



(51) International Patent Classification:
H04N 7/26 (2006.01)

(21) International Application Number:
PCT/EP2013/057803

(22) International Filing Date:
15 April 2013 (15.04.2013)

(25) Filing Language: English

(26) Publication Language: English

(30) Priority Data:
61/624,098 13 April 2012 (13.04.2012) US
61/666,185 29 June 2012 (29.06.2012) US

(71) Applicant: **FRAUNHOFER-GESELLSCHAFT ZUR FÖRDERUNG DER ANGEWANDTEN FORSCHUNG E.V.** [DE/DE]; Hansastraße 27c, 80686 München (DE).

(72) Inventors: **SCHIERL, Thomas**; Dunckerstraße 72, 10437 Berlin (DE). **GEORGE, Valeri**; John-Sieg-Straße 24, 10365 Berlin (DE). **HENKEL, Anastasia**; Fahrenheitstraße 8, 10245 Berlin (DE). **MARPE, Detlev**; Südwestkorso 70, 12161 Berlin (DE). **GRÜNEBERG, Karsten**; Adickesstraße 43, 13599 Berlin (DE). **SKUPIN, Robert**; Schleswiger Ufer 5, 10555 Berlin (DE).

(74) Agents: **SCHENK, Markus** et al.; Schoppe, Zimmermann, Stöckeler & Zinkler, P.O. Box 246, 82043 Pullach (DE).

(81) Designated States (unless otherwise indicated, for every kind of national protection available): AE, AG, AL, AM, AO, AT, AU, AZ, BA, BB, BG, BH, BN, BR, BW, BY, BZ, CA, CH, CL, CN, CO, CR, CU, CZ, DE, 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, PA, PE, PG, PH, PL, PT, QA, RO, RS, RU, RW, SC, SD, SE, SG, SK, SL, SM, ST, SV, SY, TH, TJ, TM, TN, TR, TT, TZ, UA, UG, US, UZ, VC, VN, ZA, ZM, ZW.

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

Published:

— without international search report and to be republished upon receipt of that report (Rule 48.2(g))

(54) Title: SCALABLE DATA STREAM AND NETWORK ENTITY

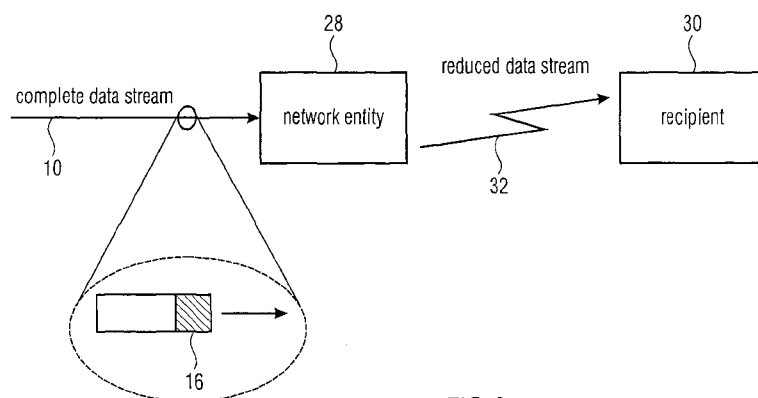


FIG 2

(57) Abstract: The handling of scalable data streams by network entities is rendered less complex by, in the data stream, accompanying packets which actually carry data by packets of a different packet type which have a scalability axes descriptor defining the number of scalability axes and a semantic meaning thereof. By this measure, it is sufficient if the data carrying packets comprise an operation point identifier identifying an operation point associated with a respective packet within the scalability space spanned by the scalability axis. By this measure, the packet operation point identifiers provide enough information to a network entity so as to associate these packets with their operation point in the scalability space provided the network entity has access to this scalability axes descriptor provided by the packet of the predetermined packet type. In another aspect, the handling of scalable data streams by network entities is rendered more efficient by conveying level and/or profile descriptors using packets other than the data carrying packets, with profile and/or level descriptors being divided up into a first set explicitly signaling the definition of the available coding option set and/or available syntax element value range for a respective operation point, and a second set of profile and/or level descriptors which signal the definition of the available coding option set and/or available syntax element value range for their respective operation points by reference another profile and/or level descriptor.



Scalable Data Stream and Network Entity

Description

5

The present invention is concerned with scalable data streams such as scalable video streams and network entities dealing with such scalable data streams such as, for example, a decoder or MANE (media-aware network element).

10

From a transport perspective, adaptation of video bitstreams in temporal or other dimensions is highly desirable, as was already identified and addressed within the standardization of H.264/AVC. The encapsulation of video data into Network Abstraction Layer (NAL) units and the design decision to signal many important but rather invariant parameters outside the video bitstream in so called Parameter Sets reflect this understanding. The Scalable Video Coding (SVC) and Multiview Video Coding (MVC) extensions of H.264/AVC allow for adaptation beyond the temporal dimension but a lack of extensibility in the base specification of H.264/AVC led to complicated approaches to extend the H.264/AVC high level syntax in a backward compatible way. In H.264/AVC and its extensions the signaling of operation points along the scalability dimensions is done by means of NAL units header extensions that include an extra byte for this purpose. The same information is provided by the so-called prefix NAL units for NAL units that contain pure H.264/AVC video data and constitute the so-called base layer. A mechanism to extend the information provided via Sequence Parameter Set (SPS) of the base layer for the enhanced layers, coded by the extensions of H.264/AVC, is established via so-called Subset Sequence Parameter Sets (SSPS).

25

While the development of the HEVC base specification is still ongoing, efforts towards a 3D video coding extension are already taken in order to assure an extensible syntax in the base specification in the beginning. These so-called scalable hooks in the base specification need to be designed carefully to be future-proof. The following paragraphs give an overview on the current status of the HEVC High Level (HL) syntax and the concepts that are under discussion at the moment.

30

35 The current status of the HEVC standardization is as follows:

During the ongoing HEVC base specification and the 3D extensions, numerous participants made proposals on how to proceed from the HL syntax as specified in

H.264/AVC. The outcome is reflected in the current working draft of the specification and the numerous contributions of the individual participants. The following paragraphs give an overview on the current discussion.

- 5 As stated above, the signalization of operation points within the scalability dimensions of an SVC or MVC video bitstream requires (specific) extensions of the H.264/AVC NAL unit headers. This is regarded as an unclean solution, causing extra effort e.g. for parsing multiple different NAL unit header structures and requiring prefix NAL units to signal the base layer. Therefore, an effort was made to ensure that the base HEVC NAL unit header
10 syntax is versatile enough to satisfy the needs of the future extensions of the base specification.

Fig. 1 gives the syntax of a NAL unit as in the current working draft. Current consensus was to use a two byte NAL unit header. In the first byte, `nal_ref_flag` is signaled with one bit opposed to the two bits of `nal_ref_idc` in H.264/AVC, as this HL feature has not been
15 widely used in applications. The syntax element `nal_unit_type` therefore has one more bit to signal the type of the NAL unit, which allows for a total of 64 distinguishable types.

The second byte of the NAL unit header is divided into two parts, where 1c bits are used to
20 signal `temporal_id` of the NAL unit as temporal scalability is already enabled in the base specification. The remaining 5 bits of the second byte are reserved to be equal to one within a HEVC conforming bitstream. The current understanding of the usage of the remaining 5 bits is that they can be used to signal scalability identifiers in future extensions, e.g. for a `layer_id` syntax element.

25 While Picture Parameter Set (PPS) and Sequence Parameter Set as defined in the current HEVC base specification are relatively similar to what has formerly been specified in H.264/AVC, two new Parameter Sets, referred to as the Adaptation Parameter Set (APS) and Video Parameter Set (VPS), have been introduced to HEVC of which only the VPS is
30 relevant for the content of this document.

The Video Parameter Set shown in Fig. 2 is supposed to signal parameters such as the number of (e.g. temporal) level/layer present in the video bitstream and the profile and level for all operation points within. Other parameters to be signaled include the
35 dependencies between scalable layers, much as it is signaled in the SVC scalability information SEI messages.

An additional brief explanation is presented below with regard to the semantics of the syntax in Figs. 1 and 2.

5 `profile_idc` and `level_idc` indicate the profile and level to which the coded video sequence conforms.

`max_temporal_layers_minus1 + 1` specifies the maximum number of temporal layers present in the sequence. The value of `max_temporal_layers_minus1` shall be in the range of 0 to 7, inclusive.

10 `more_rbsp_data()` is specified as follows.

- If there is no more data in the RBSP, the return value of `more_rbsp_data()` is equal to FALSE.
- Otherwise, the RBSP data is searched for the last (least significant, right-most) bit equal to 1 that is present in the RBSP. Given the position of this bit, which is the first bit (`rbbsp_stop_one_bit`) of the `rbbsp_trailing_bits()` syntax structure, the following applies.
- If there is more data in an RBSP before the `rbbsp_trailing_bits()` syntax structure, the return value of `more_rbsp_data()` is equal to TRUE.
- 20 – Otherwise, the return value of `more_rbsp_data()` is equal to FALSE.

`nal_ref_flag` equal to 1 specifies that the content of the NAL unit contains a sequence parameter set, a picture parameter set, an adaptation parameter set or a slice of a reference picture.

25 For coded video sequences conforming to one or more of the profiles specified in Annex 10 that are decoded using the decoding process specified in clauses 2-9, `nal_ref_flag` equal to 0 for a NAL unit containing a slice indicates that the slice is part of a non-reference picture.

30 `nal_ref_flag` shall be equal to 1 for sequence parameter set, picture parameter set or adaptation parameter set NAL units. When `nal_ref_flag` is equal to 0 for one NAL unit with `nal_unit_type` equal to 1 or 4 of a particular picture, it shall be equal to 0 for all NAL units with `nal_unit_type` equal to 1 or 4 of the picture.

35 `nal_ref_flag` shall be equal to 1 for NAL units with `nal_unit_type` equal to 5.

`nal_ref_flag` shall be equal to 0 for all NAL units having `nal_unit_type` equal to 6, 9, 10, 11, or 12.

`nal_unit_type` specifies the type of RBSP data structure contained in the NAL unit as specified in Table 1.

- 5 Decoders shall ignore (remove from the bitstream and discard) the contents of all NAL units that use reserved values of `nal_unit_type`.

Table 1 – possible (not exhaustive list of) NAL unit type codes and NAL unit type classes

<code>nal_unit_type</code>	Content of NAL unit and RBSP syntax structure	NAL unit type class
0	Unspecified	non-VCL
1	Coded slice of non-IDR, non-CRA and non-TLA picture <code>slice_layer_rbsp()</code>	VCL
2	Reserved	n/a
3	Coded slice of TLA picture <code>slice_layer_rbsp()</code>	VCL
4	Coded slice of a CRA picture <code>slice_layer_rbsp()</code>	VCL
5	Coded slice of an IDR picture <code>slice_layer_rbsp()</code>	VCL
6	Supplemental enhancement information (SEI) <code>sei_rbsp()</code>	non-VCL
7	Sequence parameter set <code>seq_parameter_set_rbsp()</code>	non-VCL
8	Picture parameter set <code>pic_parameter_set_rbsp()</code>	non-VCL
9	Access unit delimiter <code>access_unit_delimiter_rbsp()</code>	non-VCL
10-11	Reserved	n/a
12	Filler data <code>filler_data_rbsp()</code>	non-VCL
13	Reserved	n/a

14	Adaption parameter set aps_rbsp()	non-VCL
15-23	Reserved	n/a
24..63	Unspecified	non-VCL

A "profile" is a subset of the entire bitstream syntax. Within the bounds imposed by the syntax of a given profile it is still possible to require a very 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. In many applications, it is currently neither practical nor economic to implement a decoder capable of dealing with all hypothetical uses of the syntax within a particular profile.

In order to deal with this problem, "levels" are specified within each profile. A level is a specified set of constraints imposed on values of the syntax elements in the bitstream. These constraints may be simple limits on values. Alternatively they 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).

15

level: A defined set of constraints on the values that may be taken by the *syntax elements* and variables. The same set of levels is defined for all *profiles*, with most aspects of the definition of each level being in common across different *profiles*. Individual implementations may, within specified constraints, support a different level for each supported *profile*. In a different context, level is the value of a *transform coefficient* prior to *scaling*.

20

profile: A specified subset of the syntax.

In the development of the 3D video coding extension for HEVC, there has also been the proposal to shift certain parameters from the slice header to the Access Unit Delimiter (AUD), a NAL unit that optionally stands in the beginning of a new Access Unit (AU) in H.264/AVC.

Another proposal during the course of the specification of the HEVC 3D video coding extension is to signal the dependencies between operation points by means of indirection between SPSs. The NAL unit header is supposed to carry a reference to a SPS and within

30

each SPS is a reference to the relative base SPS. This (potentially cascaded) indirection has to be resolved until the SPS for the lowest (temporal, ...) level. Such an approach puts a high burden on devices such as a MANE to look deep into the bitstream and hold a substantial amount of information available for the purpose of identifying operation points.

5

In any case, it would still be favorable to have a solution at hand which facilitates, or renders more efficient, the handling of scalable data streams by network entities.

This object is achieved by the subject matter of any of the attached independent claims.

10

It is a basic finding of the present invention that the handling of scalable data streams by network entities is rendered less complex if, in the data stream, packets which actually carry data are accompanied by packets of a different packet type which have a scalability axes descriptor defining the number of scalability axes and a semantic meaning thereof. By this measure, it is sufficient if the data carrying packets comprise an operation point identifier identifying an operation point associated with a respective packet within the scalability space spanned by the scalability axis. By this measure, the packet operation point identifiers provide enough information to a network entity so as to associate these packets with their operation point in the scalability space provided the network entity has access to this scalability axes descriptor provided by the packet of the predetermined packet type. In effect, the technique allows for reducing the amount of data to be consumed for the majority of the packets as the operation point identifiers merely have to fulfill an identification task. The scalability axes descriptor may be transmitted less frequently. Concurrently, the network entity also has to perform the cumbersome task of parsing/reading the scalability axes descriptor less frequently, thereby reducing the overhead for the network entity. On the other hand, in accordance with an embodiment of the present invention, the network entity is able to interpret the operation point identifiers at least to the extent so as to distinguish packets of the lowermost operation point in the sequential operation point order, i.e. base layer packets, from other packets, i.e. packets belonging to operation points positioned higher in the sequential operation point order, i.e. "enhancement" layer packets.

25
30

In accordance with another aspect of the present invention which may be combined with a first one or may be implemented individually, it is a basic finding of the present invention that the handling of scalable data streams by network entities may be rendered more efficient if level and/or profile descriptors are conveyed using packets other than the data carrying packets, with profile and/or level descriptors being divided up into a first set explicitly signaling the definition of the available coding option set and/or available syntax

35

element value range for a respective operation point, and a second set of profile and/or level descriptors which signal the definition of the available coding option set and/or available syntax element value range for their respective operation points by reference another profile and/or level descriptor. By this measure, the length of the whole sequence
5 of profile and/or level descriptors is kept lower, thereby saving transmission overhead as well as parsing overhead.

Preferred embodiments of the present application are described below with respect to the figures. Advantageous embodiments are the subject of dependent claims. Among the
10 figures;

- Fig. 1a shows a schematic of a data stream into which a media content is coded;
- Fig. 1b schematically illustrates a composition of scalable data streams;
- 15 Fig. 1c schematically illustrates a composition of scalable data streams in accordance with another example;
- Fig. 2 shows a schematic diagram of a scenario where a network entity receives a
20 complete data stream;.
- Fig. 3a illustrates operation points within a two-dimensional scalability (e.g. temporal and spatial)
- 25 Fig. 3b illustrates a new_profile_level_flag concept
- Fig. 4 shows an NAL unit header syntax for HEVC;
- Fig. 5 shows a VPS syntax;
- 30 Fig. 6 shows a vps_extension_data() syntax;
- Fig. 7 shows a schematic diagram of a data stream in accordance with an example;
- 35 Fig. 8 shows a flow diagram for an example for partitioning the fixed-bit-length syntax element or bit sequence, respectively;

Fig. 9 shows a flow diagram for a mode of operation of an exemplary network entity;

Fig. 10 shows another mode of operation of a network entity;

Fig. 11 shows an NAL unit syntax;

Fig. 12 shows an SPS syntax;

Fig. 13 shows a VPS syntax; and

Fig. 14 shows a VPS extension data syntax.

In order to ease the understanding of the embodiments outlined below, with respect to Figs. 1 and 2 the specifics of scalable data streams and the problems in handling scalable data streams resulting from these specifics are described first. Fig. 1a illustrates a data stream 10 into which a media content is coded such as, for example, audio content, video content, picture content, 3D mesh content or the like. The media content may be time-varying with the data stream representing the media content in a sequence of time stamps such as picture of a video, transform blocks or frames of an audio signal or the like.

As shown in Fig. 1a, the data stream 10 comprises, or is composed of, packets 12. The packets may be equal in size or may differ in size. Each packet comprises a packet type identifier 14. Each packet type identifier 14 identifies a packet type associated with a respective packet 12, out of a plurality of packet types. The plurality of packet types comprises data carrying packet types. In Fig. 1a, for example, the packets within the portion of data stream 10 exemplarily shown in Fig. 1a is exemplarily shown to be of such a data carrying packet type, here exemplarily denoted as "A". Besides such data carrying packet types, there may be other packets of other packet types which do not carry data actually contributing to a coded representation of the media content, but other data such as, for example, supplementary data or information on the origin of the media content or the like.

At least each packet of a data carrying packet type such as A, comprises an operation point identifier 16 identifying an operation point associated with a respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong. To render this clear, reference is made to Fig. 1b. The operation points are

illustrated by circles 18. They are distributed or arranged within the scalability space 20 spanned by the scalability axes, wherein in Fig. 1b, for illustrative purposes, merely two such scalability spaces axes 22a and 22b are shown.

5 Fig. 1b illustrates the composition of scalable data streams such as scalable data stream 10. Packets 12, the operation point identifier of which identifies operation point 1, carry data which together form a coded representation of the media content at that operation point 1. Here, exemplarily, operation point 1 forms a kind of basic representation, i.e. a minimum
10 amount of information concerning the media content is provided by the data of the packets of operation point 1. In case of a video, for example, the data of these packets represent a coded representation of the media content at low spatial resolution and lowest SNR, for example. By combining the data carried by packets belonging to operation point 1 with the data carried by packets 12 of operation point 2 as illustrated by loop line 24, the data thus
15 derived results in a coded representation of the media content, the reconstruction of which somehow exceeds the representation obtained by reconstructed media content from the packets of operation point 1 only, such as for example in SNR or spatial resolution. The characteristic of the media content relating to which the coded representation is extended by adding the data of packets of operation point 2 to the data of the packets of operation point 1 is the semantic meaning of axis 2 in Fig. 1b and the scale of axis 22b is an ordinal
20 scale which sorts the operation points 18 according to this semantic meaning.

In Fig. 1b, for example, axis 22b relates to the spatial resolution, while axis 22a relates to SNR quality. However, other semantic meanings of scalability axes may be available as well. For example, the additional data of packets of a further operation point may add
25 information to the reconstructible version of the media content which was not available before, i.e. before adding this data, such as for example another view, another audio channel, a depth map or the like.

Accordingly, Fig. 1b likewise illustrates by further loop lines 24 that the basic coded
30 representation of the media content may extend along axis 22b as well as along 22a. In other words, the addition of data of packets 12 of a certain operation point to data of packets of another operation point or a combination 24 of data of packets of other operation points leads to a new combination of data which, in accordance with one of the scalability axes 22a or 22b, exceeds the representation obtained without this addition, and
35 this sequential effect is shown by the arrows 26, which accordingly form a sequential operation point order among the operation points 18, which leads through the scalability space 20.

Now, imagine that a network entity receives such a complete data stream. Fig. 2 shows such a scenario. The network entity is shown as a block 28 and is, for example, a decoder, or a MANE. In Fig. 2, network entity 28 receives a complete data stream 10, i.e. a data stream comprising all of the data packets including all data carrying data packets of all operation points. The network entity 28 has the task of providing a recipient 30 with a media content. In Fig. 2, network entity 28 is exemplarily to use a wireless transmission 32 to forward the media content to recipient 30. The network entity 28 is, for example, a mobility management entity. In such a scenario, the network entity 28 has to cope with time-varying transmission capacities to recipient 30. That is, there are time instances where the transmission capacity is not sufficient in order to transfer the complete data stream 10 to recipient 30 to be reproduced in real time, for example. Thus, network entity 28 has to “reduce” the data stream 10 so as to provide recipient 30 with a trimmed down version or representation of the media content. While, for example, the transmission capacity is insufficient in order to transmit the complete data stream 10, the transmission capacity may however be sufficient to transmit data stream 10 with packets 12 of operation point 5 being omitted, for example.

It should be noted that there are other scenarios as well where a network entity 28 has to drop some of the packets 12 in order to meet some externally driven constraints. Insofar, Fig. 2 merely serves as an example.

However, in order to perform the just mentioned task of reducing the data stream, network entity 28 has to inspect at least portions of the data stream. The portions to be inspected include the operation point identifiers 16. This puts a high burden onto network entity 28 and accordingly it would be favorable to keep this overhead low. The embodiments described next achieve this aim.

Another aspect which the embodiments described hereinafter deal with relates to the fact that the capabilities of a decoder allotted for decoding the data stream may increase while extending the sequential operation point order. Low level decoders, for example, may have lower capabilities, i.e. are of lower profile and/or lower level as explained in the introductory portion of the specification. Thus, the adding of some data of packets of a certain operation point may thus be restricted to decoders which meet some minimum profile and/or minimum level criterion and accordingly it would be favorable if network entities such as a decoder or some network entity forwarding the data stream to a decoder would be able to understand the profile and/or level requirements associated with the operation points 18 in a simple manner, but on the other hand with a low rate consumption.

Before starting the description of these embodiments, however, the following is noted. In particular, the nomenclature applied in the embodiments described next differs from the wording used in the description referring to Figs. 1a to 2. To be more specific, the embodiments described next represent specific implementations of embodiments of the present application tailored to be applied in the upcoming HEVC standard. Accordingly, the above mentioned packets are denoted below as NAL units, the NAL unit header of which comprises an operation point identifier which, in accordance with embodiments relating to a first aspect of the present application, are further specified by a scalability axes descriptor in a certain type of NAL unit, namely VPS NAL units. However, the specific embodiments described now are readily transferrable to other types of codecs. Insofar, the following embodiments shall be understood merely as illustrative.

Similarly, although Fig. 1b is suggestive of combinations 24 merely forming proper subsets to each other so that each combination is either a proper subset of all other combinations 24, one of a proper subset or a proper superset of the other combinations 24 or a superset of all other combinations 24 is optional. Fig. 1c, for example, shows a case where the sequentiality of the operation point 18, i.e. the property to have a sequential operation point order defined thereamong, is to be understood as encompassing cases where the combinations also comprise the case where one combination is neither a proper subset nor a proper superset of the other combination. In Fig. 1c, for example, the combination 24 resulting from adding the data of packets of operation point 4 to the data of the combination of packets of operation points 1 and 2 overlaps with the combination 24 resulting from adding the data of packets of operation point 1c to the combination of data of packets of operation points 1 and 2 merely partially, with the intersection being formed by the latter combination, i.e. the combination of operation points 1 and 2. Thereby, the sequential operation point order comprises a branch at operation point 2.

The following embodiment presents a high level syntax including an NAL unit header and a VPS syntax that allows for signaling of scalability information in a way to be useful in video decoders, video bitstream extractors or network devices for adaptation. The NAL unit header includes a scalability identifier that allows signaling up to three dimensions of scalability.

Before describing the VPS syntax in further detail, an overview is presented with respect to Fig. 3a and 3b. The concepts may be used, for example, in the HEVC standard.

The **new_profile_level_flag**, present below, is used to signal that a certain operation point within the scalability dimensions of the video (e.g. temporal, spatial or any other) uses a different profile or level than the formerly described operation points. Fig. 3a shows the different operation points within a spatial and temporal scalable video sequence according to their scalable identifiers (temporal_id and layer_id). Fig. 3a illustrates operation points within two-dimensional scalability (e.g. temporal and spatial).

The Profile and Level needed to describe decoders capabilities in terms of supported coding tools, macroblock throughput and so on, might change with every operation point, which strongly depends on the scalability in use. In loops that signal profile and level or further parameters of operation points, the **new_profile_level_flag** is used to signal whether a new profile and level with respect to the formerly described operation points is present or to that the profile and level of a specific operation point has to be inherited from a formerly described ones by explicit reference.

Another aspect described below relates to an adaptable list of profile and level syntax elements. Profile, Level and optionally other parameters of operation points in parameter sets are given via while loops that allow for straight-forward adaptation, i.e. the exclusion or cut out of certain operation points as well as the adding thereof without changing other syntax elements within the given syntax, i.e. within the VPS. This is achieved by using the more_rbsp_data() syntax element that allows for a while loop which has no necessity for signaling the absolute number of layers and thereby eases the adaptation of parameter sets. Fig. 3b illustrates a new_profile_level_flag concept.

Another aspect described below relates to **Layer scenarios**. Layer scenario represents a ways in which to signal a interpretation of the five reserved bits in the second byte of the NAL unit header that may be used in HEVC extensions as scalable layer identifier **layer_id** in a scalable context.

In the layer scenario approaches, a scenario identifier is signaled in the NAL unit header that allows a decoder or MANE to interpret the five reserved bits as separate one- or many-dimensional identifier for video codings that feature one or multiple dimensions of scalability, such as temporal and spatial, spatial and multiview or others. A detailed description of the scalable dimension and the operation points within is given by means of the VPS, whereas the NAL unit header only provides information on how to interpret the five reserved bits of the NAL unit header syntax in order to derive one or more scalable identifiers.

The VPS syntax described next provides means to signal profile and level of each operation point and is designed to allow for straight-forward on-the-fly adaptation. The following sections give details on this syntax.

- 5 The proposed NAL unit header syntax is described first.

The second byte of the NAL unit header is used to signal a so called layer scenario (layer_scenario) and a scalability identifier (layer_id) that identifies an operation point in the context of the layer scenario.

10

Fig. 4 gives the NAL unit header syntax for HEVC with the new syntax elements layer_scenario and layer_id. The syntax element layer_scenario signals the scenario of scalability dimensions, for example, one scalability dimension for using only temporal scalability (e.g. temporal scalability as in the HEVC base specification) or combining two types of scalability (e.g. temporal and spatial). The syntax element layer_id describes the operation point of a specific NAL unit within the signaled scalability dimensions and has to be interpreted according to the layer_scenario syntax element.

15

In the case of no or one dimension of scalability, i.e. layer_scenario with a value of zero, the value of layer_id can be interpreted as an unsigned integer, listing the operation points in consecutive order according to their decoding dependencies and importance. In case of no scalability, all NAL units shall have a layer_id value of zero to signal that there is no meaningful hierarchical adaptation procedure and all NAL units are of the same importance.

20

25

With two dimensions of scalability, i.e. layer_scenario with a value of one, the value of layer_id can be interpreted as two 3-bit unsigned integer variables. The first three bits giving the operation point of the NAL unit in the first scalability dimension, e.g. temporal, and the second 3-bits give the operation point of the NAL unit in the second scalability dimension, e.g. spatial.

30

When layer_scenario has a value of two, the value of layer_id can be interpreted as a 2-bit unsigned integer variable giving the operation point in the first scalability dimension, followed by a 4-bit unsigned integer variable describing the operation point in the second scalability dimension. This scenario can be beneficial when the amount of layers of one dimension is larger than in the other dimension.

35

Given three dimensions of scalability, the corresponding value of layer_scenario is three, and the value of the layer_id syntax element can be interpreted as three 2-bit unsigned integer variables giving the operation point according to the three scalability dimensions in use, e.g. temporal, spatial and multiview scalability.

5

Table 2 gives an overview on the values of layer_scenario and presents the corresponding interpretation of layer_id.

Table 2: Overview of layer_scenario values.

10

layer_scenario	Number of scalability dimensions	interpretation of layer_id	number of possible operation points per dimension
0	0-1	one 6-bit unsigned integer	64
1	2	two 3-bit unsigned integer	8
2	2	one 2-bit and one 4-bit unsigned integer	4 (first dimension) and 16 (second dimension)
3	3	three 2-bit unsigned integer	4

The corresponding video parameter set syntax is described next.

The syntax of the Video Parameter Set has to be adjusted in order to signal profile and level for each operation point in the possibly three dimensional scalability space and provide means to adapt it in a straight-forward way to reflect the current level of adaptation of the video bitstream. This is achieved by using the vps_extension_data() syntax element for profile and level signaling of NAL units that use any kind of scalability, e.g. temporal, spatial or others. Furthermore, it provides detailed information for the interpretation of the layer_id syntax element within the NAL unit header.

20

The VPS syntax given is given in Fig. 5.

Fig. 5 signals profile and level for NAL units that do not use any kind of scalability, i.e. video data coded according to the HEVC base specification with the lowest temporal level.

25

Furthermore, the syntax element layer_id_type is given in order to allow a more detailed understanding of the scalability dimensions described with the layer_id syntax element within the NAL unit header syntax. Table 1c specifies the actual scalability dimensions signaled in layer_id for the layer_id_type syntax element. For the HEVC base

30

specification, only temporal scalability has to be defined. Future HEVC scalability extension would consequently specify further layer_id_type values for their specific needs, e.g. 1: temporal scalability, 2: spatial scalability for an SVC-like extension of HEVC.

5 This allows any device in possession of the VPS to completely interpret the operation point of a NAL unit within the known scalability dimensions, whereas devices that do not have access to the VPS can still perform correct adaptation of the video bitstream but without knowledge of the actual scalability dimensions.

10 **Table 3: layer_id_type and detailed description of scalability dimension in layer_id**

layer_id_type	Scalability dimensions
0	none
1	1: temporal scalability
1 .. 127	unspecified

Descriptions of operation points that use scalability in any dimensions (e.g. temporal level > 0, spatial scalability ...) are given by means of the vps_extension_data() syntax as given
15 in Fig. 6. Each operation point is identified via its layer_id. The new_profile_level_flag allows for either explicit signaling of profile and level of the described operation or implicit signaling by referencing another operation point description by its layer_id using the syntax element ref_layer.

20 Using the vps_extension_data() syntax allows for adaptation in a straight-forward way, as operation points are explicitly referenced and can be omitted by excluding a fixed number of bytes. Whenever the omitted vps_extension_data() syntax is not referenced by other operation point description, no bits of the VPS have to be changed apart from the omitted ones, which simplifies on-the-fly adaptation considerably.

25 Thus, using the nomenclature applied with respect to Figs. 1a to 1c, the just outlined specific embodiments describe a data stream composed as depicted in Fig. 7. The data stream which is generally indicated using reference sign 10 has a media content coded thereinto such as, as denoted above, audio, video or other media content. The data stream
30 10 comprises packets 12 called NAL units. Each packet 12 comprises a packet type identifier, namely nal_unit_type, identifying a packet type associated with a respective packet 12 out of a plurality of packet types. Each packet 12 having a packet type out of a first set of the plurality of packet types associated therewith such as all NAL units, comprises an operation point identifier 34, namely the combination of layer_scenario and

layer_id, identifying an operation point 18 associated with the respective packet 12, out of a plurality of operation points 18 within the scalability space 20 spanned by a number n of scalability axes 22a, b with $0 > n$, the operation points 18 having a sequential operation point order defined thereamong. In this regard, it should be noted that not necessarily each NAL unit or packet 12 has an operation point identifier 34. Rather, there may be packets 12 of a certain packet type, which are of general nature as far as the operation points are concerned and accordingly do not comprise any operation point identifier.

In the embodiment of Fig. 7, however, each packet 12 having a packet type out of a second set of the packet types associated therewith, namely the non-VPS NAL units, additionally carries data 36 contributing to a coded representation of the media content at the operation point 18 associated with a respective packet along with packets 12 having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet in accordance with the sequential operation point order. Reference is made to the description of Figs. 1b and Fig. 1c for details in this regard.

Each packet having a predetermined packet type disjoint to the second set of packet types associated therewith, has a scalability axes descriptor 38 defining the number n and a semantic meaning of the one or more scalability axes with the scalability axes descriptor 38 being represented by layer_id_type occurring in the VPS NAL units, i.e. NAL units 12 where the rbsp data portion 40 is filled according to the VPS syntax 42.

As also described above, each VPS packet may also comprise a sequence of profile and/or level descriptors 44, namely vps_extension_data, individually associated with at least a subset of the plurality of operation points 18 such as, for example, the leading ones in terms of the sequential operation point order. Each profile and/or level descriptor 44 defines an available coding option set, i.e. a minimum profile to be supported for reconstruction purposes and/or an available syntax element value range, i.e. a minimum level required for reconstruction, which a syntax of packets 12 having the operation point associated with the respective profile and/or level descriptor 44, associated therewith, obey. In other words, the combination of data resulting from adding the data of packets of a certain operation point 18 to data of packets associated with one or a combination of other operation points 18, is reconstructible only for decoders which support the respective profile and/or level.

A first set of the profile and/or level descriptors 44 explicitly signal the definition of the available coding option set and/or available syntax element value range, namely those for

which the syntax element `new_profile_level_flag`, i.e. flag 46, equals 1. A second, disjoint non-empty set of the profile and/or level descriptors 44 signal the definition of the available coding option set and/or available syntax element value range by referencing another preceding profile and/or level descriptor 44, namely those for which `new_profile_level_flag` equals 0. In Fig. 6 the explicit signaling of profile is done using syntax element `profile_idc` at 48, while syntax element `level_idc` represents a level indicator 50. The referencing to another profile and/or level descriptor 44 in case of `new_profile_level_flag=0`, is done using a syntax element `ref_layer_id` 52. Obviously, the use of referencing profile and/or level descriptors 44 reduces the size necessary for transmitting the sequence 54 of profile and/or level descriptors, and this aspect may, as further outlined below, be exploited independent from the co-use of the operation point identifier 34 and the scalability axes descriptor 38.

While the sequence 54 of profile and/or level descriptors 44 may be sequentially associated in the sequential operation point order with the at least subset of plurality of operation points, this is not necessarily the case. Rather, each profile and/or level descriptor may have a field 56, namely `layer_id`, which identifies the operation point 18 the respective profile and/or level descriptor 44 is associated with. The network entity which deals with the data stream 10 may ascertain the sequential operation point order defined among the operation points 18 from the order of the fields of the profile and/or level descriptors 44 in sequence 54. Alternatively, however, the sequential operation point order may be known to the network entity by default depending on, for example, the scalability axes descriptor. To be even more precise, the order of the operation points identified by the fields 56 and the sequence of the profile and/or level descriptors 44 may be used by a network entity such as network entity 28 of Fig. 2, so as to derive the sequential operation point order.

The field 56 may identify the respective operation points using a bit sequence which is common to the operation point identifiers 34. In the embodiment outlined below, for example, all operation point identifiers 34 had a fixed-bit-length syntax element, namely `layer_id`, and a scenario syntax element, namely `layer_scenario`, where the scenario syntax element signaling `n` and an association of `n` partitions of the fixed-bit-length syntax element with the `n` scalability axes, the fixed-bit-length syntax element is common to, or equally constructed to, the bit sequence of fields 56. Fig. 8 illustratively shows an example for partitioning the fixed-bit-length syntax element or bit sequence 60 of fields 56 or fixed-bit-length syntax element of the operation point identifier 34 into a different number of partitions in accordance with three different states 62 of the scenario syntax element 64 with the partitions exemplarily shown with a dashed line for a first state, a dash-dot line for a second state and dotted lines for a third state. That is, a network entity interpreting the

fixed-bit-length syntax element or bit sequence 60 may identify the correct operation point identified by the respective fixed-bit-length syntax element or bit sequence 60 by using the bits contained in the individual partitions as components in the n-dimensional scalability space 20.

5

In the embodiment described below, the bit sequence used for the syntax element `ref_layer_id` 52 uses the same bit sequence as shown in Fig. 8. Alternatively, a relative pointer defined in units of ranking positions in the sequential operation point order may have been used. For example, the network entity may merely add `ref_layer_id` bit-wise to syntax element `layer_id` 56 so as to access or obtain the `layer_id` of the operation point which the current profile and/or level descriptor 44 refers to so as to adopt the profile identifier and/or level identifier `profile_idc` and/or `level_idc` 48 and/or 50 for the current operation point.

10

15

In the embodiment described above, the scalability axes descriptor 38, namely `layer_id_type`, forms a syntax element, the possible states of which are individually associated with different combinations of values for `n` and the corresponding semantic meaning of the `n` scalability axis. That is, possible states of the scalability axes descriptor are individually mapped to different (`n`) meaning (1), ..., meaning (`n`) (wherein reference is made to table 3). That is, using the scalability axes descriptor 38, a network entity reading the data stream knows about the semantics of meaning of the property for which the values within the partitions in Fig. 8 are in ordinal measure. A network entity which, however, has not yet had an opportunity to successfully read a VCL NAL unit, may however at least rely on the ordinal characteristic of the values in the partitions of bit sequence 60 of operation point identifiers 34 as obtained using the partitioning according to the scenario syntax element 64, so as to decide on possible reductions of the data stream in processing the data stream.

20

25

30

Before proceeding to the next specifically outlined embodiment, it is noted that in accordance with the above mentioned further aspects of the present application, the scalability axes descriptor 38 could be omitted with the semantics thereof being included, for example, in the operation point identifiers 34, which would however accordingly have to consume more bits than compared to the embodiments described right now. Nevertheless, however, such alternative embodiments would still be advantageous in that the above described sequence 54 of profile and/or level descriptors 44 is present, which involves both explicitly signaling profile and/or level descriptors 44 as well as indirectly signaling/referencing profile and/or level descriptors 44.

35

With regard to the description above, however, it should be noted that various modifications could be performed onto the embodiments outlined above. For example, the number of bits chosen in the above example for the fixed-length-syntax element or bit sequence 60, namely 6, could be changed to be greater or smaller. A similar note is valid as far as the possible scalability axes number n adjustable by scenario syntax element 64 is concerned: the set of possible states was (1, 2, 3), but a different set of possible integers for n could be realized as well.

For the sake of completeness only, Fig. 9 shows the mode of operation of a network entity which for example reads the sequence of profile and/or level descriptors 44. First of all, the field of the first profile and/or level descriptor 44 in list 54 is read at step 66. Then, the network entity checks at step 68 whether the current profile and/or level descriptor 44, the field of which was read in step 66, explicitly signals profile and/or level or indirectly signals the same. In accordance with the embodiment above, step 68 involves reading syntax element reserved_zero_one_bit with the actual decision being dependent on the value of the syntax element: explicit signaling is indicated by new_profile_level_flag being equal to 1, and the indirect signaling by the syntax element being equal to 0.

In case of the explicit signaling, the network entity reads in step 70 the profile and/or level identifier 58/50 and associates same with the operation point identified by the field read in step 66. In case of indirect signaling, however, the network entity reads in step 72 from the data stream the reference field 52 and adopts in step 74 the profile and/or level identifier associated with the operation point identified by this reference field 52 as the profile and/or level identifier of the operation point identified by the field read in step 66. These steps are then repeated until the sequence of profile and/or level descriptors 44 has been scanned through.

Similarly, Fig. 10 illustrates the mode of operation of a network entity which takes advantage of those embodiments described above, where the scalability axes descriptor 38 completes the information provided by the operation point identifiers 34 in the individual packets. As shown in Fig. 10, first of all the operation point identifier of a current packet is read in step 76. This is done for all packets 12. Further, the network entity inspects the VPS NAL units so as to read therefrom in step 78 the scalability axes descriptor. Both steps 76 and 78 are performed each time a respective packet of the respective packet type is encountered by the network entity. Each time an operation point identifier is read at step 76, the network entity distinguishes in step 80 between the case where the scalability axes descriptor has not yet been read, thereby not being available, or has already been read thereby being available. In case of the availability, the network entity interprets the

operation point identifier of the current packet according to the scalability axes descriptor read in step 78 in step 82. After interpretation, the network entity knows about the meaning of the scalability axis and may act accordingly. In particular, irrespective of the availability or non-availability, the network entity selects packets out of packets 12 in step 84 such as
5 selecting those packets which are to be decoded or to be forwarded to a recipient such as a recipient 30. The step 84 may be dependent on external data such as, for example, data indicating the recipient's capability of reproducing media data, such as maximum reproducible spatial resolution, maximum pixel bit depth of the display and so forth. In case the interpretation is not available due to the non-availability of the scalability axes
10 descriptor, network entity 28 may be configured to act defensively by for example merely further processing the packets belonging to the lowest operation point in accordance with the sequential operation point order.

Again, the network entity 28 may be a decoder, a transcoder, a media aware network entity
15 such as a mobility management entity or the like.

Moreover, an entity which generates any of the embodiments for a data stream described above, such as an encoder of the like, may be configured to intersperse the VPS NAL units in between the data carrying packets sufficiently frequently so as to keep the time durations
20 low where a network entity faces the non-availability problems occurring if the scalability axes descriptor is not yet available as described with respect to Fig. 10.

For a network entity, which reads the sequence of profile and/or level identifiers 44, it is noted that such a network entity may be configured to in addition to the process of Fig. 10,
25 or alternatively thereto, render a selection among the packets 12 based on the availability coding option set and/or available syntax element value range associated with the individual operating points: for example, the network entity may exclude operation points the required profile and/or level of which exceeds the maximum profile and/or maximum level of itself in case of the network entity being a decoder, for example or of the recipient
30 with which the network entity is communicating, in case of the network entity being, for example, an MME.

In other words, what has been described with respect to Fig. 4 to 6 is a datastream into which a media content is coded, the datastream comprising packets (c.p. NAL units), each
35 packet comprising an packet type identifier `nal_unit_type` identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith (e.g. all NAL units), comprises an operation point identifier `layer_scenario` and

layer_id identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith (e.g. non-VPS NAL units), additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet, wherein each packet (c.p. VPS NAL units) having a predetermined packet type disjoint to (i.e. external to) the second set associated therewith, has an scalability axes descriptor (c.p. layer_id_type) defining the number of the one or more scalability axes and a semantic meaning thereof.

Another example of the datastream, wherein each of the packets having the predetermined packet type associated therewith, also comprises a sequence of profile and/or level descriptors vps_extension_data individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option set (c.p. profile) and/or an available syntax element value range (c.p. level) which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, (c.p. new_profile_level_flag=1) and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor (c.p. new_profile_level_flag=1).

In another example of the datastream, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points.

In another example of the datastream, each profile and/or level descriptor has a field (c.p. layer_id) identifying the operation point the respective profile and/or level descriptor is associated with (i.e. along with 4), the sequential operation point order is derivable from the sequence of profile and/or level descriptors.

In another example of the datastream, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.

5

In another example of the datastream, all operation point identifiers have a fixed-bit-length syntax element (c.p. `layer_id`) and a scenario syntax element (c.p. `layer_scenario`), the scenario syntax element signaling n and an association (but without semantic meaning) of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.

10

In another example of the datastream, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order (with the second alternative being an alternative to above description).

15

According to a further aspect of the datastream, all operation point identifiers have a fixed-bit-length syntax element and a scenario syntax element, the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes so that the n -th portion is a ordinal measure of a position of the respective operation point along the n -th scalability axis.

20

According to another aspect of the datastream, the scalability axes descriptor (c.p. `layer_id_type`) comprises a syntax element the possible states of which are individually associated with different combinations of values for n and semantic meaning of the corresponding semantic meaning of the n scalability axes (i.e. a function maps $\{n, \text{meaning}(1), \dots, \text{meaning}(n)\}$ to $\{1 \dots \text{max. possible state of } \text{layer_id_type}\}$).

25

In another datastream into which a media content is coded, the datastream comprises packets, each packet comprising an packet type identifier identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a

30

35

- sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet, wherein each packet having a predetermined packet type disjoint to (external to) the second set associated therewith, comprises a sequence of profile and/or level descriptors individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor.
- 20 In another example of the datastream, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points.
- 25 In another example of the datastream, each profile and/or level descriptor has a field identifying the operation point the respective profile and/or level descriptor is associated with.
- 30 In another example of the datastream, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.
- 35 In another example of the datastream, all operation point identifiers have a fixed-bit-length syntax element and a scenario syntax element, the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.

In another example of the datastream, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order.

In another example of the datastream, all operation point identifiers have a flag signaling as to whether same belongs to the first set of the profile and/or level descriptors explicitly signaling the definition of the available coding option set and/or available syntax element value range, and the second, disjoint non-empty set of the profile and/or level descriptors signaling the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor.

Another aspect may have a network entity (e.g., decoder or MANE) for processing a datastream into which a media content is coded, the datastream comprising packets, each packet comprising an packet type identifier identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet, wherein the network entity is configured to read from a packet having a predetermined packet type disjoint to (i.e. external to) the second set associated therewith, a scalability axes descriptor defining the number of the one or more scalability axes and a semantic meaning thereof, and interpret the operation point identifiers depending on the scalability axes descriptor.

In another example, the network entity is configured to read from the packet having the predetermined packet type disjoint to (external to) the second set associated therewith, a sequence of profile and/or level descriptors individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option

set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configured to, responsive to a signaling in the datastream, explicitly read from a first set of the profile and/or level descriptors the
5 definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another, preceding profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced
10 preceding profile and/or level descriptor.

In another example of the network entity, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of
15 operation points.

In another example of the network entity, each profile and/or level descriptor has a field identifying the operation point the respective profile and/or level descriptor is associated with.
20

In another example of the network entity, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.
25

In another example of the network entity, all operation point identifiers have a fixed-bit-length syntax element and a scenario syntax element, the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.
30

In another example of the network entity, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level
35 descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order.

In another example of the network entity, all operation point identifiers have a fixed-bit-length syntax element and a scenario syntax element, wherein the network entity is configured to use the scenario syntax element to determine n and associate n partitions of the fixed-bit-length syntax element with the n scalability axes so that the n -th portion is a
5 ordinal measure of a position of the respective operation point along the n -th scalability axis.

In another example, the network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types
10 associated therewith based on the interpretation.

Another example may have a network entity (e.g., decoder or MANE) for processing a datastream into which a media content is coded, the datastream comprising packets, each packet comprising an packet type identifier identifying a packet type associated with the
15 respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation
20 point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective
25 packet, or is equals the operation point associated with the respective packet, wherein the network entity is configured to read from a packet having a predetermined packet type disjoint to (external to) the second set associated therewith, a sequence of profile and/or level descriptors sequentially associated, in the sequential operation point order, with at least a subset (the leading ones, in terms of the sequential operation point order) of the
30 plurality of operation points, each profile and/or level descriptor defining an available coding option set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configured to, responsive to a
35 signaling in the datastream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another, preceding profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of

the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced preceding profile and/or level descriptor.

5 In another example, the network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types associated therewith based on the associated available coding option set and/or an available syntax element value range.

10 The description of the next embodiment starts with a further introduction which would be, however, also transferrable to the embodiments described so far.

Besides recording, storing and playout of video, the transmission thereof is a major application in today's communication networks. The unsteady nature of specific
15 communication channels requires means of adaptation to be provided by the high level syntax of HEVC to perform well in different environments. Straight-forward adaptation of video bitstreams within the network during transmission, e.g. by Media-Aware Network Elements (MANEs), is one of the major challenges. Such devices usually only have limited resources in terms of processing power and have to operate with very low delay which
20 prevents in-depth bitstream analysis.

The demands of this setup were already identified within the standardization of H.264/AVC. The encapsulation of video data into Network Abstraction Layer (NAL) units and the design decision to signal many important but rather invariant parameters outside
25 the video bitstream in so called Parameter Sets reflect this understanding. A lack of extensibility of the base specification of H.264/AVC led to complicated approaches to extend the H.264/AVC high level syntax in a backwards-compatible way when extensions were specified. For HEVC, a flexible and extensible high level syntax should be taken care of in the base specification. Therefore, the high level syntax has to provide devices, such as
30 MANEs, with the necessary information in an accessible form, e.g. not entropy decoded, and suitable for signaling adaptation in a straight-forward way.

The below takes on the concept of a Video Parameter Set (VPS) as defined in the current draft specification to provide information necessary for adaptation of the video bitstream,
35 such as operation point descriptions and coding dependencies. It combines the information that in H.264 and its extensions is distributed across the layer specific sequence parameters sets (SPSs) and sits on top of the SPSs in the parameter set hierarchy.

The current definition of the VPS in the HEVC draft base specification defines the VPS to hold only redundant copies of some syntax related to sub-level demands for temporal scalability that are located in the SPS in H.264 and its extensions. Hence, the VPS is not necessary for the decoding of the coded video sequence. In accordance with the embodiment below, however, redundant signaling is omitted and the syntax element is kept in the VPS only. Furthermore, in accordance with the embodiment below, however, the profile, level and sub-level demands signaling is changed in a way that its inherent hierarchy is reflected in their position within the hierarchy of parameters sets (VPS, SPS, and PPS). This means that the most basic and fundamental parameters for bitstream adaptation and capability exchange, i.e. profile and level information, are provided top of the parameter set hierarchy, namely the VPS, whereas less important sub-level demands, i.e. `sps_max_latency_increase[ i ]`, `vps_num_reorder_pics[ i ]`, `vps_max_dec_pic_buffering[ i ]` are signaled in the SPS as in former versions of the draft specification. In this way, we will make most important information available to MANEs in a straightforward way without the necessity to compile information from several SPSs. Applications that rely on lower level demands, such as sub-level demands for temporal scalability, can still find the desired information in an uncompiled form in lower steps of the parameter set hierarchy. Generally speaking, a non-application specific MANE is hardly interested in such low level capabilities information of end devices. Profile and level demands are expected to be fine-grained enough for most application and corner-case application can still retrieve this information from the lower-level parameter sets.

A VPS defined in this way allows a MANE to be fully aware of the scalability dimensions and profile and level associated with every operation point. Operation points within the scalability dimensions with redundant profile and level identifiers can be signaled by reference to decrease redundant signaling. Utilizing the VPS extension data syntax to signal individual operation point descriptions allows for straight-forward modular structure that can easily be constructed by an encoder.

A SEI message is proposed to make MANEs and receivers aware of possibly adapted video sequences, where not all the operation points described in the VPS and its extensions are present. Allowing MANEs to be aware of the actual (adapted) content of the video sequence is of importance to facilitate further adaptation, e.g. in a cascaded adaptation process with multiple MANEs.

In order to generically signal the operation point of a NAL unit within the occurring scalability dimensions on the NAL unit header level, the reserved bits in the second byte of the NAL unit header are used to signal a layer identifier to identify enhancement layers or

views from the base layer or view. The information necessary to interpret the layer identifier of a NAL unit in terms of the scalability dimensions of the coded video sequence is provided by means of the VPS. This signaling includes the type of scalability and information on how the individual bits of the layer identifier correspond to applicable scalability dimensions. This allows a MANE or any other device to correctly build a possibly multi-dimensional (temporal, quality, spatial, multiview, depth...) scalability operation point matrix from the information provided in the NAL unit header and the VPS.

The NAL unit syntax of the embodiment described now is shown in Fig. 11 where `layer_id_plus1` minus 1 indicates the layer identification. A `layer_id_plus1` equal to 1 indicates that the layer identification is equal to 0, and the NAL unit belongs to the base layer or base view. Higher values of `layer_id_plus1` indicate that the corresponding NAL unit belongs to an enhancement layer or view associated with the specified `layer_id_plus1`.

The corresponding SPS syntax is shown in Fig. 12 and the corresponding VPS syntax in Fig. 13, where: `scalability_type` indicates the scalability types in use in the coded video sequence and the dimensions signaled through `layer_id_plus1` in the NAL unit header. When `scalability_type` is equal to 0, the coded video sequence conforms to the base HEVC specification, thus `layer_id_plus1` of all NAL units is equal to 1 and there are no NAL units belonging to an enhancement layer or view. Higher values of `scalability_type` have to be interpreted according to Table 4.

`profile_space`, `profile_idc`, `constraint_flags`, and `profile_compatibility_flag` signal information as specified in the current WD and the indicated values are valid for all NAL units with a `layer_id_plus1` equal to 1.

`level_idc[i]` signals information as specified in the current WD and the indicated values are valid for all NAL units with a `layer_id_plus1` equal to 1 and a `temporal_id` equal to `i`.

`layer_id_plus1_dim_len [ dim ]` indicates the length in bits of `layer_id_plus1` that is used to signal the `n`-th scalability dimension according to Fig. 13 beginning at the MSB for `n = 0` or the end of the `(n-1)`-th signaled scalability dimension for `n > 0`.

Table 4: Proposed scalability_type mapping table

scalability_type	max_dim	dimensions
0	1	none

1	2	spatial and quality scalability
2	2	multiview and depth scalability
3	4	multiview, spatial, quality and depth scalability
4...15	reserved	reserved

The scalability types exemplarily defined in Table 4 represent a set of scalabilities which can be anticipated at this point in time, but may be extended with new dimensions and/or combinations once the work on HEVC extensions progresses.

5

The corresponding VPS extension data syntax is shown in Fig. 14, where:

vps_extension_type indicates the type of the VPS extension. When the value of vps_extension_type is equal 0, the syntax given above applies.

10

layer_id_plus1 indicates the layer_id for which the syntax elements signaled in the VPS extension are valid.

15

max_temporal_layers_minus1 plus 1 specifies the maximum number of temporal sub-layers that may be present in NAL units with a layer_id_plus1 equal to the layer_id_plus1 signaled in the VPS.

20

new_profile_level_flag equal to 0 indicates that the profile and level signaling associated with a value of layer_id_plus1 equal to the values specified by profile_reference_layer_id_plus1 is valid for the NAL unit with layer_id_plus1 specific within the VPS extension. new_profile_level_flag equal to 1 indicates that profile and level for the specified value of layer_id_plus1 is signaled explicitly in the VPS extension.

25

profile_reference_layer_id_plus1 indicates the value of layer_id_plus1 whose associated profile and level signaling is valid for NAL units with a value of layer_id_plus1 equal to the value of layer_id_plus1 signaled in the VPS extension.

30

profile_space, profile_idc, constraint_flags, and profile_compatibility_flag signal information as specified in the current WD and the indicated values are valid for all NAL units with a layer_id_plus1 equal to the layer_id_plus1 values specified in the VPS extension data.

level_idc[i] signals information as specified in the current WD and the indicated values are values for all NAL units with a layer_id_plus1 equal to layer_id_plus1 values specified in the VPS extension data and a temporal_id equal to i.

- 5 num_reference_layers specifies the number of layers, identified by layer_id, to be used as references in the decoding process.

direct_coding_dependency_layer_id_plus1 indicates the layer_id_plus1 of NAL units to be used as references in the decoding process.

10

The embodiment just described is very similar to the one having been described with respect to Figs. 4 to 6. Insofar, the same reference signs as before have been used in order to identify corresponding syntax elements in Figs. 11 to 14. Insofar, the description of Fig. 7 is applicable to the new embodiment as well, wherein however differences are highlighted below.

15

In particular, in accordance with the embodiments of Figs. 11 to 14 the operation point identifier 34 has been fixedly split-up into two components, namely a first component identifying the rank in a temporal scalability axis in the form of syntax element temporal_id 90, and a second component treating the rank or ranks according to the further scalability axis, with the second component being denoted 92. Insofar, the embodiment having been described with respect to Figs. 11 to 14 corresponds to the previous embodiment except that it has the second component 92, which is further specified by the scalability axes descriptor.

25

A next difference pertains to the knowledge of the number of scalability axes. In the embodiment of Figs. 4 to 6, an intermediate information in the form of the scenario syntax element 60 was provided in order to provide network entities with the opportunity to derive, solely based on the individual packet itself, where the operation point of this packet is, in an ordinary sense, positioned within the scalability space. In the present embodiment, a network entity is merely able to derive the following facts in case of the non-availability of a VPS NAL unit 42: 1) the position of the operation point of the current packet along the temporal axis using temporal_id 90. 2) If layer_id_plus1 is equal to 1, the operation point is at the lower most rank with respect to all other scalability axes except the temporal scalability axis.

30

35

As described hereinafter, all differences now highlighted may be applied individually and accordingly, if the syntax element temporal_id 90 is not used, a network entity may derive

from the fact that `layer_id_plus1` equals 1, that the operation point of the current packet denotes the base layer in accordance with all scalability axes.

5 In this regard, it should be noted that choosing `layer_id_plus1` equaling 1 as denoting the lower most rank with respect to the other scalability axes except the temporal one, has merely been chosen as an example and may be varied, such as for example to `layer_id_plus1` equaling 0.

10 A further difference to the above-outlined embodiment of Figs. 4 to 6 relates to the scalability axes descriptor 38. One scalar valued syntax element 94, namely `scalability_type`, identifies the number of scalability axes (above the temporal one) and the corresponding meanings of these (additional) scalability axes, while for each of this indicated number of (additional) scalability axes, a syntax portion 96 reveals the partitioning of the fixed-length-syntax element 60 underlying component 92, field 56 and
15 reference field 52 by sequentially assigning bits of this fixed-length-syntax element 60 the individual (additional) scalability axes.

A further difference relates to the VPS packets and the fact that same additionally comprise operation point identifier reference lists 98. For each field 56, such an operation point
20 identifier reference list 98 identifies those operation point identifiers or those values of `layer_id_plus1`, which form the necessary basis for a reconstruction of the operation point which field 56 relates to. In other words, the operation point identifier reference list 98 identifies all operation points contained in a combination 24 (see Fig. 1b or 3) resulting from adding the data of packets of the operation point identified by field 56. It should be
25 noted that for a certain field 56 the operation point identifier reference list 98 does not necessarily have to list all operation points included in the respective combination 24 of the operation point identified by field 56. Rather, it would suffice if the operation point identifier reference list lists one or more of reference operation points, the corresponding combinations 24 of which together result in the combination 24 of the operation point
30 identified by the respective field 56.

In other words, in this aspect which is advantageous for its efficient coding of the level and/or profile constraints imposed onto the scalability layers, a data stream into which a media content is coded, comprises packets (cp. NAL units), each packet comprising an
35 packet type identifier `nal_unit_type` identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith (e.g. all NAL units), comprises an operation point identifier `layer_id_plus1` identifying an operation point associated with

the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith (e.g. non-VPS NAL units), additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet, wherein each packet (c.p. VPS NAL units) having a predetermined packet type disjoint to (i.e. external to) the second set associated therewith, has an scalability axes descriptor (c.p. scalability_type and layer_id_plus1_dim_len) defining the number of the one or more scalability axes and a semantic meaning thereof.

Another example of the data stream, wherein each of the packets having the predetermined packet type associated therewith, also comprises a sequence of profile and/or level descriptors vps_extension_data individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option set (c.p. profile) and/or an available syntax element value range (c.p. level) which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, (c.p. new_profile_level_flag=1) and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor (c.p. new_profile_level_flag=0).

In another example of the data stream, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points.

In another example of the data stream, each profile and/or level descriptor has a field (c.p. layer_id_plus1) identifying the operation point the respective profile and/or level descriptor is associated with (i.e. along with 4), the sequential operation point order is derivable from the sequence of profile and/or level descriptors.

In another example of the data stream, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.

5

In another example of the data stream, all operation point identifiers have a fixed-bit-length syntax element (c.p. `layer_id_plus1`), and the scalability access descriptor signals `n_max_dim` and an association of `n` partitions `layer_id_plus1_dim_len` of the fixed-bit-length syntax element with the `n` scalability axes, wherein the bitsequence is common to the

10

In another example of the datastream, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order (with the second alternative being an alternative to above description).

15

According to a further aspect of the datastream, all operation point identifiers have a fixed-bit-length syntax element, and the scalability axes descriptor signals `n` and an association of `n` partitions of the fixed-bit-length syntax element with the `n` scalability axes so that the `n`-th portion is a ordinal measure of a position of the respective operation point along the `n`-th scalability axis.

20

According to another aspect, of the datastream, the scalability axes descriptor (c.p. `scalability_type`) comprises a syntax element the possible states of which are individually associated with different combinations of values for `n` and semantic meaning of the corresponding semantic meaning of the `n` scalability axes (i.e. a function maps $\{n, \text{meaning}(1), \dots, \text{meaning}(n)\}$ to $\{1 \dots \text{max. possible state of scalability_type}\}$).

25

In another datastream into which a media content is coded, the datastream comprises packets, each packet comprising an packet type identifier identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number `n` of scalability axes, with $0 < n$, the operation points having a

30

35

sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet, wherein each packet having a predetermined packet type disjoint to (external to) the second set associated therewith, comprises a sequence of profile and/or level descriptors individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option set and/or an available syntax element value range which a sequence of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor.

In another example of the datastream, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points.

In another example of the datastream, each profile and/or level descriptor has a field identifying the operation point the respective profile and/or level descriptor is associated with.

In another example of the datastream, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.

In another example of the datastream, all operation point identifiers have a fixed-bit-length syntax element and the scalability access descriptor signals $n_{\max_dim}$ and an association of n partitions $layer_id_plus1_dim_len$) of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.

In another example of the datastream, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order.

In another example of the datastream, all operation point identifiers have a flag signaling as to whether same belongs to the first set of the profile and/or level descriptors explicitly signaling the definition of the available coding option set and/or available syntax element value range, and the second, disjoint non-empty set of the profile and/or level descriptors signaling the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor.

Another aspect may have a network entity (e.g., decoder or MANE) for processing a datastream into which a media content is coded, the datastream comprising packets, each packet comprising an packet type identifier identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong, wherein each packet having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet, wherein the network entity is configured to read from a packet having a predetermined packet type disjoint to (i.e. external to) the second set associated therewith, a scalability axes descriptor defining the number of the one or more scalability axes and a semantic meaning thereof, and interpret the operation point identifiers depending on the scalability axes descriptor.

In another example, the network entity is configured to read from the packet having the predetermined packet type disjoint to (external to) the second set associated therewith, a sequence of profile and/or level descriptors individually associated with at least a subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available coding option

set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configured to, responsive to a signaling in the datastream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another, preceding profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced preceding profile and/or level descriptor.

In another example of the network entity, the sequence of profile and/or level descriptors is sequentially associated, in the sequential operation point order, with the at least subset (e.g. the leading ones, in terms of the sequential operation point order) of the plurality of operation points.

In another example of the network entity, each profile and/or level descriptor has a field identifying the operation point the respective profile and/or level descriptor is associated with.

In another example of the network entity, in each profile and/or level descriptor, the field identifies the operation point the respective profile and/or level descriptor is associated with, using a bitsequence common to the operation point identifiers identifying the operation point the respective profile and/or level descriptor is associated with.

In another example of the network entity, all operation point identifiers have a fixed-bit-length syntax element and the scalability access descriptor signals `n_max_dim` and an association of `n` partitions `layer_id_plus1_dim_len` of the fixed-bit-length syntax element with the `n` scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.

In another example of the network entity, the second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor using an explicit signaling of the field of the referenced profile and/or level descriptor, or using a relative pointer defined in units of ranking positions in the sequential operation point order.

In another example of the network entity, all operation point identifiers have a fixed-bit-length syntax element and the scalability access descriptor signals n max_dim and an association of n partitions layer_id_plus1_dim_len of the fixed-bit-length syntax element with the n scalability axes so that the n -th portion is a ordinal measure of a position of the
5 respective operation point along the n -th scalability axis.

In another example, network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types associated therewith based on the interpretation.

10

Another example may have a network entity (e.g., decoder or MANE) for processing a datastream into which a media content is coded, the datastream comprising packets, each packet comprising an packet type identifier identifying a packet type associated with the respective packet, out of a plurality of packet types, wherein each packet having a packet
15 type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier identifying an operation point associated with the respective packet, out of a plurality of operation points within a scalability space spanned by a number n of scalability axes, with $0 < n$, the operation points having a sequential operation point order defined thereamong, wherein each packet having any packet type out of a
20 second set of the first set of packet types associated therewith, additionally carries data contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet, wherein the
25 network entity is configured to read from a packet having a predetermined packet type disjoint to (external to) the second set associated therewith, a sequence of profile and/or level descriptors sequentially associated, in the sequential operation point order, with at least a subset (the leading ones, in terms of the sequential operation point order) of the plurality of operation points, each profile and/or level descriptor defining an available
30 coding option set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configure to, responsive to a signaling in the datastream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element
35 value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another, preceding profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of

the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced preceding profile and/or level descriptor.

5 In another example, the network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types associated therewith based on the associated available coding option set and/or an available syntax element value range.

10 The following deviations from above embodiments would be feasible. For example, instead of having one profile/level descriptor for exactly one operation point or layer ID, an grouping of the operation points or layer IDs into layer sets could be signaled in the data stream such as in the VPS, with associating each profile/level descriptor to a respective one of the sets. For example, a field of flags could be signaled to signal the grouping: for each set, one flag for each operation point could be present for signaling as to whether the
15 respective operation point belongs to the respective set or not. Thus, rather than having field 56 citing any layer ID, the profile/level descriptor's field 56 could index a set out of the signaled sets of layer IDs, i.e. out of the sets of operation points. Accordingly, the referencing of another profile/level descriptor would be done via signaling another layer set in field 52 rather than signaling an individual layer ID therein.

20 Further, with regard to the scalability axes descriptor 38, same could be signaled differently: a scalability type field composed of a sequence of flags, each associated with a certain scalability type, could be present in the data stream in the VPS, for example, and signal as to whether the respective scalability type is present or not. In the order the flags
25 within this field, the flags being set would thus signal the number of scalability axes as well as the semantic meaning thereof. The subdivision of the layer ID field's 34 bit sequence into the fragments 62 could be done using a length indicator per scalability type, i.e. per flag set in the scalability type field, just as done by layer_id_plus1_dim_len. Together, scalability type field and layer_id_plus1_dim_len would form the scalability
30 axes descriptor 38.

Above, a pair of a list 98 of reference operation point identifiers and a field 56 identifying the operation point (or set of operation points) the list 98 of reference operation point identifiers is associated with, was used to indicate the sequential operation point order, i.e.
35 the dependency among the layers. This was one possibility to allow for operation points to depend on more than one predecessor of the operation points along their order of operation point identifiers. In a use case with multiple camera views, for example, signalling of multiple reference views (i.e. multiple values of nuh_layer_id) per coded view (i.e. an

individual value of nuh_layer_id) is beneficial as occlusions and revelation of corresponding image areas in neighboring camera views may occur depending on the translation of cameras with respect to each other. Therefore, referencing multiple camera views can lead to a higher coding efficiency as revealed image areas (i.e. texture) may be predicted instead of intra-coded from either of the referenced camera views. Coding benefits are not limited to references within the same scalability dimension, e.g. a depth view (related to the currently coded camera view or another camera view) could be utilized to predict motion vectors for the currently coded dependent camera view. However, signaling which allows for one operation point to be dependent on more than one operation point having lower operation point identifier associated therewith, could also be implemented differently. For example, for each operation point, flags could signal as to which of the operation points preceding in layer ID, i.e. having a lower operation point identifier, the respective operation point depends on. The field could be signaled in a condensed form: the N operation points occurring in the bit stream have an order defined thereamong which is defined by their operation point identifier. The sequence of operation points may, thus, scanned from the second but lowest one in operation point identifier sense to the highest one, with inserting one flag per operation point with lower operation point identifier into the VPS, thereby needing $N(N-1)/2$ bits. That is, for each of the plurality of operation points – or at least a subset thereof –, a syntax, such as the reference layer ID list 98 or the sequence of flags concerning the operation point predecessors - identifying other operation points of the plurality of operation points, identified by lower operation point identifiers, may be present in the VPS, with these identified operation point identifiers forming the basis of the respective operation point: In other words, the syntax identifies those operation points, the respective operation point the syntax belongs to, is an extension of, thereby indicating the sequential operation point order.

Although some aspects have been described in the context of an apparatus, it is clear that these aspects also represent a description of the corresponding method, where a block or device corresponds to a method step or a feature of a method step. Analogously, aspects described in the context of a method step also represent a description of a corresponding block or item or feature of a corresponding apparatus. Some or all of the method steps may be executed by (or using) a hardware apparatus, like for example, a microprocessor, a programmable computer or an electronic circuit. In some embodiments, some one or more of the most important method steps may be executed by such an apparatus.

The inventive encoded data stream can be stored on a digital storage medium or can be transmitted on a transmission medium such as a wireless transmission medium or a wired transmission medium such as the Internet.

Depending on certain implementation requirements, embodiments of the invention can be implemented in hardware or in software. The implementation can be performed using a digital storage medium, for example a floppy disk, a DVD, a Blu-Ray, a CD, a ROM, a
5 PROM, an EPROM, an EEPROM or a FLASH memory, having electronically readable control signals stored thereon, which cooperate (or are capable of cooperating) with a programmable computer system such that the respective method is performed. Therefore, the digital storage medium may be computer readable.

10 Some embodiments according to the invention comprise a data carrier having electronically readable control signals, which are capable of cooperating with a programmable computer system, such that one of the methods described herein is performed.

15 Generally, embodiments of the present invention can be implemented as a computer program product with a program code, the program code being operative for performing one of the methods when the computer program product runs on a computer. The program code may for example be stored on a machine readable carrier.

20 Other embodiments comprise the computer program for performing one of the methods described herein, stored on a machine readable carrier.

In other words, an embodiment of the inventive method is, therefore, a computer program having a program code for performing one of the methods described herein, when the
25 computer program runs on a computer.

A further embodiment of the inventive methods is, therefore, a data carrier (or a digital storage medium, or a computer-readable medium) comprising, recorded thereon, the computer program for performing one of the methods described herein. The data carrier,
30 the digital storage medium or the recorded medium are typically tangible and/or non-transitionary.

A further embodiment of the inventive method is, therefore, a data stream or a sequence of signals representing the computer program for performing one of the methods described
35 herein. The data stream or the sequence of signals may for example be configured to be transferred via a data communication connection, for example via the Internet.

A further embodiment comprises a processing means, for example a computer, or a programmable logic device, configured to or adapted to perform one of the methods described herein.

- 5 A further embodiment comprises a computer having installed thereon the computer program for performing one of the methods described herein.

10 A further embodiment according to the invention comprises an apparatus or a system configured to transfer (for example, electronically or optically) a computer program for performing one of the methods described herein to a receiver. The receiver may, for example, be a computer, a mobile device, a memory device or the like. The apparatus or system may, for example, comprise a file server for transferring the computer program to the receiver .

- 15 In some embodiments, a programmable logic device (for example a field programmable gate array) may be used to perform some or all of the functionalities of the methods described herein. In some embodiments, a field programmable gate array may cooperate with a microprocessor in order to perform one of the methods described herein. Generally, the methods are preferably performed by any hardware apparatus.

20

The above described embodiments are merely illustrative for the principles of the present invention. It is understood that modifications and variations of the arrangements and the details described herein will be apparent to others skilled in the art. It is the intent, therefore, to be limited only by the scope of the impending patent claims and not by the
25 specific details presented by way of description and explanation of the embodiments herein.

References

- 5 [1] Thomas Wiegand, Gary J. Sullivan, Gisle Bjontegaard, Ajay Luthra, "Overview of the H.264/AVC Video Coding Standard", IEEE Trans. Circuits Syst. Video Technol., vol. 13, N7, July 2003.
- [2] JCT-VC, "High-Efficiency Video Coding (HEVC) text specification Working Draft 6", JCTVC-H1003, February 2012.
- 10 [3] ISO/IEC 13818-1: MPEG-2 Systems specification.

Claims

1. A data stream into which a media content is coded, the data stream comprising packets (12), each packet (12) comprising an packet type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,
- wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,
- wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet,
- wherein each packet (44) having a predetermined packet type disjoint to the second set associated therewith, has an scalability axes descriptor (38) defining the number n and a semantic meaning of the one or more scalability axes.
2. The data stream according to claim 1, wherein each of the packets having the predetermined packet type associated therewith, also comprises a sequence (54) of profile and/or level descriptors (44) associated with one or more of at least a subset of the plurality of operation points, each profile and/or level descriptor (44) defining an available coding option set and/or an available syntax element value range which a syntax of packets having an operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another profile and/or level descriptor.

3. The data stream according to claim 2, wherein each profile and/or level descriptor (44) has a field (56) identifying the operation point (18) the respective profile and/or level descriptor (44) is associated with.
- 5 4. The data stream according to claim 3, wherein the second, disjoint non-empty set of the profile and/or level descriptors (44) signal the definition of the available coding option set and/or available syntax element value range by referencing another profile and/or level descriptor using an explicit signaling (52) of the field of the referenced profile and/or level descriptor.
- 10 5. The data stream according to any of claims 1 to 4, wherein each of the packets having the predetermined packet type associated therewith also comprises, for each of at least a subset of the plurality of operation points, a syntax (98) identifying other operation points of the plurality of operation points, identified by lower operation point identifiers, which the respective operation point is an extension of, thereby indicating the sequential operation point order.
- 15 6. The data stream according to any of claims 3 to 5, wherein each field (56) identifies the respective operation point using a bitsequence (60) common to the operation point identifiers identifying the respective operation point.
- 20 7. The data stream according to claim 6, wherein all operation point identifiers have a fixed-bit-length syntax element (60), and the scalability axes descriptor (38) signals n and an association of n partitions of the fixed-bit-length syntax element (60) with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.
- 25 8. The data stream according to claim 6, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60), and a scenario syntax element (64), the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.
- 30 9. The data stream according to any of claims 1 to 8, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60), and the scalability axes descriptor (38) signals n and an association of n partitions of the fixed-bit-length
- 35

syntax element with the n scalability axes so that the n-th portion is a ordinal measure of a position of the respective operation point along the n-th scalability axis.

5 10. The data stream according to any of claims 1 to 8, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60) and a scenario syntax element (64), the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes so that the n-th portion is a ordinal measure of a position of the respective operation point
10 along the n-th scalability axis.

11. The data stream according to any of claims 1 to 10, wherein the scalability axes descriptor (38) comprises a syntax element (94) the possible states of which are individually associated with different combinations of values for n and semantic
15 meaning of the corresponding semantic meaning of the n scalability axes.

12. A data stream into which a media content is coded, the data stream comprising packets (12), each packet comprising a packet type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,
20

wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,
25

wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the
30 respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet,

35 wherein each packet (12) having a predetermined packet type disjoint to the second set associated therewith, comprises a sequence (54) of profile and/or level descriptors (44) each associated with one or more of at least a subset of the plurality

of operation points, each profile and/or level descriptor (44) defining an available coding option set and/or an available syntax element value range which a sequence of packets having an operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein a first set of the profile and/or level descriptors explicitly signal the definition of the available coding option set and/or available syntax element value range, and a second, disjoint non-empty set of the profile and/or level descriptors signal the definition of the available coding option set and/or available syntax element value range by referencing another profile and/or level descriptor.

13. The data stream according to claim 12, wherein each profile and/or level descriptor (44) has a field identifying the operation point the respective profile and/or level descriptor is associated with.

14. The data stream according to claim 13, wherein the second, disjoint non-empty set of the profile and/or level descriptors (44) signal the definition of the available coding option set and/or available syntax element value range by referencing another profile and/or level descriptor using an explicit signaling (52) of the field of the referenced profile and/or level descriptor.

15. The data stream according to any of claims 12 to 14, wherein each of the packets having the predetermined packet type associated therewith also comprises, for each of at least a subset of the plurality of operation points, , a syntax (98) identifying other operation points of the plurality of operation points, identified by lower operation point identifiers, which the respective operation point is an extension of, thereby indicating the sequential operation point order.

16. The data stream according to any of claims 12 to 15, wherein each field (56) identifies the respective operation point using a bitsequence (60) common to the operation point identifiers identifying the respective operation point.

17. The datastream according to claim 16, wherein all operation point identifiers have a fixed-bit-length syntax element (60) and the scalability axes descriptor (38) signals n and an association of n partitions of the fixed-bit-length syntax element (60) with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.

18. The data stream according to claim 16, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60) and a scenario syntax element (64), the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.
19. The data stream according to any of claims 12 to 18, wherein all operation point identifiers (34) have a flag signaling as to whether same belongs to the first set of the profile and/or level descriptors explicitly signaling the definition of the available coding option set and/or available syntax element value range, and the second, disjoint non-empty set of the profile and/or level descriptors signaling the definition of the available coding option set and/or available syntax element value range by referencing another, preceding profile and/or level descriptor.
20. A network entity for processing a data stream into which a media content is coded, the data stream comprising packets (12), each packet (12) comprising an packet type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,
- wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,
- wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet,
- wherein the network entity is configured to
- read from a packet (12) having a predetermined packet type disjoint to the second set associated therewith, a scalability axes descriptor (38) defining the number n and a semantic meaning of the one or more scalability axes, and

interpret the operation point identifiers depending on the scalability axes descriptor.

21. The network entity according to claim 20, wherein the network entity is configured to read from the packet having the predetermined packet type disjoint to the second set associated therewith, a sequence (54) of profile and/or level descriptors (44) individually associated with at least a subset of the plurality of operation points, each profile and/or level descriptor (44) defining an available coding option set and/or an available syntax element value range which a syntax of packets having the operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configured to, responsive to a signaling in the data stream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another, profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced profile and/or level descriptor.
22. The network entity according to claim 21, wherein each profile and/or level descriptor (44) has a field (56) identifying the operation point (18) the respective profile and/or level descriptor (44) is associated with.
23. The network entity according to claim 22, wherein the second, disjoint non-empty set of the profile and/or level descriptors (44) signal the definition of the available coding option set and/or available syntax element value range by referencing another profile and/or level descriptor using an explicit signaling (52) of the field of the referenced profile and/or level descriptor.
24. The network entity according to any of claims 21 to 23, wherein the network entity is configured to read, from the packets having the predetermined packet type associated therewith, also, for each of at least a subset of the plurality of operation points, a syntax (98) identifying other operation points of the plurality of operation points, identified by lower operation point identifiers, which the respective operation point is an extension of, thereby indicating the sequential operation point order.

25. The network entity according to claim 24, wherein each field (54) identifies the respective operation point using a bitsequence (60) common to the operation point identifiers identifying the respective operation point.
- 5 26. The network entity according to claim 25, wherein all operation point identifiers have a fixed-bit-length syntax element (60) and the scalability axes descriptor (38) signals n and an association of n partitions of the fixed-bit-length syntax element (60) with the n scalability axes, wherein the bitsequence is common to the fixed-bit-length syntax element.
- 10 27. The network entity according to claim 25), wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60) and a scenario syntax element (64), the scenario syntax element signaling n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes, wherein the bitsequence
- 15 is common to the fixed-bit-length syntax element.
28. The network entity according to any of claims 20 to 27, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60) and the scalability axes descriptor (38) signals n and an association of n partitions of the fixed-bit-length syntax element with the n scalability axes so that the n -th portion is a ordinal measure of a position of the respective operation point along the n -th scalability axis.
- 20 29. The network entity according to any of claims 20 to 27, wherein all operation point identifiers (34) have a fixed-bit-length syntax element (60) and a scenario syntax element (64), wherein the network entity is configure to use the scenario syntax element to determine n and associate n partitions of the fixed-bit-length syntax element with the n scalability axes so that the n -th portion is a ordinal measure of a position of the respective operation point along the n -th scalability axis.
- 25 30. The network entity according to any of claims 20 to 29, wherein the network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types associated therewith based on the interpretation.
- 30 31. A network entity for processing a data stream into which a media content is coded, the data stream comprising packets (12), each packet (12) comprising an packet
- 35

type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,

wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,

wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet,

wherein the network entity is configured to

read from a packet having a predetermined packet type disjoint to the second set associated therewith, a sequence of profile and/or level descriptors each associated with one or more of at least a subset of the plurality of operation points, each profile and/or level descriptor defining an available coding option set and/or an available syntax element value range which a syntax of packets having an operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configure to, responsive to a signaling in the data stream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced profile and/or level descriptor.

32. The network entity according to claim 31, wherein the network entity is configured to render a selection among the packets having any packet type out of the first set of the plurality of packet types associated therewith based on the associated available coding option set and/or an available syntax element value range.

33. A method for processing a data stream into which a media content is coded, the data stream comprising packets (12), each packet (12) comprising an packet type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,

wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,

wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or equals the operation point associated with the respective packet,

wherein the method comprises:

reading from a packet (12) having a predetermined packet type disjoint to the second set associated therewith, a scalability axes descriptor (38) defining the number n and a semantic meaning of the one or more scalability axes, and

interpreting the operation point identifiers depending on the scalability axes descriptor.

34. A method for processing a data stream into which a media content is coded, the data stream comprising packets (12), each packet (12) comprising an packet type identifier (16) identifying a packet type associated with the respective packet, out of a plurality of packet types,

wherein each packet (12) having a packet type out of a first set of the plurality of packet types associated therewith, comprises an operation point identifier (34) identifying an operation point (18) associated with the respective packet, out of a plurality of operation points within a scalability space (20) spanned by a number n

of scalability axes (22a, 22b), with $0 < n$, the operation points having a sequential operation point order defined thereamong,

wherein each packet (12) having any packet type out of a second set of the first set of packet types associated therewith, additionally carries data (36) contributing to a coded representation of the media content at the operation point associated with the respective packet along with the packets having any operation point associated therewith which precedes the operation point associated with the respective packet, or is equals the operation point associated with the respective packet,

wherein the method comprises:

reading from a packet having a predetermined packet type disjoint to the second set associated therewith, a sequence of profile and/or level descriptors each associated with one or more of at least a subset of the plurality of operation points, each profile and/or level descriptor defining an available coding option set and/or an available syntax element value range which a syntax of packets having an operation point associated with the respective profile and/or level descriptor, associated therewith, obey, wherein the network entity is configure to, responsive to a signaling in the data stream, explicitly read from a first set of the profile and/or level descriptors the definition of the available coding option set and/or available syntax element value range, and read, for a second, disjoint non-empty set of the profile and/or level descriptors, a reference to another profile and/or level descriptor and derive the definition of the available coding option set and/or available syntax element value range of the second, disjoint non-empty set of the profile and/or level descriptors from the respective referenced profile and/or level descriptor.

35. A computer program for performing a method for processing as claimed in claims 33 and 34, when the computer program runs on a computer or processor.

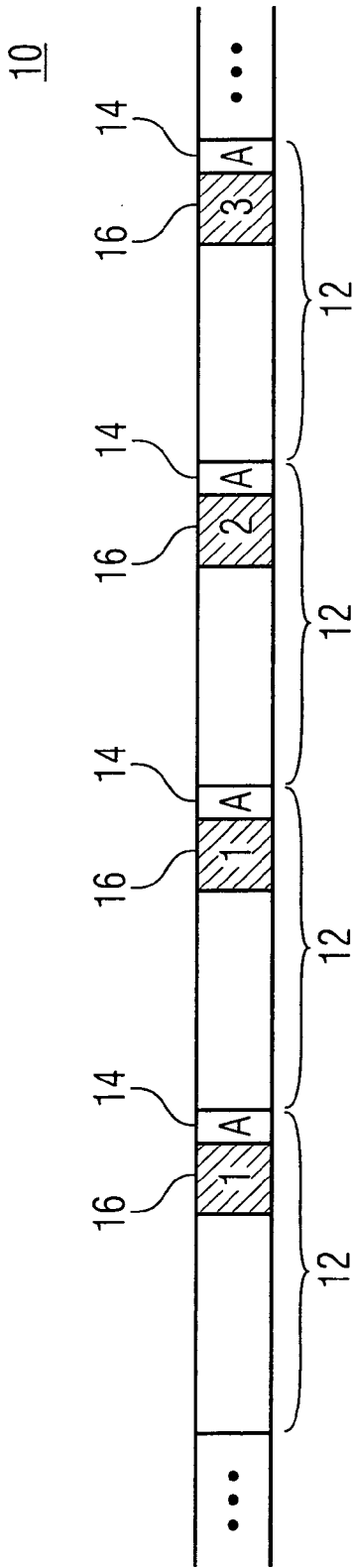


FIG 1A

2/17

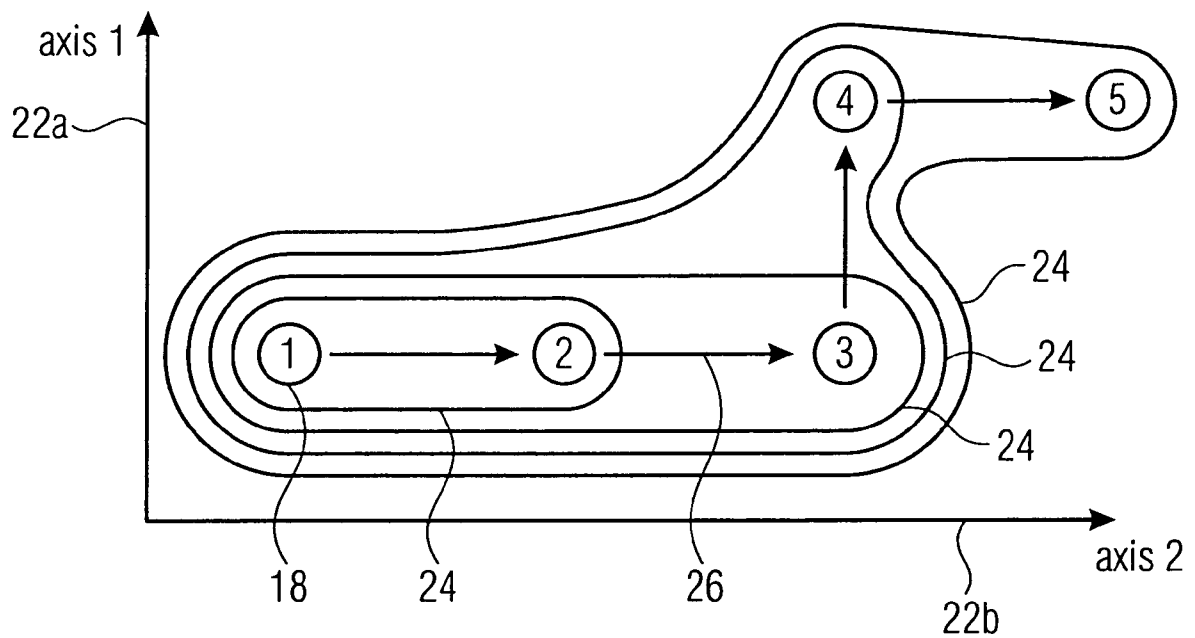
20

FIG 1B

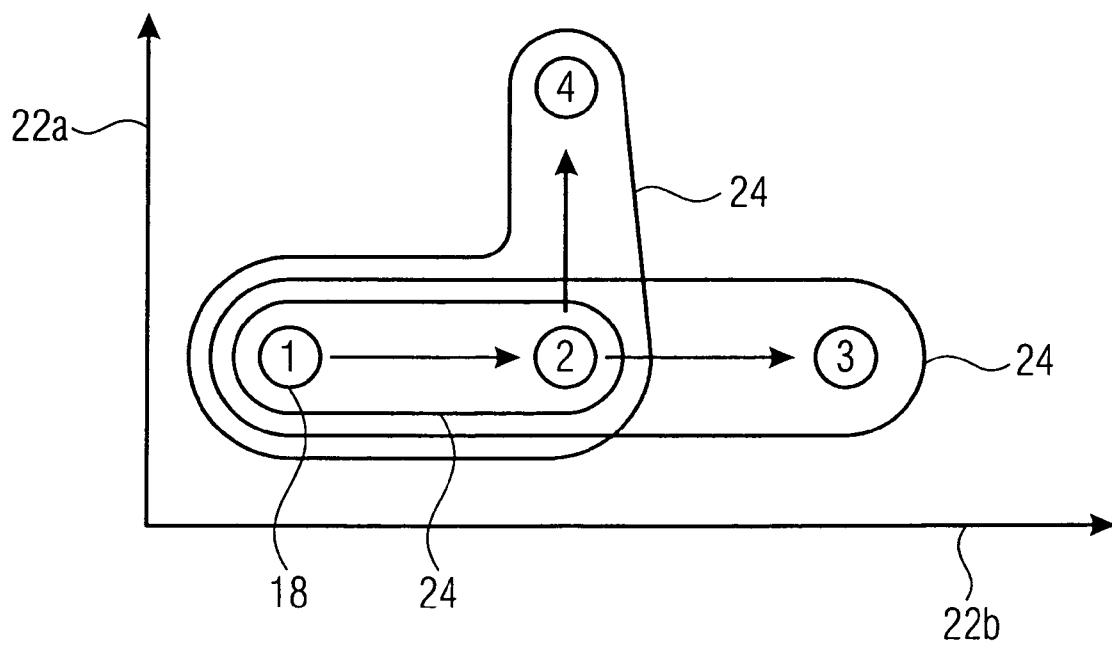


FIG 1C

3/17

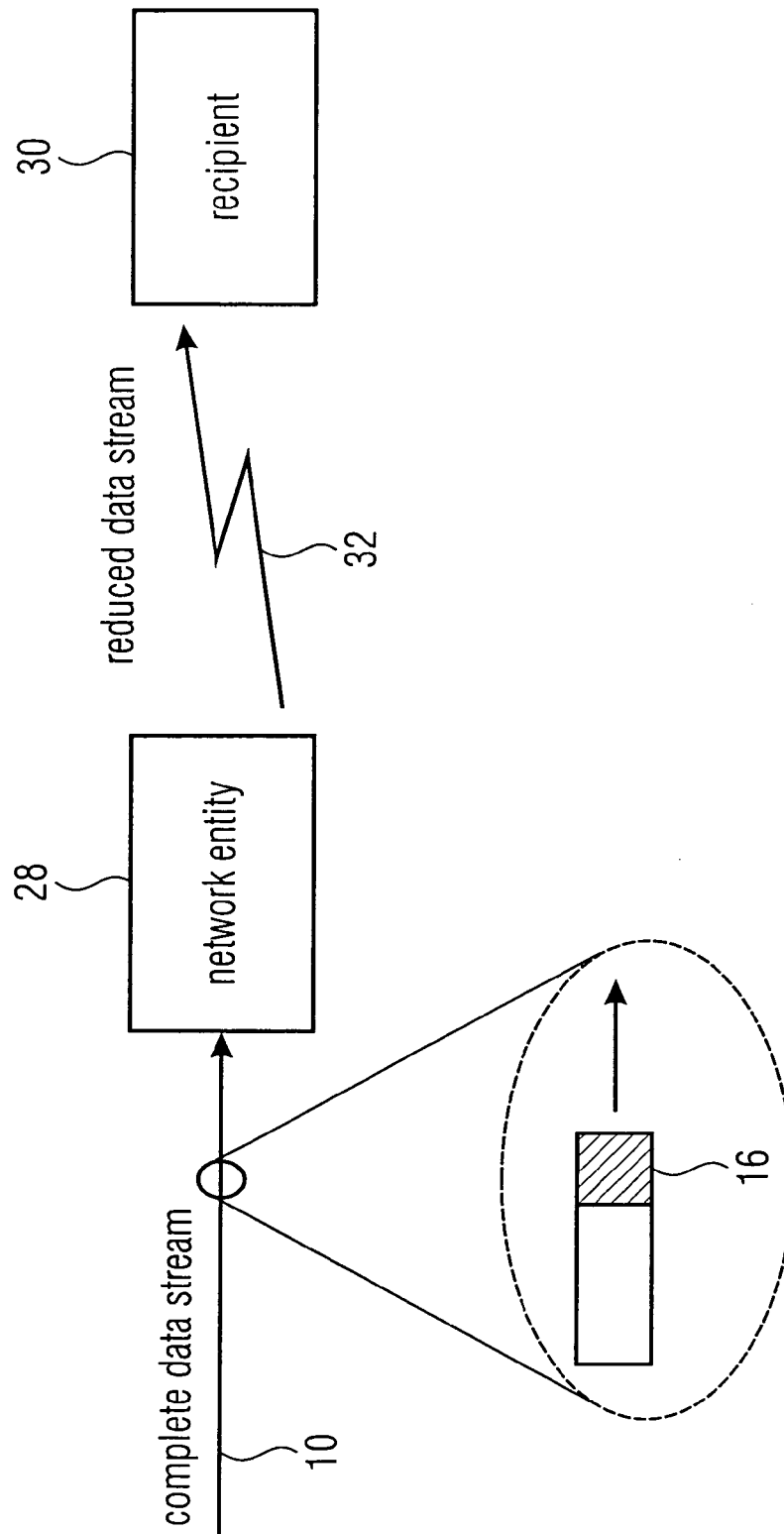


FIG 2

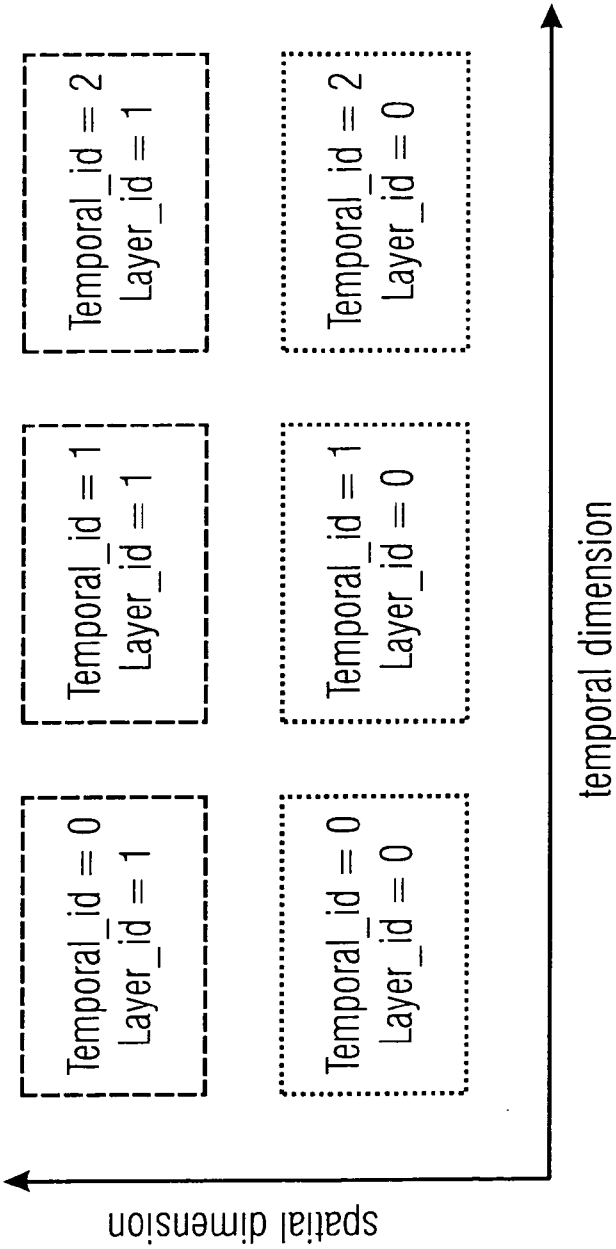


FIG 3A

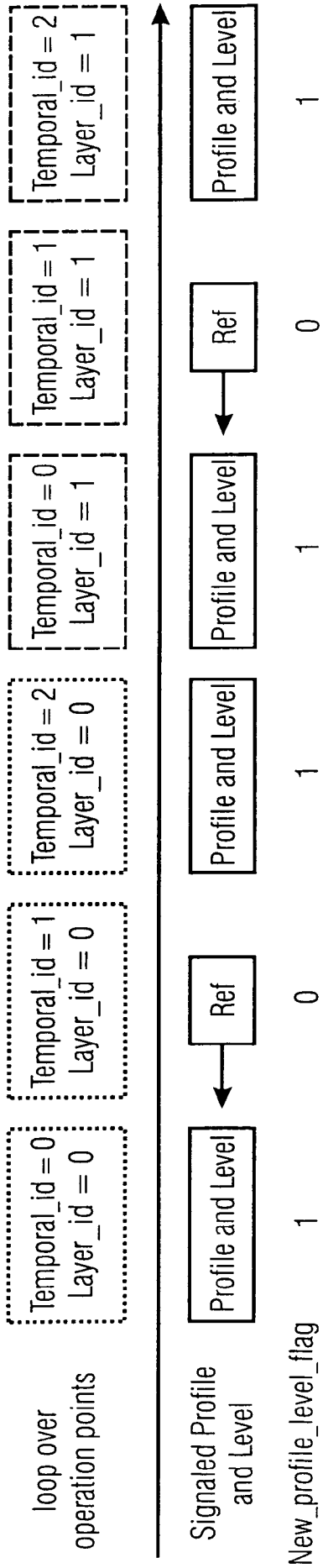


FIG 3B

12

nal_unit(NumBytesInNALunit) {		Descriptor
forbidden_zero_bit		f(1)
nal_ref_flag	16	u(1)
nal_unit_type		u(6)
NumBytesInRBSP = 0		
layer_scenario	64	u(2)
layer_id	60 34	u(6)
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x0000003) {		
rbsp_byte[NumBytesInRBSP++]		b(8)
rbsp_byte[NumBytesInRBSP++]		b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */		f(8)
} else		
rbsp_byte[NumBytesInRBSP++]		b(8)
}		
}		

40

FIG 4

6/17

42

video_parameter_set_rbsp() {	Descriptor
video_parameter_set_id	u(8)
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
layer_id_type	u(7)
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data()	u(1)
rbsp_trailing_bits()	
}	

38

54

44

FIG 5

7/17

44

vps_extension_data() {	Descriptor
layer_id	u(6)
new_profile_level_flag	u(1)
reserved_zero_1_bit	u(1)
if (new_profile_level_flag == 1) {	
profile_idc	u(8)
reserved_zero_8bits /* equal to 0 */	u(8)
level_idc	u(8)
} else {	
ref_layer_id	u(6)
reserved_zero_2_bits	u(2)
}	
}	

FIG 6

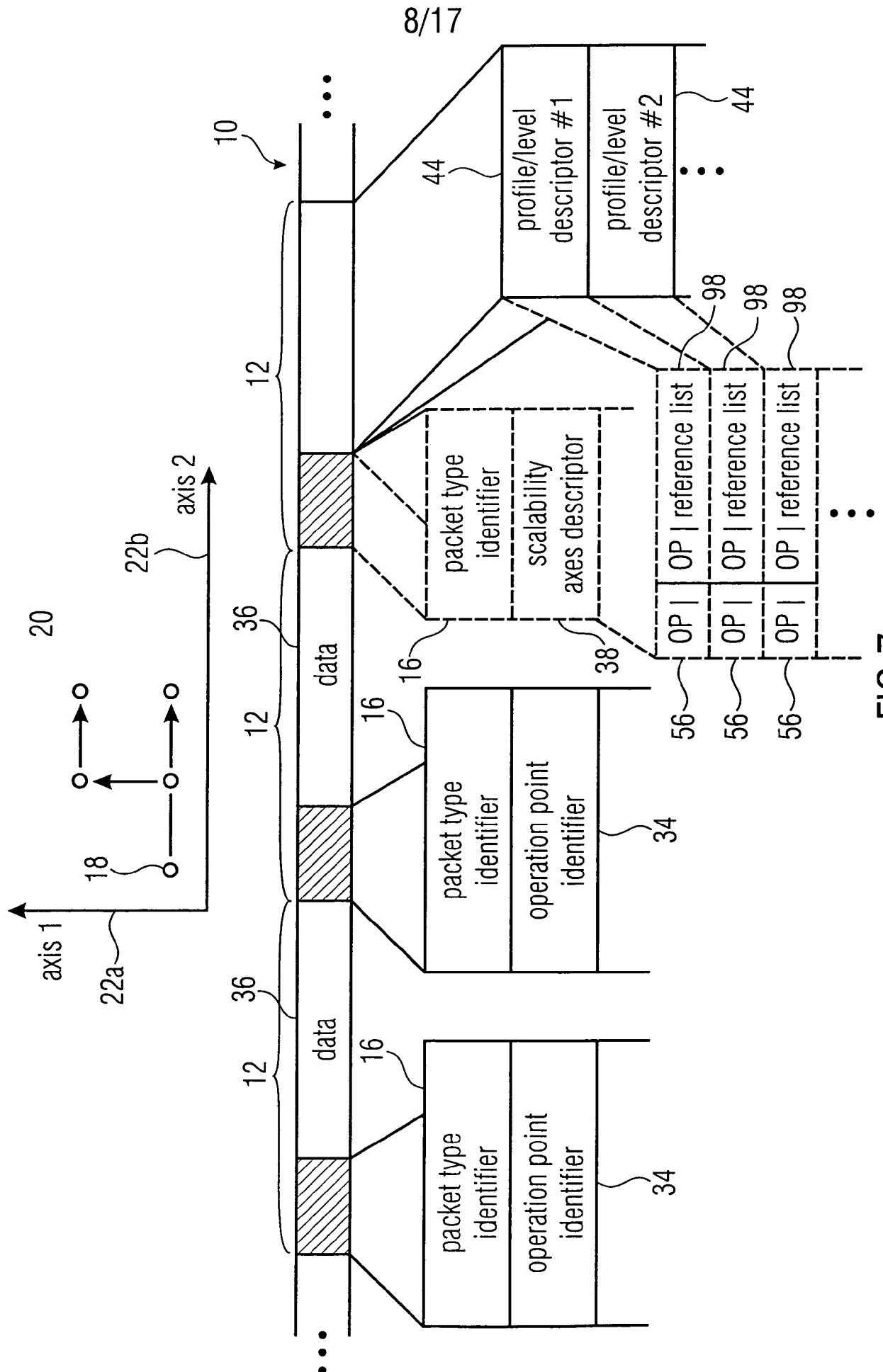


FIG 7

9/17

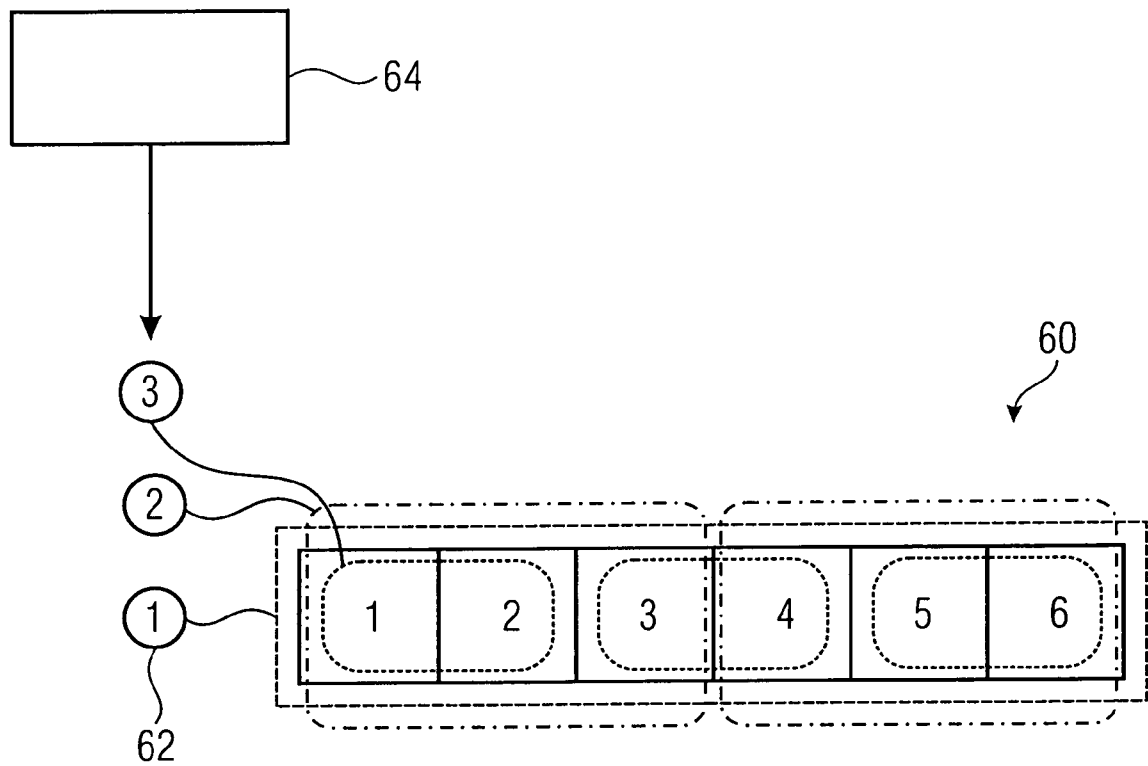


FIG 8

10/17

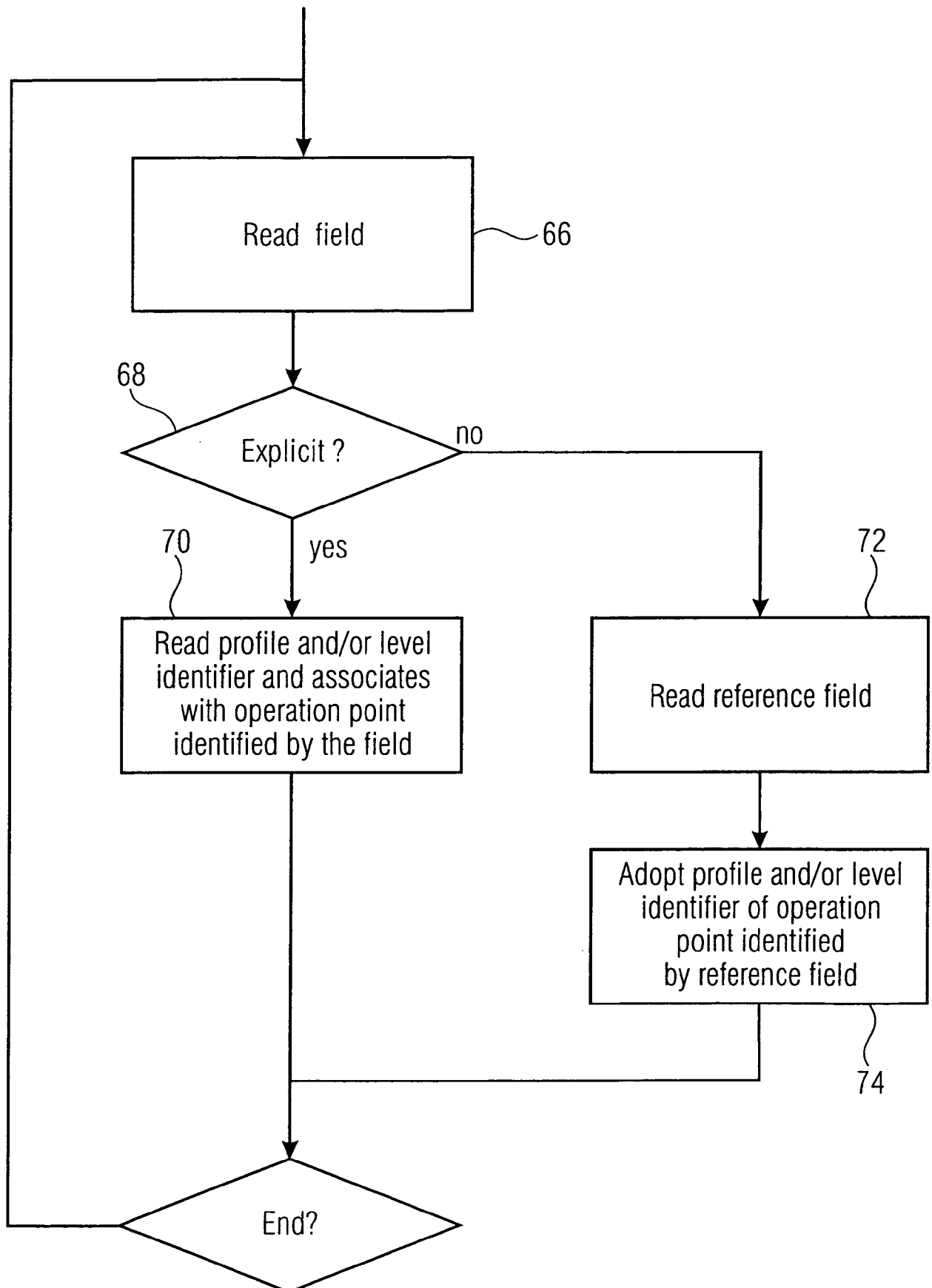


FIG 9

11/17

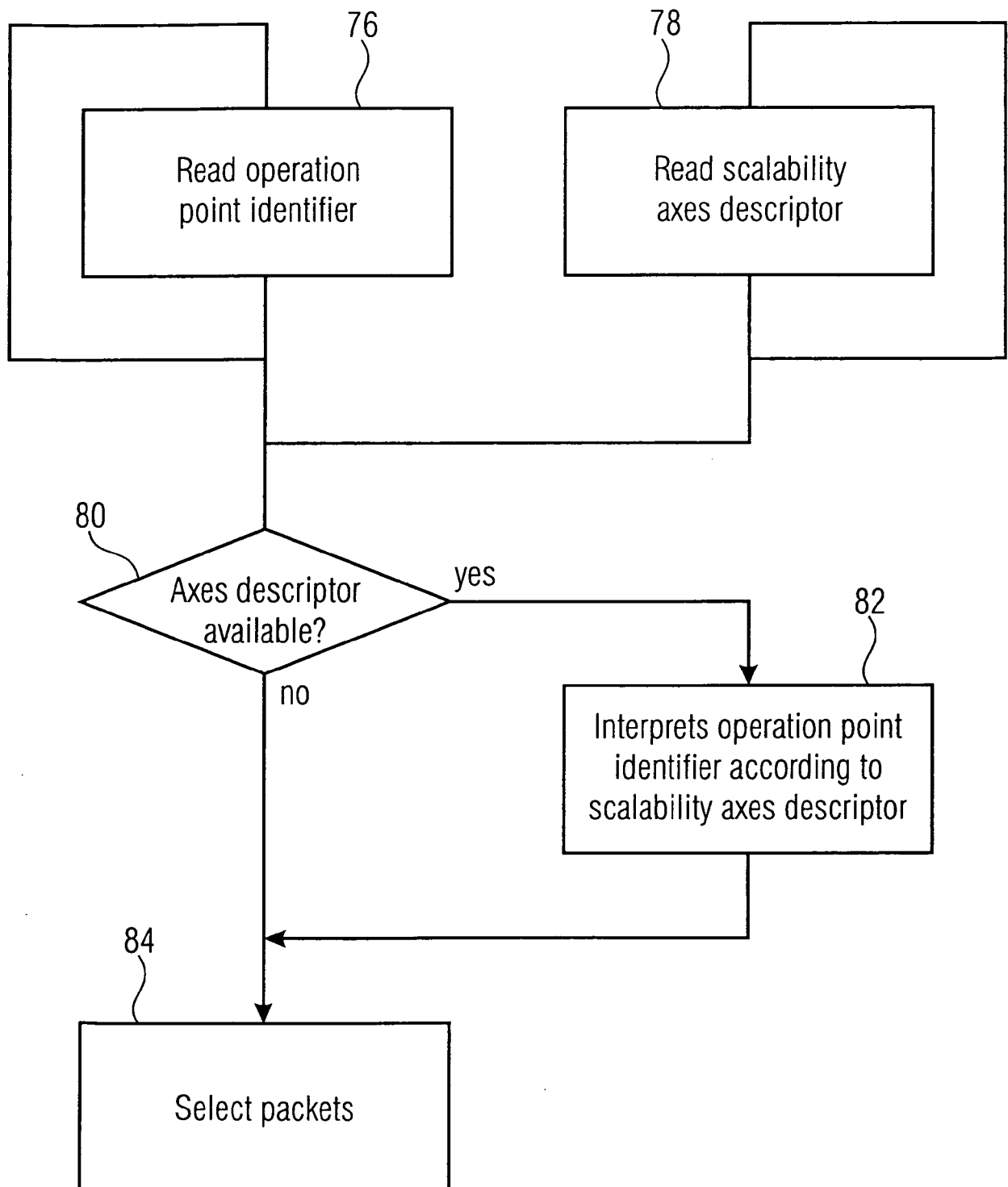


FIG 10

nal_unit(NumBytesInNALunit) {		Descriptor
forbidden_zero_bit		f(1)
nal_ref_flag		u(1)
nal_unit_type		u(6)
NumBytesInRBSP = 0		
temporal_id		u(3)
reserved_one_5bits		u(5)
layer_id_plus1		u(5)
for(i = nalUnitHeaderBytes; i < NumBytesInNALunit; i++) {		
if(i + 2 < NumBytesInNALunit && next_bits(24) == 0x0000003) {		
rbsp_byte[NumBytesInRBSP++]		b(8)
rbsp_byte[NumBytesInRBSP++]		b(8)
i += 2		
emulation_prevention_three_byte /* equal to 0x03 */		f(8)
} else		
rbsp_byte[NumBytesInRBSP++]		b(8)
}		
}		

FIG 11

13/17

seq_parameter_set_rbsp() {	Descriptor
—profile_space	u(3)
—profile_idc	u(5)
—constraint_flags	u(16)
—level_idc	u(8)
—for(i = 0; i < 32; i++)	
— profile_compatibility_flag[i]	u(1)
seq_parameter_set_id	ue(v)
video_parameter_set_id	ue(v)
chroma_format_idc	ue(v)
if(chroma_format_idc == 3)	
separate_colour_plane_flag	u(1)
—sps_max_temporal_layers_minus1	u(3)
pic_width_in_luma_samples	ue(v)
pic_height_in_luma_samples	ue(v)
pic_cropping_flag	u(1)
if(pic_cropping_flag) {	
pic_crop_left_offset	ue(v)
pic_crop_right_offset	ue(v)
pic_crop_top_offset	ue(v)
pic_crop_bottom_offset	ue(v)
}	
bit_depth_luma_minus8	ue(v)
bit_depth_chroma_minus8	ue(v)
pcm_enabled_flag	u(1)
if(pcm_enabled_flag) {	
pcm_sample_bit_depth_luma_minus1	u(4)
pcm_sample_bit_depth_chroma_minus1	u(4)

FIG 12A

FIG 12	FIG 12A
	FIG 12B
	FIG 12C

14/17

}	
log2_max_pic_order_cnt_lsb_minus4	ue(v)
for(i = 0; i <= sps_max_temporal_layers_minus1; i++) {	
sps_max_dec_pic_buffering[i]	ue(v)
sps_num_reorder_pics[i]	ue(v)
sps_max_latency_increase[i]	ue(v)
}	
restricted_ref_pic_lists_flag	u(1)
if(restricted_ref_pic_lists_flag)	
lists_modification_present_flag	u(1)
log2_min_coding_block_size_minus3	ue(v)
log2_diff_max_min_coding_block_size	ue(v)
log2_min_transform_block_size_minus2	ue(v)
log2_diff_max_min_transform_block_size	ue(v)
if(pcm_enabled_flag) {	
log2_min_pcm_coding_block_size_minus3	ue(v)
log2_diff_max_min_pcm_coding_block_size	ue(v)
}	
max_transform_hierarchy_depth_inter	ue(v)
max_transform_hierarchy_depth_intra	ue(v)
scaling_list_enable_flag	u(1)
if(scaling_list_enable_flag) {	
sps_scaling_list_data_present_flag	u(1)
if(sps_scaling_list_data_present_flag)	
scaling_list_param()	
}	
chroma_pred_from_luma_enabled_flag	u(1)
transform_skip_enabled_flag	u(1)
seq_loop_filter_across_slices_enabled_flag	u(1)

FIG 12B

FIG 12	FIG 12A
	FIG 12B
	FIG 12C

15/17

asymmetric_motion_partitions_enabled_flag	u(1)
nsrqt_enabled_flag	u(1)
sample_adaptive_offset_enabled_flag	u(1)
adaptive_loop_filter_enabled_flag	u(1)
if(adaptive_loop_filter_enabled_flag)	
alf_coef_in_slice_flag	u(1)
if(pcm_enabled_flag)	
pcm_loop_filter_disable_flag	u(1)
sps_temporal_id_nesting_flag	u(1)
if(log2_min_coding_block_size_minus3 == 0)	
inter_4x4_enabled_flag	u(1)
num_short_term_ref_pic_sets	ue(v)
for(i = 0; i < num_short_term_ref_pic_sets; i++)	
short_term_ref_pic_set(i)	
long_term_ref_pics_present_flag	u(1)
sps_temporal_mvp_enable_flag	u(1)
vui_parameters_present_flag	u(1)
if(vui_parameters_present_flag)	
vui_parameters()	
sps_extension_flag	u(1)
if(sps_extension_flag)	
while(more_rbsp_data())	
sps_extension_data_flag	u(1)
rbsp_trailing_bits()	
}	

FIG 12C

FIG 12	FIG 12A
	FIG 12B
	FIG 12C

16/17

42

video_parameter_set_rbsp() {	Descriptor
video_parameter_set_id	u(8)
scalability_type	u(4)
vps_temporal_id_nesting_flag	u(1)
vps_max_temporal_layers_minus1	u(3)
profile_space	u(3)
profile_idc	u(5)
constraint_flags	u(16)
for(i = 0; i < 32; i++)	
profile_compatibility_flag[i]	u(1)
for(i = 0; i < vps_max_temporal_layers_minus1; i++) {	
level_idc[i]	u(8)
vps_max_dec_pic_buffering[i]	ue(v)
vps_num_reorder_pics[i]	ue(v)
vps_max_latency_increase[i]	ue(v)
}	
for(n = 0; n < max_dim(scalability_type); n++) {	
layer_id_plus1_dim_len[n]	u(3)
}	
vps_extension_flag	u(1)
if(vps_extension_flag)	
while(more_rbsp_data())	
vps_extension_data()	u(1)
rbsp_trailing_bits()	
}	

FIG 13

SUBSTITUTE SHEET (RULE 26)

17/17

44

vps_extension_data() {	Descriptor
vps_extension_type	u(2)
layer_id_plus1	u(5)
max_temporal_layers_minus1	u(3)
new_profile_level_flag	u(1)
if (new_profile_level_flag == 0) {	
profile_reference_layer_id_plus1	u(5)
}	
else {	
profile_space	u(3)
profile_idc	u(5)
constraint_flags	u(16)
for(i = 0; i < 32; i++)	
profile_compatibility_flag[i]	u(1)
for(i = 0; i < max_layer_tmp_level; i++)	
level_idc[i]	u(8)
}	
num_reference_layers	u(5)
for(n = 0; n < num_reference_layers; n++)	
direct_coding_dependency_layer_id_plus1[n]	u(5)
}	

FIG 14