codec specific value that a playback system can provide to a decoder to uniquely identify one out of several preselections in the media.
[0083] selection_priority is an integer that declares the priority of the preselection in cases where no other differentiation such as through the media language is possible. A lower number indicates a higher priority.
[0084] segment_order specifies, if present, an order rule of segments that is suggested to be followed for ordering received segments of the Preselection. The following values are specified with semantics according to ISO/IEC 23009-1:2022, subclause 5.3.11.5: [0085] 0: undefined
[0086] 1: time-ordered [0087] 2: fully-ordered
[0088] Other values are reserved. If segment_order is not present, its value shall be inferred to be equal to 0. [0089] NOTE 3. Not all tracks contributing to the playout of a preselection may be delivered in the same file. [0090] NOTE 4. The kind box might utilize the Role scheme defined in ISO/IEC 23009-1:2022, subclause 5.8.5.5 as it provides a commonly used scheme to describe characteristics of preselections. [0091] NOTE 5. This box carries information about the initial experience of the preselection in the referenced tracks. The preselection experience can change during the playback of these tracks, e.g., audio language can change during playback. These changes are not subject to the information presented in this box.
[0092] Further media type specific boxes may be used to describe properties of the preselection. Readers may ignore and skip boxes that are not recognized.
8.3.4.4.4 Preselection Track Group Entry Box
8.3.4.4.4.1 Definition
[0093] Box Type: ‘prse’ [0094] Container: TrackGroupDescriptionBox [0095] Mandatory: No
[0096] Quantity: Zero or More
[0097] Preselections can be qualified, for example, by language, kind or media specific attributes like audio rendering indications, object interactivity or channel layouts. Attributes signalled in a PreselectionTrackGroupEntryBox shall take precedence over attributes signalled in contributing tracks. Preselection TrackGroupEntryBox shall described only track groups identified by track_group_type equal to ‘pres’. All attributes uniquely qualifying a preselection shall be present in PreselectionTrackGroupEntryBox of the preselection.
8.3.4.4.4.2 Syntax
TABLE-US-00005 aligned(8) class PreselectionTrackGroupEntryBox (track_group_entry_type = ‘prse’, version = 0, flags) extends TrackGroupEntryBox(track_group_entry_type, version, flags) { if (flags & 1) { unsigned int(8) selection_priority=1; }
PreselectionInformationBox presel_info; PreselectionProcessingBox presel_processing; }
8.3.4.4.4.3 Semantics
[0098] selection_priority is an integer that declares the priority of the preselection in cases where no other differentiation such as through the media language is possible. A lower number indicates a higher priority. presel_info is an instance of the PreselectionInformationBox, providing information that describes the preselection. presel_processing is an instance of the PreselectionProcessingBox, providing information needed for processing the preselection.

8.3.4.4.5 Preselection Information Box
8.3.4.4.5.1 Definition
[0099] Box Type: ‘prsi’ [0100] Container: PreselectionGroupBox [0101] Mandatory: No [0102] Quantity: Zero or One
[0103] This Box aggregates all semantic information about the preselection.
8.3.4.4.5.2 Syntax
TABLE-US-00006 aligned(8) class PreselectionInformationBox extends FullBox(‘prsi’, version=0, 0){ unsigned int(8) numTracks; // Boxes describing the preselection }
8.3.4.4.5.3 Semantics
[0104] This box contains information on what experience is available when this preselection is selected. Boxes suitable to describe a preselection include but are not limited to the following list of boxes defined in this specification: [0105] The audio object box (clause 8.3.4.4.5) [0106] The audio object selection box (clause 8.3.4.4.7) [0107] The extended language tag (clause 8.4.6) [0108] The user data box (clause 8.10.1) [0109] The track kind (clause 8.10.4) [0110] The label box (clause 8.18.4) [0111] The audio rendering indication (clause 8.18.5) [0112] The channel layout (clause 12.2.4.1)
[0113] If the user data box is contained in a PreselectionBox, then it shall not carry any of the above boxes. numTracks declares how many tracks are contributing to the playout of the preselection. This value shall match the number of tracks containing a PreselectionGroupBox with the same track_group_id. Note 1: not all tracks contributing to the playout of a preselection may be delivered in the same file. Note 2: The kind box might utilize the Role scheme defined in ISO/IEC 23009-1, clause 5.8.5.5 as it provides a commonly used scheme to describe characteristics of preselections. Further media type specific boxes may be used to describe properties of the preselection.
. . .
8.3.4.4.10 Preselection Processing Box
8.3.4.4.10.1 Definition
[0114] Box Type: ‘prsp’ [0115] Container: PreselectionGroupBox [0116] Mandatory: Yes [0117] Quantity: Exactly One
[0118] This box contains information how the tracks contributing to the preselection shall be processed. Media type specific boxes may be used to describe further processing.
8.3.4.4.10.2 Syntax
TABLE-US-00007 aligned(8) class PreselectionProcessingBox extends FullBox(‘prsp’, version=0, flags){ utf8string preselection_tag; if (flags & 1) { unsigned int(8) order=0; } unsigned int(8) track_order; unsigned int(1) sample_merge_flag; unsigned int(7) reserved; // further attributes and Boxes defining additional processing of // the track contributing to the preselection }
8.3.4.4.10.3 Semantics
[0119] preselection_tag is a codec specific value that a playback system can provide to a decoder to uniquely identify one out of several preselections in the media. order specifies the conformance rules for Representations in Adaptation Sets within the Preselection according to ISO/IEC 23009-1, from the following enumerated set: [0120] 0: undefined [0121] 1: time-ordered [0122] 2: fully-ordered
[0123] track_order defines the order of this track for the merging process described below. sample_merge_flag equal to 1 indicates that this track is enabled to be merged with another track as described below. Tracks contributing to a selected preselection and having the sample_merge_flag equal to 1 shall merge their samples according to the respective media type. If the media type does not specify such a process, contributing samples may be appended to the samples of the track with the next lower track_order. If the generated output samples from this merging process shall be embedded into a new track, this track shall be conformant to a media type derived from the base

media type. Note: for MPEG high efficiency (MPEG-H) three dimensional (3D) Audio, this process is defined in ISO/IEC 23008-3, clause 14.6.

. . .

3. TECHNICAL PROBLEMS SOLVED BY DISCLOSED TECHNICAL SOLUTIONS

[0124] Example designs for preselections signalling in the ISOBMFF have the following problems:

[0125] First, a DASH Adaptation Set in the ISOBMFF maps to an alternate group. A DASH Preselection comprises Adaptation Sets. Although signalled through track grouping, an ISOBMFF preselection should comprise alternate groups. In other words, an ISOBMFF preselection track group should not group more than one track of any particular alternate group. However, currently, this is allowed.

[0126] Second, a DASH Preselection comprises one or more Adaptation Sets, and one of them is indicated as the Main Adaptation Set. However, there is no corresponding indication for an ISOBMFF preselection.

[0127] Third, the following is specified:

[0128] All preselections with at least one contributing track having the track_in_movie flag set to 1 shall be qualified by PreselectionTrackGroupEntryBoxes. Otherwise, the presence of the Preselection TrackGroupEntryBoxes is optional.

[0129] However, tracks contributing to a preselection may be stored in separate ISOBMFF files, therefore it is possible that a preselection in a current file has one or more tracks having the track_in_movie flag equal to 1, but those tracks are all in other files, while in the current file the tracks contributing to the preselection all have the track_in_movie flag equal to 0. In this case, the presence of the Preselection TrackGroupEntryBox in the current file for the preselection should not be required.

[0130] Fourth, the preselection_tag field in the PreselectionTrackGroupEntryBox only provides redundant information and thus should be removed. In DASH, the preselection tag is only specified for the descriptor-based preselection approach, for uniquely identifying a preselection, and it is not specified for the element-based approach, for which there is already the preselection ID as well as the characterizing parameters. In the ISOBMFF, since there is already the preselection ID, i.e., the track group ID, and the characterizing parameters, this preselection_tag field is not needed, just like in DASH for the element-based preselection approach.

[0131] Fifth, the segment_order field in the Preselection TrackGroupEntryBox is unclearly specified and it only provides redundant information, and thus should be removed. The way the specification is written in ISO/IEC 23009-1:2022, subclause 5.3.11.5 is heavily dependent on some general DASH contexts as well as on some specific details of the DASH preselection signalling, both of which are not available in the specific ISOBMFF context herein. That made the current specification of the semantics of segment_order basically impossible for ISOBMFF implementers to be able to follow. For example, merging at DASH level is segments/subsegments based and the output is fed to the underlying ISOBMFF parser, while at ISOBMFF level merging is sample based, and the output is fed to the decoder. Furthermore, segment_order field is redundant with what the track_order field in the PreselectionProcessingBox is providing. In the file format level the data provided by the track_order field in the PreselectionProcessingBox is sufficient, in terms of needed processing order in DASH level. The corresponding processing order in file format level can already be provided by the track_order fields in the PreselectionProcessingBoxes of the tracks contributing to the preselection. This is because the decoding time based order already satisfies the so-called time-ordered requirement in DASH. Further, the track_order fields in the PreselectionProcessingBoxes of the tracks contributing to the preselection can provide the so-called fully-ordered requirement in DASH.

4. A LISTING OF SOLUTIONS AND EMBODIMENTS

[0132] To solve the above-described problem, methods as summarized below are disclosed. The inventions should be considered as examples to explain the general concepts and should not be

interpreted in a narrow way. Furthermore, these inventions can be applied individually or combined in any manner.

Example 1

[0133] To solve the first problem, one or more of the following are specified:

[0134] When any two tracks belong to the same alternate group, they shall not both have a PreselectionGroupBox with the same value of track_group_id.

[0135] When processing a preselection to which a track contributes, that track can be replaced by any other track in the same alternate group that track belongs to.

Example 2

[0136] To solve the second problem, an indication of a main track for a preselection is signalled.

[0137] In one example, the indication is signalled by a field in a track contributing to the preselection.

[0138] In one example, the field is a one-bit flag.

[0139] In one example, the flag equal to 1 indicates that the track is the main track in the preselection, and the flag equal to 0 indicates that the track is not the main track in the preselection.

[0140] In one example, it is required that one and only one track among all tracks contributing to the preselection shall have the flag equal to 1.

[0141] In one example, the indication is signalled through the track_order field.

[0142] In one example, it is specified that the main track is the one with the lowest value of track_order.

[0143] In one example, it is specified that, in case there are multiple tracks contributing to the presentation having

[0144] the smallest value of track_order, any of them can be used as the main track.

[0145] In one example, it is specified that, when the track_order field is not present (e.g., when the PreselectionProcessingBox is not present) for a track contributing to the preselection, the value of track_order is inferred to be equal to 0.

[0146] In one example, it is specified that, when the media content is encapsulated as a DASH Media Presentation, the Adaptation Set containing the Representation for the main track in a preselection would be the Main Adaptation Set for the DASH Preselection.

Example 3

[0147] To solve third problem, the following is specified:

[0148] For any preselection, if at least one contributing track in this file has the track_in_movie flag equal to 1, the PreselectionTrackGroupEntryBox shall be present in this file for the preselection; otherwise, the presence of the Preselection TrackGroupEntryBox for the preselection in this file is optional.

Example 4

[0149] To solve fourth problem, the preselection_tag field in the Preselection TrackGroupEntryBox is removed.

Example 5

[0150] To solve fifth problem, the segment_order field in the PreselectionTrackGroupEntryBox is removed.

5. EMBODIMENTS

[0151] Below are some example embodiments for all the disclosure items and most of their subitems summarized above in Section 4. Most relevant parts that have been added or modified are shown in underlined bold font, and some of the deleted parts are shown in italicized bold fonts. There may be some other changes that are editorial in nature and thus not noted. Parts that remain unchanged are not included.

5.1 First Embodiment

[0152] This embodiment is for items 1-5, except for some sub-items described in example 2.

. . .

3.1 Definitions and Terms
. . .
[0153] preselection
[0154] set of one or more tracks representing one version of the media presentation content for simultaneous decoding or presentation
. . .
8.3.4.4.3 Preselection Group Box
8.3.4.4.3.1 Definition
[0155] The presence of a TrackGroupTypeBox with track_group_type equal to ‘pres’, which is also referred to as a PreselectionGroupBox, in a track indicates that this track contributes to a preselection.
[0156] All the tracks that have a track group with track_group_type equal to ‘pres’ and a particular value of track_group_id are part of the same preselection. The particular value of track_group_id is also referred to as the ID of the preselection.
[0157] NOTE 1 This means that a preselection is uniquely identified by the track_group_id of the track group.
[0158] When any two tracks belong to the same alternate group, they shall not both have a PreselectionGroupBox with the same value of track_group_id. However, when processing a preselection to which a track contributes, that track can be replaced by any other track in the same alternate group that track belongs to.
[0159] When multiple tracks contribute to a preselection, the optionally present PreselectionProcessingBox provides information on how to process the track containing this box in the context of the preselection and relative to other tracks. Consequently, the content of the PreselectionProcessingBox may differ for each track within a preselection.
[0160] NOTE 2 Preselections consisting of only one track do not require any track-related processing. In this case, the PreselectionProcessingBox is typically not present in the PreselectionGroupBox.
8.3.4.4.3.2 Syntax
TABLE-US-00008 aligned(8) class PreselectionGroupBox extends TrackGroupTypeBox(‘pres’) {
unsigned int(1) main_track_in_preselection_flag; unsigned int(7) reserved;
PreselectionProcessingBox preselection_processing; // optional }
8.3.4.4.3.3 Semantics
[0161] main_track_in_preselection_flag equal to 1 indicates that the track is the main track in the preselection. main_track_in_preselection_flag equal to 0 indicates that the track is not the main track in the preselection. One and only one track among all tracks contributing to a preselection shall have main_track_in_preselection_flag equal to 1.
[0162] NOTE When the media content is encapsulated as a DASH Media Presentation, the Adaptation Set containing the Representation for the main track in a preselection would be the Main Adaptation Set for the DASH Preselection.
[0163] preselection_processing is an instance of the PreselectionProcessingBox, providing information needed for processing the containing track in the context of the preselection and relative to other tracks.
. . .
8.3.5.4 Track Group Entry Definitions
8.3.5.4.1 Preselection Track Group Entry Box
8.3.5.4.1.1 Definition
[0164] Box Type: ‘prse’ [0165] Container: TrackGroupDescriptionBox [0166] Mandatory: No
[0167] Quantity: Zero or More
[0168] Preselections can be characterized, for example, by language, kind or media specific attributes like audio rendering indications, audio interactivity or channel layouts. Attributes

signalled in a PreselectionTrackGroupEntryBox take precedence over attributes signalled in contributing tracks.
[0169] Preselection TrackGroupEntryBox shall describe only track groups identified by track_group_type equal to ‘prse’.
[0170] All preselections with at least one contributing track having the track_in_movie flag set to 1 shall be qualified by Preselection TrackGroupEntryBoxes. Otherwise, the presence of the Preselection TrackGroupEntryBoxes is optional.
[0171] For any preselection, if at least one contributing track in this file has the track_in_movie flag equal to 1, the PreselectionTrackGroupEntryBox shall be present in this file for the preselection; otherwise, the presence of the PreselectionTrackGroupEntryBox for the preselection in this file is optional.
[0172] All attributes uniquely characterizing a preselection shall be present in the Preselection TrackGroupEntryBox for the preselection.
8.3.5.4.1.2 Syntax
TABLE-US-00009 aligned(8) class PreselectionTrackGroupEntryBox extends TrackGroupEntryBox(‘prse’, version=0, flags) { unsigned int(8) num_tracks; utf8string preselection_tag; if (flags & 1) { unsigned int(8) selection_priority; } if (flags & 2) { unsigned int(8) segment_order; } // Boxes describing the preselection }
8.3.5.4.1.3 Semantics
[0173] This box contains information on what experience is available when this preselection is selected.
[0174] Boxes suitable to describe a preselection include but are not limited to the following list of boxes defined in this document: [0175] AudioElementBox (subclause 12.2.9) [0176] AudioElementSelectionBox (subclause 12.2.13) [0177] ExtendedLanguageBox (subclause 8.4.6) [0178] UserDataBox (subclause 8.10.1) [0179] KindBox (subclause 8.10.4) [0180] LabelBox (subclause 8.10.5) [0181] AudioRenderingIndicationBox (subclause 12.2.8) [0182] ChannelLayout (subclause 12.2.4)
[0183] If a UserDataBox is contained in a PreselectionTrackGroupEntryBox, then it shall not carry any of the above boxes.
[0184] When contained in a Preselection TrackGroupEntryBox, a UserDataBox shall not carry any of the above boxes.
[0185] num_tracks specifies the number of non-alternative tracks grouped by this preselection track group.
[0186] A track grouped by this preselection track group is a track that has the ‘pres’ track group with track_group_id equal to the ID of this preselection.
[0187] The number of non-alternative tracks grouped by this preselection track group is the sum of the following: [0188] the number of tracks that have alternate_group equal to 0 and are grouped by this preselection track group, [0189] the number of unique non-zero alternate_group values in all tracks that are grouped by this preselection track group.
[0190] The value of num_tracks shall be greater than or equal to the number of non-alternative tracks grouped by this this preselection track group in file.
[0191] A value equal to 0 indicates that the number of tracks grouped by this track group is unknown or not essential for processing the track group. [0192] NOTE 1 The value of num_tracks can be greater than the number of non-alternative tracks containing a PreselectionGroupBox with the same track_group_id in this file when the preselection is split into multiple files. [0193] NOTE 2 When a player has access to fewer non-alternative tracks grouped by this preselection track group than indicated by num_tracks, the player might need to omit the tracks grouped by this preselection track group.
[0194] preselection_tag is a codec specific value that a playback system can provide to a decoder to uniquely identify one out of several preselections in the media.

[0195] selection_priority is an integer that declares the priority of the preselection in cases where no other differentiation such as through the media language is possible. A lower value of selection_priority indicates a higher priority.
[0196] segment_order specifies, if present, an order rule of segments that is suggested to be followed for ordering received segments of the Preselection. The following values are specified with semantics according to ISO/IEC 23009-1:2022, subclause 5.3.11.5: [0197] 0: undefined
[0198] 1: time-ordered [0199] 2: fully-ordered
[0200] Other values are reserved. If segment_order is not present, its value shall be inferred to be equal to 0. [0201] NOTE 3 Not all tracks contributing to the playout of a preselection may be delivered in the same file. [0202] NOTE 4 The kind box might utilize the Role scheme defined in ISO/IEC 23009-1:2022, subclause 5.8.5.5 as it provides a commonly used scheme to describe characteristics of preselections. [0203] NOTE 5 This box carries information about the initial experience of the preselection in the referenced tracks. The preselection experience can change during the playback of these tracks, e.g., audio language can change during playback. These changes are not subject to the information presented in this box.
[0204] Further media type specific boxes may be used to describe properties of the preselection. Readers may ignore and skip boxes that are not recognized.
. . .
6. REFERENCES
[0205] [1] ITU-T and ISO/IEC, “High efficiency video coding”, Rec. ITU-T H.265 | ISO/IEC 23008-2 (in force edition). [0206] [2] Rec. ITU-T H.266 | ISO/IEC 23090-3, “Versatile Video Coding”. [0207] [3] Rec. ITU-T Rec. H.274 | ISO/IEC 23002-7, “Versatile Supplemental Enhancement Information Messages for Coded Video Bitstreams”. [0208] [4] ISO/IEC 14496-12: “Information technology—Coding of audio-visual objects—Part 12: ISO base media file format”. [0209] [5] ISO/IEC 23009-1: “Information technology—Dynamic adaptive streaming over HTTP (DASH)—Part 1: Media presentation description and segment formats”. [0210] [6] ISO/IEC 14496-15: “Information technology—Coding of audio-visual objects—Part 15: Carriage of network abstraction layer (NAL) unit structured video in the ISO base media file format”. [0211] [7] ISO/IEC 23008-12: “Information technology—High efficiency coding and media delivery in heterogeneous environments—Part 12: Image File Format”. [0212] [8] ISO/IEC JTC 1/SC 29/WG 03 output document N0651, “Text of ISO/IEC DIS 14496-12 8th edition ISO Base Media File Format”, September 2022.
[0213] FIG. **1** is a block diagram showing an example video processing system **4000** in which various techniques disclosed herein may be implemented. Various implementations may include some or all of the components of the system **4000**. The system **4000** may include input **4002** for receiving video content. The video content may be received in a raw or uncompressed format, e.g., 8 or 10 bit multi-component pixel values, or may be in a compressed or encoded format. The input **4002** may represent a network interface, a peripheral bus interface, or a storage interface. Examples of network interface include wired interfaces such as Ethernet, passive optical network (PON), etc. and wireless interfaces such as Wi-Fi or cellular interfaces.
[0214] The system **4000** may include a coding component **4004** that may implement the various coding or encoding methods described in the present document. The coding component **4004** may reduce the average bitrate of video from the input **4002** to the output of the coding component **4004** to produce a coded representation of the video. The coding techniques are therefore sometimes called video compression or video transcoding techniques. The output of the coding component **4004** may be either stored, or transmitted via a communication connected, as represented by the component **4006**. The stored or communicated bitstream (or coded) representation of the video received at the input **4002** may be used by a component **4008** for generating pixel values or displayable video that is sent to a display interface **4010**. The process of generating user-viewable video from the bitstream representation is sometimes called video decompression. Furthermore,

while certain video processing operations are referred to as “coding” operations or tools, it will be appreciated that the coding tools or operations are used at an encoder and corresponding decoding tools or operations that reverse the results of the coding will be performed by a decoder.

[0215] Examples of a peripheral bus interface or a display interface may include universal serial bus (USB) or high definition multimedia interface (HDMI) or Displayport, and so on. Examples of storage interfaces include serial advanced technology attachment (SATA), peripheral component interconnect (PCI), integrated drive electronics (IDE) interface, and the like. The techniques described in the present document may be embodied in various electronic devices such as mobile phones, laptops, smartphones or other devices that are capable of performing digital data processing and/or video display.

[0216] FIG. **2** is a block diagram of an example video processing apparatus **4100**. The apparatus **4100** may be used to implement one or more of the methods described herein. The apparatus **4100** may be embodied in a smartphone, tablet, computer, Internet of Things (IoT) receiver, and so on. The apparatus **4100** may include one or more processors **4102**, one or more memories **4104** and video processing circuitry **4106**. The processor(s) **4102** may be configured to implement one or more methods described in the present document. The memory (memories) **4104** may be used for storing data and code used for implementing the methods and techniques described herein. The video processing circuitry **4106** may be used to implement, in hardware circuitry, some techniques described in the present document. In some embodiments, the video processing circuitry **4106** may be at least partly included in the processor **4102**, e.g., a graphics co-processor.

[0217] FIG. **3** is a flowchart for an example method **4200** of video processing. The method **4200** includes determining that when any two tracks are included in a same alternate group, the two tracks shall not both have a PreselectionGroupBox with a same value of track_group_id at step **4202**. A conversion is performed between a visual media data and the media data file based on the track_group_id at step **4204**. The conversion of step **4204** may include encoding at an encoder or decoding at a decoder, depending on the example.

[0218] It should be noted that the method **4200** can be implemented in an apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, such as video encoder **4400**, video decoder **4500**, and/or encoder **4600**. In such a case, the instructions upon execution by the processor, cause the processor to perform the method **4200**. Further, the method **4200** can be performed by a non-transitory computer readable medium comprising a computer program product for use by a video coding device. The computer program product comprises computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method **4200**.

[0219] FIG. **4** is a block diagram that illustrates an example video coding system **4300** that may utilize the techniques of this disclosure. The video coding system **4300** may include a source device **4310** and a destination device **4320**. Source device **4310** generates encoded video data which may be referred to as a video encoding device. Destination device **4320** may decode the encoded video data generated by source device **4310** which may be referred to as a video decoding device.

[0220] Source device **4310** may include a video source **4312**, a video encoder **4314**, and an input/output (I/O) interface **4316**. Video source **4312** may include a source such as a video capture device, an interface to receive video data from a video content provider, and/or a computer graphics system for generating video data, or a combination of such sources. The video data may comprise one or more pictures. Video encoder **4314** encodes the video data from video source **4312** to generate a bitstream. The bitstream may include a sequence of bits that form a coded representation of the video data. The bitstream may include coded pictures and associated data. The coded picture is a coded representation of a picture. The associated data may include sequence parameter sets, picture parameter sets, and other syntax structures. I/O interface **4316** may include a modulator/demodulator (modem) and/or a transmitter. The encoded video data may be transmitted

directly to destination device **4320** via I/O interface **4316** through network **4330**. The encoded video data may also be stored onto a storage medium/server **4340** for access by destination device **4320**.

[0221] Destination device **4320** may include an I/O interface **4326**, a video decoder **4324**, and a display device **4322**. I/O interface **4326** may include a receiver and/or a modem. I/O interface **4326** may acquire encoded video data from the source device **4310** or the storage medium/server **4340**. Video decoder **4324** may decode the encoded video data. Display device **4322** may display the decoded video data to a user. Display device **4322** may be integrated with the destination device **4320**, or may be external to destination device **4320**, which can be configured to interface with an external display device.

[0222] Video encoder **4314** and video decoder **4324** may operate according to a video compression standard, such as the High Efficiency Video Coding (HEVC) standard, Versatile Video Coding (VVM) standard and other current and/or further standards.

[0223] FIG. **5** is a block diagram illustrating an example of video encoder **4400**, which may be video encoder **4314** in the system **4300** illustrated in FIG. **4**. Video encoder **4400** may be configured to perform any or all of the techniques of this disclosure. The video encoder **4400** includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of video encoder **4400**. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.

[0224] The functional components of video encoder **4400** may include a partition unit **4401**, a prediction unit **4402** which may include a mode select unit **4403**, a motion estimation unit **4404**, a motion compensation unit **4405**, an intra prediction unit **4406**, a residual generation unit **4407**, a transform processing unit **4408**, a quantization unit **4409**, an inverse quantization unit **4410**, an inverse transform unit **4411**, a reconstruction unit **4412**, a buffer **4413**, and an entropy encoding unit **4414**.

[0225] In other examples, video encoder **4400** may include more, fewer, or different functional components. In an example, prediction unit **4402** may include an intra block copy (IBC) unit. The IBC unit may perform prediction in an IBC mode in which at least one reference picture is a picture where the current video block is located.

[0226] Furthermore, some components, such as motion estimation unit **4404** and motion compensation unit **4405** may be highly integrated, but are represented in the example of video encoder **4400** separately for purposes of explanation.

[0227] Partition unit **4401** may partition a picture into one or more video blocks. Video encoder **4400** and video decoder **4500** may support various video block sizes.

[0228] Mode select unit **4403** may select one of the coding modes, intra or inter, e.g., based on error results, and provide the resulting intra or inter coded block to a residual generation unit **4407** to generate residual block data and to a reconstruction unit **4412** to reconstruct the encoded block for use as a reference picture. In some examples, mode select unit **4403** may select a combination of intra and inter prediction (CIIP) mode in which the prediction is based on an inter prediction signal and an intra prediction signal. Mode select unit **4403** may also select a resolution for a motion vector (e.g., a sub-pixel or integer pixel precision) for the block in the case of inter prediction.

[0229] To perform inter prediction on a current video block, motion estimation unit **4404** may generate motion information for the current video block by comparing one or more reference frames from buffer **4413** to the current video block. Motion compensation unit **4405** may determine a predicted video block for the current video block based on the motion information and decoded samples of pictures from buffer **4413** other than the picture associated with the current video block.

[0230] Motion estimation unit **4404** and motion compensation unit **4405** may perform different operations for a current video block, for example, depending on whether the current video block is in an I slice, a P slice, or a B slice.

[0231] In some examples, motion estimation unit **4404** may perform uni-directional prediction for the current video block, and motion estimation unit **4404** may search reference pictures of list 0 or list 1 for a reference video block for the current video block. Motion estimation unit **4404** may then generate a reference index that indicates the reference picture in list 0 or list 1 that contains the reference video block and a motion vector that indicates a spatial displacement between the current video block and the reference video block. Motion estimation unit **4404** may output the reference index, a prediction direction indicator, and the motion vector as the motion information of the current video block. Motion compensation unit **4405** may generate the predicted video block of the current block based on the reference video block indicated by the motion information of the current video block.
[0232] In other examples, motion estimation unit **4404** may perform bi-directional prediction for the current video block, motion estimation unit **4404** may search the reference pictures in list 0 for a reference video block for the current video block and may also search the reference pictures in list 1 for another reference video block for the current video block. Motion estimation unit **4404** may then generate reference indexes that indicate the reference pictures in list 0 and list 1 containing the reference video blocks and motion vectors that indicate spatial displacements between the reference video blocks and the current video block. Motion estimation unit **4404** may output the reference indexes and the motion vectors of the current video block as the motion information of the current video block. Motion compensation unit **4405** may generate the predicted video block of the current video block based on the reference video blocks indicated by the motion information of the current video block.
[0233] In some examples, motion estimation unit **4404** may output a full set of motion information for decoding processing of a decoder. In some examples, motion estimation unit **4404** may not output a full set of motion information for the current video. Rather, motion estimation unit **4404** may signal the motion information of the current video block with reference to the motion information of another video block. For example, motion estimation unit **4404** may determine that the motion information of the current video block is sufficiently similar to the motion information of a neighboring video block.
[0234] In one example, motion estimation unit **4404** may indicate, in a syntax structure associated with the current video block, a value that indicates to the video decoder **4500** that the current video block has the same motion information as another video block.
[0235] In another example, motion estimation unit **4404** may identify, in a syntax structure associated with the current video block, another video block and a motion vector difference (MVD). The motion vector difference indicates a difference between the motion vector of the current video block and the motion vector of the indicated video block. The video decoder **4500** may use the motion vector of the indicated video block and the motion vector difference to determine the motion vector of the current video block.
[0236] As discussed above, video encoder **4400** may predictively signal the motion vector. Two examples of predictive signaling techniques that may be implemented by video encoder **4400** include advanced motion vector prediction (AMVP) and merge mode signaling.
[0237] Intra prediction unit **4406** may perform intra prediction on the current video block. When intra prediction unit **4406** performs intra prediction on the current video block, intra prediction unit **4406** may generate prediction data for the current video block based on decoded samples of other video blocks in the same picture. The prediction data for the current video block may include a predicted video block and various syntax elements.
[0238] Residual generation unit **4407** may generate residual data for the current video block by subtracting the predicted video block(s) of the current video block from the current video block. The residual data of the current video block may include residual video blocks that correspond to different sample components of the samples in the current video block.
[0239] In other examples, there may be no residual data for the current video block for the current

video block, for example in a skip mode, and residual generation unit **4407** may not perform the subtracting operation.
[0240] Transform processing unit **4408** may generate one or more transform coefficient video blocks for the current video block by applying one or more transforms to a residual video block associated with the current video block.
[0241] After transform processing unit **4408** generates a transform coefficient video block associated with the current video block, quantization unit **4409** may quantize the transform coefficient video block associated with the current video block based on one or more quantization parameter (QP) values associated with the current video block.
[0242] Inverse quantization unit **4410** and inverse transform unit **4411** may apply inverse quantization and inverse transforms to the transform coefficient video block, respectively, to reconstruct a residual video block from the transform coefficient video block. Reconstruction unit **4412** may add the reconstructed residual video block to corresponding samples from one or more predicted video blocks generated by the prediction unit **4402** to produce a reconstructed video block associated with the current block for storage in the buffer **4413**.
[0243] After reconstruction unit **4412** reconstructs the video block, the loop filtering operation may be performed to reduce video blocking artifacts in the video block.
[0244] Entropy encoding unit **4414** may receive data from other functional components of the video encoder **4400**. When entropy encoding unit **4414** receives the data, entropy encoding unit **4414** may perform one or more entropy encoding operations to generate entropy encoded data and output a bitstream that includes the entropy encoded data.
[0245] FIG. **6** is a block diagram illustrating an example of video decoder **4500** which may be video decoder **4324** in the system **4300** illustrated in FIG. **4**. The video decoder **4500** may be configured to perform any or all of the techniques of this disclosure. In the example shown, the video decoder **4500** includes a plurality of functional components. The techniques described in this disclosure may be shared among the various components of the video decoder **4500**. In some examples, a processor may be configured to perform any or all of the techniques described in this disclosure.
[0246] In the example shown, video decoder **4500** includes an entropy decoding unit **4501**, a motion compensation unit **4502**, an intra prediction unit **4503**, an inverse quantization unit **4504**, an inverse transformation unit **4505**, a reconstruction unit **4506**, and a buffer **4507**. Video decoder **4500** may, in some examples, perform a decoding pass generally reciprocal to the encoding pass described with respect to video encoder **4400**.
[0247] Entropy decoding unit **4501** may retrieve an encoded bitstream. The encoded bitstream may include entropy coded video data (e.g., encoded blocks of video data). Entropy decoding unit **4501** may decode the entropy coded video data, and from the entropy decoded video data, motion compensation unit **4502** may determine motion information including motion vectors, motion vector precision, reference picture list indexes, and other motion information. Motion compensation unit **4502** may, for example, determine such information by performing the AMVP and merge mode.
[0248] Motion compensation unit **4502** may produce motion compensated blocks, possibly performing interpolation based on interpolation filters. Identifiers for interpolation filters to be used with sub-pixel precision may be included in the syntax elements.
[0249] Motion compensation unit **4502** may use interpolation filters as used by video encoder **4400** during encoding of the video block to calculate interpolated values for sub-integer pixels of a reference block. Motion compensation unit **4502** may determine the interpolation filters used by video encoder **4400** according to received syntax information and use the interpolation filters to produce predictive blocks.
[0250] Motion compensation unit **4502** may use some of the syntax information to determine sizes of blocks used to encode frame(s) and/or slice(s) of the encoded video sequence, partition

information that describes how each macroblock of a picture of the encoded video sequence is partitioned, modes indicating how each partition is encoded, one or more reference frames (and reference frame lists) for each inter coded block, and other information to decode the encoded video sequence.

[0251] Intra prediction unit **4503** may use intra prediction modes for example received in the bitstream to form a prediction block from spatially adjacent blocks. Inverse quantization unit **4504** inverse quantizes, i.e., de-quantizes, the quantized video block coefficients provided in the bitstream and decoded by entropy decoding unit **4501**. Inverse transform unit **4505** applies an inverse transform.

[0252] Reconstruction unit **4506** may sum the residual blocks with the corresponding prediction blocks generated by motion compensation unit **4502** or intra prediction unit **4503** to form decoded blocks. If desired, a deblocking filter may also be applied to filter the decoded blocks in order to remove blockiness artifacts. The decoded video blocks are then stored in buffer **4507**, which provides reference blocks for subsequent motion compensation/intra prediction and also produces decoded video for presentation on a display device.

[0253] FIG. **7** is a schematic diagram of an example encoder **4600**. The encoder **4600** is suitable for implementing the techniques of VVC. The encoder **4600** includes three in-loop filters, namely a deblocking filter (DF) **4602**, a sample adaptive offset (SAO) **4604**, and an adaptive loop filter (ALF) **4606**. Unlike the DF **4602**, which uses predefined filters, the SAO **4604** and the ALF **4606** utilize the original samples of the current picture to reduce the mean square errors between the original samples and the reconstructed samples by adding an offset and by applying a finite impulse response (FIR) filter, respectively, with coded side information signaling the offsets and filter coefficients. The ALF **4606** is located at the last processing stage of each picture and can be regarded as a tool trying to catch and fix artifacts created by the previous stages.

[0254] The encoder **4600** further includes an intra prediction component **4608** and a motion estimation/compensation (ME/MC) component **4610** configured to receive input video. The intra prediction component **4608** is configured to perform intra prediction, while the ME/MC component **4610** is configured to utilize reference pictures obtained from a reference picture buffer **4612** to perform inter prediction. Residual blocks from inter prediction or intra prediction are fed into a transform (T) component **4614** and a quantization (Q) component **4616** to generate quantized residual transform coefficients, which are fed into an entropy coding component **4618**. The entropy coding component **4618** entropy codes the prediction results and the quantized transform coefficients and transmits the same toward a video decoder (not shown). Quantization components output from the quantization component **4616** may be fed into an inverse quantization (IQ) components **4620**, an inverse transform component **4622**, and a reconstruction (REC) component **4624**. The REC component **4624** is able to output images to the DF **4602**, the SAO **4604**, and the ALF **4606** for filtering prior to those images being stored in the reference picture buffer **4612**.

[0255] FIG. **8** is a flowchart for an example method **4700** of video processing. The method **4700** includes determining information from a PreselectionGroupBox at step **4702**. A segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof. A conversion is performed between a visual media data and a bitstream based on the PreselectionGroupBox at step **4704**.

[0256] It should be noted that the method **4700** can be implemented in an apparatus for processing video data comprising a processor and a non-transitory memory with instructions thereon, such as video encoder **4400**, video decoder **4500**, and/or encoder **4600**. In such a case, the instructions upon execution by the processor, cause the processor to perform the method **4200**. Further, the method **4700** can be performed by a non-transitory computer readable medium comprising a computer program product for use by a video coding device. The computer program product comprises computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method **4700**.

[0257] A listing of solutions preferred by some examples is provided next.
[0258] The following solutions show examples of techniques discussed herein. [0259] 1. A method for processing video data comprising: determining that when any two tracks are included in a same alternate group, the two tracks shall not both have a PreselectionGroupBox with a same value of track_group_id; and performing a conversion between a visual media data and the media data file based on the track_group_id. [0260] 2. The method of solution 1, wherein when processing a preselection to which a track contributes, that track can be replaced by any other track in a same alternate group including the track. [0261] 3. The method of any of solutions 1-2, wherein an indication of a main track for a preselection is signalled. [0262] 4. The method of any of solutions 1-3, the indication is signaled by a field in a track contributing to the preselection. [0263] 5. The method of any of solutions 1-4, wherein the field is a one-bit flag. [0264] 6. The method of any of solutions 1-5, wherein the flag equal to 1 indicates that the track is the main track in the preselection, and wherein the flag equal to 0 indicates that the track is not the main track in the preselection. [0265] 7. The method of any of solutions 1-6, wherein a rule requires that one and only one track among all tracks contributing to the preselection shall have a flag equal to 1. [0266] 8. The method of any of solutions 1-7, wherein the indication is signalled through the track_order field. [0267] 9. The method of any of solutions 1-8, wherein the main track has a lowest value of track_order. [0268] 10. The method of any of solutions 1-9, wherein when multiple tracks contribute to a presentation having the smallest value of track_order, a main track can be any of the multiple tracks. [0269] 11. The method of any of solutions 1-10, wherein when a track_order field is not present for a track contributing to a preselection, the value of track_order is inferred to be equal to 0. [0270] 12. The method of any of solutions 1-11, wherein when media content is encapsulated as a dynamic adaptive streaming over hypertext transfer protocol (DASH) media presentation, an adaptation set containing a representation for a main track in a preselection is a main adaptation set for the DASH media preselection. [0271] 13. The method of any of solutions 1-12, wherein for any preselection, when at least one contributing track has a track_in_movie flag equal to 1, a PreselectionTrackGroupEntryBox shall be present for the preselection, and wherein a presence of the PreselectionTrackGroupEntryBox for the preselection is otherwise optional. [0272] 14. The method of any of solutions 1-13, wherein a preselection_tag field is not present in a Preselection TrackGroupEntryBox. [0273] 15. The method of any of solutions 1-14, wherein a segment_order field is not present in a Preselection TrackGroupEntryBox. [0274] 16. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of solutions 1-15. [0275] 17. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of solutions 1-15. [0276] 18. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining that when any two tracks are included in a same alternate group, the two tracks shall not both have a PreselectionGroupBox with a same value of track_group_id; and generating a bitstream based on the determining. [0277] 19. A method for storing bitstream of a video comprising: determining that when any two tracks are included in a same alternate group, the two tracks shall not both have a PreselectionGroupBox with a same value of track_group_id; generating a bitstream based on the determining; and storing the bitstream in a non-transitory computer-readable recording medium. [0278] 20. A method, apparatus or system described in the present document.
[0279] The following solutions show further examples of techniques discussed herein. [0280] 1. A method for processing video data comprising: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox

and sub-boxes thereof; and performing a conversion between a visual media data and the media data file based on the PreselectionGroupBox. [0281] 2. The method of solution 1, wherein when any two tracks are included in a same alternate group, the two tracks shall not both have a PreselectionGroupBox with a same value of track_group_id. [0282] 3. The method of any of solutions 1-2, wherein when processing a preselection to which a track contributes, the track can be replaced by another track in a same alternate group including the track. [0283] 4. The method of any of solutions 1-3, wherein an indication of a main track for a preselection is signalled. [0284] 5. The method of any of solutions 1-4, wherein the indication is signaled by a field in a track contributing to the preselection. [0285] 6. The method of any of solutions 1-5, wherein the field is a one-bit flag. [0286] 7. The method of any of solutions 1-6, wherein the flag equal to 1 indicates that the track is the main track in the preselection, and wherein the flag equal to 0 indicates that the track is not the main track in the preselection. [0287] 8. The method of any of solutions 1-7, wherein one and only one track among all tracks contributing to the preselection shall have a flag equal to 1. [0288] 9. The method of any of solutions 1-8, wherein the indication is signalled through the track_order field. [0289] 10. The method of any of solutions 1-9, wherein the main track has a lowest value of track_order. [0290] 11. The method of any of solutions 1-10, wherein when multiple tracks contribute to a presentation having a smallest value of track_order, any of the multiple tracks can be a main track. [0291] 12. The method of any of solutions 1-11, wherein when a track_order field is not present for a track contributing to a preselection, a value of track_order is inferred to be equal to 0. [0292] 13 The method of any of solutions 1-12, wherein when media content is encapsulated as a dynamic adaptive streaming over hypertext transfer protocol (DASH) media presentation, an adaptation set containing a representation for a main track in a preselection is a main adaptation set for a DASH media preselection. [0293] 14. The method of any of solutions 1-13, wherein for any preselection, when at least one contributing track in a file has a track_in_movie flag equal to 1, a Preselection TrackGroupEntryBox shall be present in the file for the preselection, and wherein a presence of the PreselectionTrackGroupEntryBox for the preselection is otherwise optional. [0294] 15. The method of any of solutions 1-14, wherein a preselection_tag field is not present in a Preselection TrackGroupEntryBox. [0295] 16. The method of any of solutions 1-15, wherein the conversion includes encoding the visual media data into the visual media data file. [0296] 17. The method of any of solutions 1-15, wherein the conversion includes decoding the visual media data from the visual media data file. [0297] 18. An apparatus for processing video data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to perform the method of any of solutions 1-17. [0298] 19. A non-transitory computer readable medium comprising a computer program product for use by a video coding device, the computer program product comprising computer executable instructions stored on the non-transitory computer readable medium such that when executed by a processor cause the video coding device to perform the method of any of solutions 1-17. [0299] 20. A non-transitory computer-readable recording medium storing a bitstream of a video which is generated by a method performed by a video processing apparatus, wherein the method comprises: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof; and generating a bitstream based on the determining. [0300] 21. A method for storing bitstream of a video comprising: determining information from a PreselectionGroupBox, wherein a segment_order field is omitted from the PreselectionGroupBox and sub-boxes thereof; generating a bitstream based on the determining; and storing the bitstream in a non-transitory computer-readable recording medium.

[0301] In the solutions described herein, an encoder may conform to the format rule by producing a coded representation according to the format rule. In the solutions described herein, a decoder may use the format rule to parse syntax elements in the coded representation with the knowledge of presence and absence of syntax elements according to the format rule to produce decoded video.

[0302] In the present document, the term “video processing” may refer to video encoding, video decoding, video compression or video decompression. For example, video compression algorithms may be applied during conversion from pixel representation of a video to a corresponding bitstream representation or vice versa. The bitstream representation of a current video block may, for example, correspond to bits that are either co-located or spread in different places within the bitstream, as is defined by the syntax. For example, a macroblock may be encoded in terms of transformed and coded error residual values and also using bits in headers and other fields in the bitstream. Furthermore, during conversion, a decoder may parse a bitstream with the knowledge that some fields may be present, or absent, based on the determination, as is described in the above solutions. Similarly, an encoder may determine that certain syntax fields are or are not to be included and generate the coded representation accordingly by including or excluding the syntax fields from the coded representation.

[0303] The disclosed and other solutions, examples, embodiments, modules and the functional operations described in this document can be implemented in digital electronic circuitry, or in computer software, firmware, or hardware, including the structures disclosed in this document and their structural equivalents, or in combinations of one or more of them. The disclosed and other embodiments can be implemented as one or more computer program products, i.e., one or more modules of computer program instructions encoded on a computer readable medium for execution by, or to control the operation of, data processing apparatus. The computer readable medium can be a machine-readable storage device, a machine-readable storage substrate, a memory device, a composition of matter effecting a machine-readable propagated signal, or a combination of one or more them. The term “data processing apparatus” encompasses all apparatus, devices, and machines for processing data, including by way of example a programmable processor, a computer, or multiple processors or computers. The apparatus can include, in addition to hardware, code that creates an execution environment for the computer program in question, e.g., code that constitutes processor firmware, a protocol stack, a database management system, an operating system, or a combination of one or more of them. A propagated signal is an artificially generated signal, e.g., a machine-generated electrical, optical, or electromagnetic signal, that is generated to encode information for transmission to suitable receiver apparatus.

[0304] A computer program (also known as a program, software, software application, script, or code) can be written in any form of programming language, including compiled or interpreted languages, and it can be deployed in any form, including as a stand-alone program or as a module, component, subroutine, or other unit suitable for use in a computing environment. A computer program does not necessarily correspond to a file in a file system. A program can be stored in a portion of a file that holds other programs or data (e.g., one or more scripts stored in a markup language document), in a single file dedicated to the program in question, or in multiple coordinated files (e.g., files that store one or more modules, sub programs, or portions of code). A computer program can be deployed to be executed on one computer or on multiple computers that are located at one site or distributed across multiple sites and interconnected by a communication network.

[0305] The processes and logic flows described in this document can be performed by one or more programmable processors executing one or more computer programs to perform functions by operating on input data and generating output. The processes and logic flows can also be performed by, and apparatus can also be implemented as, special purpose logic circuitry, e.g., an field programmable gate array (FPGA) or an application specific integrated circuit (ASIC).

[0306] Processors suitable for the execution of a computer program include, by way of example, both general and special purpose microprocessors, and any one or more processors of any kind of digital computer. Generally, a processor will receive instructions and data from a read only memory or a random-access memory or both. The essential elements of a computer are a processor for performing instructions and one or more memory devices for storing instructions and data.

Generally, a computer will also include, or be operatively coupled to receive data from or transfer data to, or both, one or more mass storage devices for storing data, e.g., magnetic, magneto optical disks, or optical disks. However, a computer need not have such devices. Computer readable media suitable for storing computer program instructions and data include all forms of non-volatile memory, media and memory devices, including by way of example semiconductor memory devices, e.g., erasable programmable read-only memory (EPROM), electrically erasable programmable read-only memory (EEPROM), and flash memory devices; magnetic disks, e.g., internal hard disks or removable disks; magneto optical disks; and compact disc read-only memory (CD ROM) and Digital versatile disc-read only memory (DVD-ROM) disks. The processor and the memory can be supplemented by, or incorporated in, special purpose logic circuitry.

[0307] While this patent document contains many specifics, these should not be construed as limitations on the scope of any subject matter or of what may be claimed, but rather as descriptions of features that may be specific to particular embodiments of particular techniques. Certain features that are described in this patent document in the context of separate embodiments can also be implemented in combination in a single embodiment. Conversely, various features that are described in the context of a single embodiment can also be implemented in multiple embodiments separately or in any suitable subcombination. Moreover, although features may be described above as acting in certain combinations and even initially claimed as such, one or more features from a claimed combination can in some cases be excised from the combination, and the claimed combination may be directed to a subcombination or variation of a subcombination.

[0308] Similarly, while operations are depicted in the drawings in a particular order, this should not be understood as requiring that such operations be performed in the particular order shown or in sequential order, or that all illustrated operations be performed, to achieve desirable results. Moreover, the separation of various system components in the embodiments described in this patent document should not be understood as requiring such separation in all embodiments.

[0309] Only a few implementations and examples are described and other implementations, enhancements and variations can be made based on what is described and illustrated in this patent document.

[0310] A first component is directly coupled to a second component when there are no intervening components, except for a line, a trace, or another medium between the first component and the second component. The first component is indirectly coupled to the second component when there are intervening components other than a line, a trace, or another medium between the first component and the second component. The term “coupled” and its variants include both directly coupled and indirectly coupled. The use of the term “about” means a range including ±10% of the subsequent number unless otherwise stated.

[0311] While several embodiments have been provided in the present disclosure, it should be understood that the disclosed systems and methods might be embodied in many other specific forms without departing from the spirit or scope of the present disclosure. The present examples are to be considered as illustrative and not restrictive, and the intention is not to be limited to the details given herein. For example, the various elements or components may be combined or integrated in another system or certain features may be omitted, or not implemented.

[0312] In addition, techniques, systems, subsystems, and methods described and illustrated in the various embodiments as discrete or separate may be combined or integrated with other systems, modules, techniques, or methods without departing from the scope of the present disclosure. Other items shown or discussed as coupled may be directly connected or may be indirectly coupled or communicating through some interface, device, or intermediate component whether electrically, mechanically, or otherwise. Other examples of changes, substitutions, and alterations are ascertainable by one skilled in the art and could be made without departing from the spirit and scope disclosed herein.

Claims

1. A method for processing media data, comprising: performing a conversion between the media data and a media data file according to a format rule, wherein the format rule specifies that a field indicating an order rule of segments is omitted from a preselection track group entry box for describing a preselection.
2. The method of claim 1, wherein attributes signalled in the preselection track group entry box take precedence over attributes signalled in contributing tracks.
3. The method of claim 1, wherein the preselection track group entry box shall describe only track groups identified by a track group type equal to ‘prse’.
4. The method of claim 1, wherein the format rule further specifies that for any preselection, if at least one contributing track in a file has a flag equal to 1, the preselection track group entry box shall be present in the file for the preselection; otherwise, a presence of the preselection track group entry box for the preselection in the file is optional.
5. The method of claim 4, wherein the flag is a track_in_movie flag.
6. The method of claim 1, wherein the format rule further specifies that all attributes uniquely characterizing the preselection should be present in the preselection track group entry box for the preselection.
7. The method of claim 1, wherein the format rule further specifies that a presence of a track group type box with a track group type equal to ‘pres’ in a track indicates that the track contributes to a preselection.
8. The method of claim 1, wherein the format rule further specifies that all tracks that have a track group with a track group type equal to ‘pres’ and a particular value of a track group identifier are part of a same preselection.
9. The method of claim 8, wherein the particular value of the track group identifier is referred to as an identifier of the preselection.
10. The method of claim 1, wherein the format rule further specifies that when multiple tracks contribute to the preselection, an optionally present preselection processing box provides information on how to process a track containing the preselection processing box in a context of the preselection and relative to other tracks.
11. The method of claim 1, wherein the preselection is a set of one or more tracks representing one version of media presentation for simultaneous decoding or presentation.
12. The method of claim 1, wherein the conversion includes generating the media data file from the media data.
13. The method of claim 1, wherein the conversion includes parsing the media data from the media data file.
14. An apparatus for processing media data comprising: a processor; and a non-transitory memory with instructions thereon, wherein the instructions upon execution by the processor, cause the processor to: perform a conversion between the media data and a media data file according to a format rule, wherein the format rule specifies that a field indicating an order rule of segments is omitted from a preselection track group entry box for describing a preselection.
15. The apparatus of claim 14, wherein attributes signalled in the preselection track group entry box take precedence over attributes signalled in contributing tracks.
16. The apparatus of claim 14, wherein the preselection track group entry box shall describe only track groups identified by a track group type equal to ‘prse’.
17. A non-transitory computer readable storage medium storing instructions that cause a processor to: perform a conversion between media data and a media data file according to a format rule, wherein the format rule specifies that a field indicating an order rule of segments is omitted from a preselection track group entry box for describing a preselection.

18. The non-transitory computer readable storage medium of claim 17, wherein attributes signalled in the preselection track group entry box take precedence over attributes signalled in contributing tracks.

19. A non-transitory computer-readable recording medium storing a media data file which is generated by a method performed by a video processing apparatus, wherein the method comprises: generating the media data file according to a format rule, wherein the format rule specifies that a field indicating an order rule of segments is omitted from a preselection track group entry box for describing a preselection.

20. The non-transitory computer-readable recording medium of claim 19, wherein attributes signalled in the preselection track group entry box take precedence over attributes signalled in contributing tracks.