INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO

ISO/IEC JTC1/SC29/WG11 MPEG2013/M30293
July 2013, Vienna (AT)

Source **Telecom ParisTech, Canon Research Centre France**
Status **For consideration at the 105th MPEG meeting**
Title **Coding Dependencies Signaling in ISOBMFF**
Authors Jean Le Feuvre, Cyril Concolato, Franck Denoual, Frédéric Mazé, Eric Nassor

1 Introduction

During our investigations on how to store HEVC tiles in the ISO Base Media File Format, we have found out that ISO Base Media Files lack a tool to indicate that some tracks are dependent from other tracks in terms of decoding, and should therefore have their samples decoded after the samples of the track they depend on. This contribution proposes some use cases, a problem description and a proposal of a tool allowing for describing a complete dependency graph, including decoding order, of a multi-track media representation in ISOBMFF, together with rules for media processing of such dependencies.

2 Use cases

Many use cases require that a precise order of track processing is needed. The use cases include:

- Simple scalability (spatial, SNR, temporal) where each scalable level is stored in a dedicated track, each layer being dependent on a single lower layer. Each layer has to be passed to the media processor in order, starting from the base.
- Independent decoding of spatial areas in a video stream, similar to independent tiles in HEVC. In this case, each tile can be processed independently, in any order but has to be processed after the track containing non-VCL NAL units.
- Advanced scalability such as SNR refinements of region of interests, where several regions can be defined and decoded independently of each other;
- Scalable hybrid coding where base layer and enhancement layers do not use the same media format (similar to SHVC design).

3 Possible solutions and problems

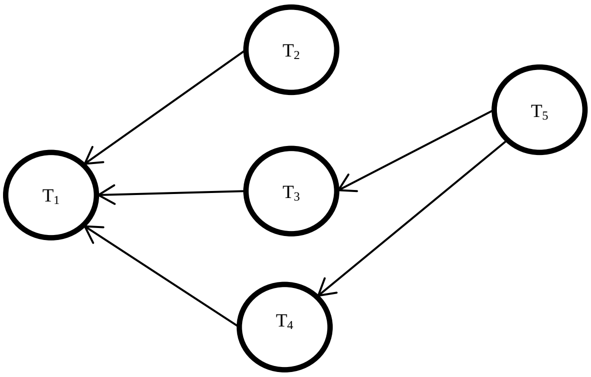
We have investigated the existing tools and evaluated:

- The track dependencies such as ‘scal’ and ‘dpnd’.
- The use of extractors

The MPEG-4 Systems standard has the notion of dependsOn_ESID in the decoderConfigurationDescriptor, but this was translated in the mp4 file format as a simple ‘dpnd’ track dependency, whose exact processing semantics are given by MPEG-4 Systems.

The ‘scal’ dependency is used in ISO/IEC 14496 Part 15 for SVC and MVC coding formats but seems never defined/registered (in opposite to other track reference types in Part 12). In addition, it seems closely related to the extractors (Annex A.3) but it is not clear whether it can be used without extractors. And if it could be used without extractors, the processing order could not be specified.

In both cases, in a file with 5 tracks T_i with dependencies such that T2, T3 and T4 depend on T1 but not on each other, and T5 depends on T3 and T4, references (‘scal’ or ‘dpnd’) will be set as follows, without any notion of order:



Decoding T5 will therefore give a dependency graph of:

T5-> [T3, T4] or [T4, T3] -> T1

As can be seen, T2 will not be in the dependency graph with such a mechanism and therefore reconstruction of the complete stream from these dependencies is problematic:

- Implies the scan of all tracks to discover T2->T1 dependency
- Still misses the order in which T2 should be processed

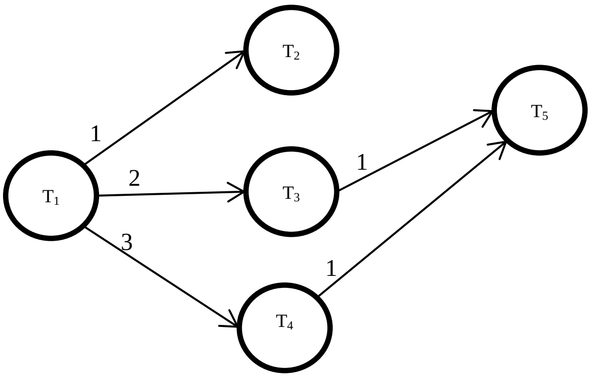
As a summary, while they cover some of the required features, ‘dpnd’ and ‘scal’ only describe downwards dependencies (from higher layer to lower), and fail to address complex cases where decoding order is required between some layers but not fixed between other layers.

Extractors have some limitations which are not satisfactory for the proposed use cases:

- In most cases a fixed pattern of base and enhancement (or non-VCL + tile) is observed, and therefore a simple rule for concatenation is sufficient. Such a simple rule also reduces the overhead (no need for extractor NALUs).
- Extractors cannot point to extractors (see subclause A.3.1), and therefore the N-th scalable layer requires N extractors per sample to aggregate the complete stream.
- Additionally, extractors are not generic enough and cannot for instance be used for HEVC or SHVC.

4 Proposal

From the previous investigations, we propose to describe dependencies using track references going upward. For the previous example, the dependency graph would become (note the numbering on arrows, which gives the order in which samples shall be processed):



In Subclause "**8.3.3.3 Semantics**", add the following:

"

'dond': a 'dond' reference defines a dependency subtree between the referencing track and the reference tracks. Referenced tracks may also have a 'dond', defining other subtrees. A track having a 'dond' and not being referenced by any 'dond' is the root of a dependency tree. For any media time MT, if there is a sample in one of the tracks in the dependency tree, the sample from $\mathrm{Track}_{\mathrm{tref[i]}}$ with media time MT shall be passed to the media processor before the sample, if any, from $\mathrm{Track}_{\mathrm{tref[i+1]}}$ with media time MT but after the sample, if any, with media time MT of the referring track. Starting from the root, all tracks in one level of the dependency tree are handled first, and tracks at a deeper level are then handled. If multiple dependencies referring to the same track are found, only the samples corresponding to first occurrence of the reference shall be passed to the media processor.

For example, if a file has 5 tracks T_i with dependencies such that T2, T3 and T4 depend on T1 but not on each other, and T5 depends on T3 and T4, three 'dond' references will be used:

'dond' on T1 indicates T2, T3 and T4
'dond' on T3 indicates T5
'dond' on T4 indicates T5

and the processing order of samples S_i at time MT_1 will be

$S_1(MT_1)$, $S_2(MT_1)$, $S_3(MT_1)$, $S_4(MT_1)$, $S_5(MT_1)$

If at time MT2 there is no sample on the track T_1 and no sample on the track T_4, the processing order of samples will be:

$S_2(MT_2)$, $S_3(MT_2)$, $S_5(MT_2)$

It is an error to have 'dond' references between tracks that do not have the same media handler, but it is allowed to have 'dond' references between tracks that do not have the same sample description types. Whether the media samples have to be physically concatenated or not is both media- and implementation- specific.

It is allowed to have a track with no samples used to carry a 'dond' track reference; this allows for example describing coding hierarchy of HEVC tiles stored in separated tracks in an ISOBMF file with all parameter sets stored in the sample description of an empty track carrying the 'dond'.

Tracks referenced by ‘dond’ may not be processable individually, in which case, for backward compatibility reasons, these tracks shall be marked as disabled. A ‘dond’ aware media processor may decide to play all or part of the disabled tracks.

NOTE: an ‘sbas’ dependency type could be used on referenced tracks to indicate the root of the dependency tree.
”

5 Conclusion

We suggest adding the new dependency type in the ISO BMFF to allow for flexible description of coded-dependent media or to provide simple sample data partitioning.