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/m52527
January 2020, Brussels, Belgium

Source:	Apple Inc.	
Status:	Input document	
Title:	G-PCC: High-level syntax issues	
Author(s):	David Flynn	*davidflynn@apple.com*
	Khaled Mammou	*kmammou@apple.com*

Abstract

Todo: abstract

The following sections address a number of issues in no particular order.

Number of points per slice

The current draft specification [1] contains a per-slice indication of the number of decodable points (gsh_num_points). On multiple occasions, issues have been raised with this syntax element [2, 3] and may be summarised as:

- Implementation safety concerns: The value is not used by the decoding process, but an implementation may use the value to allocate buffers. If the value under-represents the number of points that are decodable, a buffer overflow may occur.
- Encoder burdens: The value is exp-golomb coded at the start of a slice. An encoder that performs in-loop lossy coding may add or remove points (RDO, in-tree quantisation). However, this either introduces buffering delays, or the need for careful buffer management in case the length of the field changes.
- Redundancy: It is intended that level limits will provide a maximum number of points per slice which may be used for resource allocation.

The field does, however, provide some utility in determining the number of points in a sequence, rather than the worst-case that can be determined by using the level limit for each slice.

To address implementation safety concerns, we may update the decoding process to decode up to gsh_num_points, or until the octree is complete, whichever is lower. For conformance, a bitstream geometry slice may not contain more points than gsh_num_points. However, it may contain fewer.

To address encoder burdens, we may update the field to be fixed length u(24) or u(32) for example[1]. Furthermore, we may choose to place the field at the end of the slice data. An encoder may then write the value out after geometry coding is complete.

Order of parameters

The geometry slice header order of the specification and test model is currently inconsistent. Items in the headers should be ordered in a manner that simplifies access for higher layers. For instance, a higher-layer

[1] While it is possible to link the length to a level limit, this seems to be unnecessary

may need to know the position of a slice. Whereas it is unlikely to care about, or at least to a lesser extent, the quantisation parameters.

It is recommended to order the geometry slice header as follows:

```
gsh_geometry_parameter_set_id = ue(v)
gsh_tile_id = ue(v)
gsh_slice_id = ue(v)
frame_idx = u(v)
gsh_num_points = ??(?)
if( gps_box_present_flag ) { ... }
if( gps_implicit_geom_partition_flag ) { ... }
if( geom_scaling_enabled_flag ) { ... }
```

Accidental disabling of IDCM

The adoption of planar mode contains a modification to IDCM:

```
+   // planar mode initialazation
+   const int th_idcm = gps.geom_planar_idcm_threshold * 127 * 127;
...
-      bool idcmEnabled = gps.inferred_direct_coding_mode_enabled_flag;
+      // decode planarity if eligible
+      int planarProb[3] = {127, 127, 127};
+      if (
+        gps.geom_planar_mode_enabled_flag
+        && (planarEligible[0] || planarEligible[1] || planarEligible[2]))
+        decoder.determinePlanarMode(
+          planarEligible, kNumPlanarPlanes, child, planes, node0.neighPattern,
+          x, y, z, planarProb, planarRate);
+
+      bool idcmEnabled = gps.inferred_direct_coding_mode_enabled_flag
+        && planarProb[0] * planarProb[1] * planarProb[2] <= th_idcm;
```

With the geometry parameter set as follows:

```
geom_planar_mode_enabled_flag = u(1)
geom_planar_idcm_threshold = ue(v)
if (geom_planar_mode_enabled_flag) {
  geom_planar_threshold0 = ue(v)
  geom_planar_threshold1 = ue(v)
  geom_planar_threshold2 = ue(v)
}
```

The result is that with planar mode disabled, in order to use IDCM, geom_planar_idcm_threshold must be signalled as 127, which is not necessarily obvious.

Recommendation: guard geom_planar_idcm_threshold by geom_planar_mode_enabled_flag and infer its value to be 127 if not present.

Behaviour of the tile inventory in multi-frame sequences

The current working draft contains a means to represent multi-frame sequences [4]. However, one aspect of the proposal was omitted from the presentation, namely what should the behaviour of the tile inventory in a multi-frame sequence:

> 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.

Recommendation: modify as originally suggested.

References

[1] 3DG, "G-PCC Future Enhancements," ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. w18887, Oct. 2019.

[2] D. Flynn, "Suggested changes to G-PCC high-level syntax," ISO/IEC JTC1/SC29/WG11, 125th meeting, Marrakech, Tech. Rep. m46530, Jan. 2019.

[3] S. . Secretariat, "Summary of voting on ISO/IEC CD 23090-9," ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. m49977, Oct. 2019.

[4] D. Flynn and K. Mammou, "[G-PCC][New proposal] G-PCC multi-frame sequence coding," ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. m51025, Oct. 2019.