INTERNATIONAL STANDARD

ISO/IEC 14496-15

Information technology — Coding of audio-visual objects —

Part 15:
Advanced Video Coding (AVC) file format

AMENDMENT 1: Sub-track definitions

Technologies de l'information — Codage des objets audiovisuels —

Partie 15: Format de fichier de codage vidéo avancé (AVC)

AMENDEMENT 1: Définitions des sous-pistes

Foreword

ISO (the International Organization for Standardization) and IEC (the International Electrotechnical Commission) form the specialized system for worldwide standardization. National bodies that are members of ISO or IEC participate in the development of International Standards through technical committees established by the respective organization to deal with particular fields of technical activity. ISO and IEC technical committees collaborate in fields of mutual interest. Other international organizations, governmental and non-governmental, in liaison with ISO and IEC, also take part in the work. In the field of information technology, ISO and IEC have established a joint technical committee, ISO/IEC JTC 1.

International Standards are drafted in accordance with the rules given in the ISO/IEC Directives, Part 2.

The main task of the joint technical committee is to prepare International Standards. Draft International Standards adopted by the joint technical committee are circulated to national bodies for voting. Publication as an International Standard requires approval by at least 75 % of the national bodies casting a vote.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. ISO and IEC shall not be held responsible for identifying any or all such patent rights.

Amendment 1 to ISO/IEC 14496-15:2010 was prepared by Joint Technical Committee ISO/IEC JTC 1, *Information technology*, Subcommittee SC 29, *Coding of audio, picture, multimedia and hypermedia information*.

Information technology — Coding of audio-visual objects —

Part 15:
Advanced Video Coding (AVC) file format

AMENDMENT 1: Sub-track definitions

In Clause 5, add the following subclause:

5.3.17 Post-decoder requirements scheme for signalling of SEI for AVC

5.3.17.1 General

In order to handle situations where the file author requires certain actions on the player or renderer, the ISO base media file format specifies the restricted-video mechanism where sample entries are hidden behind the generic sample entry 'resv'. The mechanism applies to AVC and for this case a file author can list occurring SEI message IDs [ISO/IEC 14496-10] and classify them into two categories: those that are deemed required by the file author for correct playback, and others. The occurrence of either type of SEI messages can be signalled in the SEI Information box.

The scheme for signalling of SEI for AVC is defined here.

The SchemeType 'aSEI' (AVC SEI) is used.

5.3.17.2 SEI Information box

5.3.17.2.1 Definition

Box Type: 'seii'
Container: Scheme Information box ('schi')
Mandatory: Yes (when the SchemeType is 'aSEI')
Quantity: One

The SEI Information box is contained in the Scheme Information box when the SchemeType is 'aSEI', and contains information about the SEI messages present in the AVC stream. Although the SEI messages are not required for decoding, the file author may require certain actions for rendering or other purposes. The box distinguishes between SEI which is required to be understood for correct playback and SEI which is not required for correct playback (but may enhance playback).

By inspecting the SEI Information box a player will know which SEI messages that occur in the bitstream and whether they are required for rendering.

When the coding system is AVC and the restriction is signaled with SEI messages, then the SEI messages listed here should be stored either in the bitstream or in the AVC Configuration Record. The SEI Information box does not contain the actual SEI messages, it only lists those that occur in the bitstream.

5.3.17.2.2 Syntax

```
aligned(8) class SeiInformationBox extends Box('seii') {
   unsigned int(16) numRequiredSEIs;
   for (i=0; i<numRequiredSEIs; i++) {
      unsigned int(16) requiredSEI_ID;
   }
   unsigned int(16) numNotRequiredSEIs;
   for (i=0; i<numNotRequiredSEIs; i++) {
      unsigned int(16) notrequiredSEI_ID;
   }
}
```

5.3.17.2.3 Semantics

requiredSEI_ID takes on the value "payloadType" of an SEI message present in the AVC stream that
is deemed necessary by the file author for correct playback.

notrequiredSEI_ID takes on the value "payloadType" of an SEI message present in the AVC stream
that is not deemed necessary by the file author for correct playback.

*Change the title of Annex C from "SVC and MVC sample group definitions" to "SVC and MVC sample group
and sub track definitions".*

In Annex C, add the following subclause:

C.6 Sub track definitions

C.6.1   General

Tracks may be divided into sub tracks that can be assigned alternate and switch groups that indicate whether
those (sub) tracks are alternatives to each other and whether one can switch between them during a session.
Alternate and switch groups can consist of sub tracks as well as entire tracks.

Codec-specific sub track definitions for SVC and MVC are defined below. If more than one sub track definition
is present for a sub track, the union of the sub track definitions defines the sub track.

C.6.2   SVC Sub Track Layer box

C.6.2.1 Definition

Box Type: 'sstl'
Container: Sub Track Definition box ('strd')
Mandatory: No
Quantity: Zero or more

C.6.2.2 Syntax

```
aligned(8) class SVCSubTrackLayerBox
   extends FullBox('sstl', 0, 0) {
   unsigned int(16) item_count;
   for(i = 0; i< item_count; i++) {
      unsigned int(3)   dependency_id;
      unsigned int(4)   quality_id;
      unsigned int(3)   temporal_id;
      unsigned int(6)   priority_id;
      bit(2)   dependency_id_range;
      bit(2)   quality_id_range;
      bit(2)   temporal_id_range;
      bit(2)   priority_id_range;
   }
}
```

C.6.2.3 Semantics

The provided ranges of SVC layer parameters `dependency_id`, `quality_id`, `temporal_id` and `priority_id` (DQTP) specify the parts of the track that belong to the sub track. A unique combination of DQTP determines an SVC layer. The union of different DQTP values (and therefore the union of SVC layers) describes the sub track defined by this box.

`item_count` counts the number of DQTP quadruplets listed in this box.
`dependency_id` indicates the dependency_id value of the NAL units.
`quality_id` indicates the quality_id value of the NAL units.
`temporal_id` indicates the temporal_id value of the NAL units.
`priority_id` indicates the priority_id value of the NAL units.
`dependency_id_range` indicates the range of dependency_id values that belong to the sub track.
`quality_id_range` indicates the range of quality_id values that belong to the sub track.
`temporal_id_range` indicates the range of temporal_id values that belong to the sub track.
`priority_id_range` indicates the range of priority_id values that belong to the sub.

Each SVC layer parameter provides one value that together with the corresponding range parameter specifies the SVC layer parameter values that belong to the sub track. For each range indication, those values are

0x00   exactly equal to the specified value,
0x01   less than or equal to the specified value,
0x02   greater than or equal to the specified value,
0x03   any, i.e., the parameter is not specified.

C.6.3   MVC Sub Track View box

C.6.3.1 Definition

Box Type: 'mstv'
Container: Sub Track Definition box ('strd')
Mandatory: No
Quantity: Zero or more

C.6.3.2 Syntax

```
aligned(8) class MVCSubTrackViewBox
    extends FullBox('mstv', 0, 0) {
    unsigned int(16) item_count;
    for(i = 0; i< item_count; i++) {
        unsigned int(10)  view_id;
        unsigned int(4)   temporal_id;
        unsigned int(2)   reserved;
    }
}
```

C.6.3.3 Semantics

The list of `view_id` and `temporal_id` (VT) pairs specifies the parts of the track that belong to the sub track. A combination of VT determines the MVC layer. Hence, each VT pair listed in the MVC Sub Track View box determines a single MVC operating point. The union of different VT pairs of values (and therefore the union of MVC views at particular temporal resolution) describes the sub track defined by this box.

 `item_count` counts the number of VT pairs listed in this box.
 `view_id` indicates the view_id value in the MVC NAL unit header.
 `temporal_id` indicates the temporal_id value in the MVC NAL unit header.

C.6.4   SVC and MVC Sub Track Tier box

C.6.4.1 Definition

Box Type: `'stti'`
Container: Sub Track Definition box (`'strd'`)
Mandatory: No
Quantity: Zero or more

C.6.4.2 Syntax

```
aligned(8) class SubTrackTierBox
    extends FullBox('stti', 0, 0) {
    unsigned int(16) item_count;
    for(i = 0; i< item_count; i++) {
        unsigned int(16)  tierID;
    }
}
```

C.6.4.3 Semantics

The union of `tierID`s in this box describes the sub track defined by this box. The tier can be either an SVC or an MVC tier.

 `item_count` counts the number of tiers listed in this box.
 `tierID` gives the identifier of the tier(s) contained in this sub track.