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

ISO/IEC JCTC1/SC29/WG11 MPEG/m51025
October 2019, Geneva, Switzerland

Source:	Apple Inc.	
Status:	Input document	
Title:	G-PCC multi-frame sequence coding	
Author(s):	David Flynn	*davidflynn@apple.com*
	Khaled Mammou	*kmammou@apple.com*

Abstract

The current (draft 7) G-PCC specification [1] does not include any support for multi-frame sequences. While the codec is currently an intra-frame only codec, there is no internal mechanism to identify frame boundaries. As such, decoding of a multi-frame sequence requires assistance from the file format to identify either what payloads belong to a particular frame, or where the boundary between two frames are.

This contribution proposes a mechanism to allow disambiguation of consecutive frames using the slice header.

Introduction

The current decoding model is based upon two layers. A codec layer that operates on particular data units in a particular order and an encapsulation layer that assembles a byte stream for transmission (such as a file format).

To decode a frame from a single-frame sequence, the decoder is passed data units until reaching the end of the sequence. The data units themselves represent parameter sets, geometry data, and attribute data. When there is no more data to be decoded, the decoder is flushed and the output frame retrieved.

Proposal

The proposed solution is to add a global frame counter to the codec state and to signal a number of LSBs (least significant bits) of this counter in the slice header, akin to the existing picture order count (POC) of AVC/HEVC. The number of LSBs is indicated in the SPS. All slices belonging to the same frame use an identical frame index variable.

The following constraints would apply to the ordering of data units:

- The data units of two frames may not be interleaved.
- The frame counter is monotonically increasing.
- Two adjacent frames may not use the same value of the frame counter as indicated by the LSBs in the slice header.

A future edition of the standard may further describe how the frame counter relates to any decoding processes. It is not suggested to tie the frame counter to an interpretation of frame rate. Rather this should be handled by a higher layer.

The effect of these constraints is to allow a decoder to detect the frame boundary between consecutive frames by spotting a change in the frame counter. Furthermore, individual slices may be rearranged without any

effect on decoding.

It is only necessary to add the frame counter variable to the geometry slice header since any attribute slices have a decoding dependency upon an identified geometry slice. However, there may be a benefit in terms of error detection in case of data loss to add the variable to all slice headers.

The following simple example uses one LSB bit to identify frame boundaries:

```
       SPS ... G A G A G A G A
slice:         0 0 1 1 0 0 0 0
LSBs:          0 . 0 . 1 . 0 .
frame:         0 0 0 0 1 1 2 2
```

Frame boundary marker data unit

Certain applications requiring either 'low-delay' or where ambiguity may be introduced through bitstream manipulation (imagine playing every other frame in the above example), may benefit from being able to explicitly mark frame boundaries.

An empty frame boundary marker data unit can address these issues by causing the codec to reset any necessary state to force a frame boundary and flush any (partially) decoded frame.

Tile inventory

There is an issue as to how metadata in the tile inventory relates to multiple frames since there are two perfectly valid, yet contradictory interpretations:

- that the tile metadata is valid for the entire sequence and does not change, or
- that the metadata is valid for only a part of the sequence.

To address both these cases, it is suggested to add the frame index counter to the tile inventory data unit along with a time-to-live field that indicates a duration of validity.

Specification highlights

```
sequence_parameter_set() {
  ...
  log2_max_frame_idx = u(5)
  ...
}

geometry_slice_header() {
  ...
  frame_idx = u(n)
  ...
}
```

log2_max_frame_idx plus 1 specifies the number of bits used by the frame_idx syntax variable.

frame_idx specifies the log2_max_frame_idx + 1 least significant bits of a notional frame number counter. Consecutive slices with differing values of frame_idx form parts of different output point cloud frames. Consecutive slices with identical values of frame_idx without an intervening frame boundary marker data unit form parts of the same output point cloud frame.

References

[1] 3DG, "G-PCC Improvements," ISO/IEC JTC1/SC29/WG11, 127th meeting, Gothenburg, Tech. Rep. w18669, Jul. 2019.