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/m52520
January 2020, Brussels, Belgium

Source: Apple Inc.
Status: Input document
Title: G-PCC: Duplicate point handling in predictive geometry coding
Author(s): David Flynn | *davidflynn@apple.com*
Khaled Mammou | *kmammou@apple.com*

Abstract

The proposed octree geometry coder includes explicit support for signalling duplicate points. The previously proposed predictive geometry coder is capable of handling duplicate points implicitly due to the signalling of prediction residues in an arbitrary tree structure. The proposed scheme adds a duplicate point count to each node rather than having to use multiple nodes to enumerate all the duplicated points.

Introduction

The predictive geometry scheme [1] uses a depth-first tree with each node having the following properties:

- a prediction mode (eg delta from parent)
- a residual that is combined with the prediction to generate the position of a single point, and
- a maximum of three child nodes.

In order to signal duplicate points with the currently proposed scheme, one additional node is required for each duplicate point. The prediction mode is signalled as delta-from-parent and the residual is set to 0.

Adding a duplicate point count

For sequences with large numbers of duplicate points the inefficiency of processing a node for each duplicate point can be reduced by adding a per-node count of the number of duplicate points.

This is signalled using an exp-golomb code as follows:

```
+int
+PredGeomDecoder::decodeNumDuplicatePoints()
+{
+  if (!_aed->decode(_ctxNumDupPointsGt0))
+    return 0;
+  return 1 + _aed->decodeExpGolomb(0, _ctxBypass, _ctxNumDupPoints);
+}
```

Encoder implementation

The current encoder implementation handles sequential duplicate points by scanning the input stream for subsequent duplicate points:

```
-  for (int32_t nodeIdx = 0; nodeIdx < pointCount; ++nodeIdx) {
+  for (int nodeIdx = 0, nodeIdxN; nodeIdx < pointCount; nodeIdx = nodeIdxN) {
     auto& node = nodes[nodeIdx];
```

```
      auto queryPoint = cloud[nodeIdx];

+     // scan for duplicate points
+     node.numDups = 0;
+     for (nodeIdxN = nodeIdx + 1; nodeIdxN < pointCount; nodeIdxN++) {
+       if (queryPoint != cloud[nodeIdxN])
+         break;
+       node.numDups++;
+     }
+
```

Results

The proposed method has been integrated into the high-latency variant of the predictive geometry encoder studied in CE13.22 [2].

Table 1 illustrates the performance of the proposed method relative to the CE13.22 result according to the common test conditions [3]. Since only the QNX cat3-frame sequences contain duplicate points, Table 2 shows the per sequence results for these sequences. In both tables, results according to condition CY (near-lossless attribute coding) are not reported since the distortion curves do not overlap sufficiently.

Table 1 – Performance of proposed duplicate point signalling relative to proposed predictive geometry coder

		BPP Ratio [%]			BD-Rate [Δ%]						Avg. of ratio maxrssk [%]		Ratio of avg. runtime [%]	
Condition	Class	Geometry	Colour	Refl	D1	D2	Y	Cb	Cr	R	Encoder	Decoder	Encoder	Decoder
C1_ai	cat1-A						0.0	0.0	0.0		101	100	102	98
C1_ai	cat3-fused						0.0	0.0	0.0	0.0	99	100	100	95
C1_ai	cat3-frame									3.5	94	99	99	102
C1_ai	overall						0.0	0.0	0.0	2.4	99	100	101	99
C2_ai	cat1-A				0.2	0.2	0.0	0.0	0.0		100	100	101	100
C2_ai	cat1-B				0.1	0.1					100	100	104	
C2_ai	cat3-fused				0.1	0.1	0.0	0.0	0.0	0.0	100	100	99	97
C2_ai	cat3-frame				0.1	0.1				0.0	96	101	107	101
C2_ai	overall				0.1	0.1	0.0!	0.0!	0.0!	0.0	100	100	103	
CW_ai	cat1-A	100.1	100.0								100	100	102	101
CW_ai	cat1-B	100.0									101	100	104	102
CW_ai	cat3-fused	100.0	100.0	100.0							102	100	100	92
CW_ai	cat3-frame	98.0		99.1							94	99	103	103
CW_ai	overall	99.7	100.0!	99.5							100	100	103	101

Table 2 – Performance of proposed duplicate point signalling relative to proposed predictive geometry coder

		BPP Ratio [%]			BD-Rate [Δ%]						Avg. of ratio maxrssk [%]		Ratio of avg. runtime [%]	
Condition	Sequence	Geometry	Colour	Refl	D1	D2	Y	Cb	Cr	R	Encoder	Decoder	Encoder	Decoder
C1_ai	qnxadas-junction-approach						0.0	0.0	0.0	5.9	89	99	102	101
C1_ai	qnxadas-junction-exit						0.0	0.0	0.0	8.4	87	98	92	96
C1_ai	qnxadas-motorway-join						0.0	0.0	0.0	8.3	88	98	92	98
C1_ai	qnxadas-navigating-bends						0.0	0.0	0.0	1.7	89	99	97	101
C2_ai	qnxadas-junction-approach				0.1	0.1	0.0	0.0	0.0	0.0	93	106	104	109
C2_ai	qnxadas-junction-exit				0.1	0.1	0.0	0.0	0.0	0.0	93	101	105	95
C2_ai	qnxadas-motorway-join				0.1	0.1	0.0	0.0	0.0	0.0	93	101	103	103
C2_ai	qnxadas-navigating-bends				0.1	0.1	0.0	0.0	0.0	0.0	93	102	109	102
CW_ai	qnxadas-junction-approach	96.0	0.0	97.8							89	99	92	99
CW_ai	qnxadas-junction-exit	95.0	0.0	95.4							87	98	93	99
CW_ai	qnxadas-motorway-join	95.6	0.0	97.8							88	99	116	102
CW_ai	qnxadas-navigating-bends	97.4	0.0	98.3							88	98	99	108

References

[1] D. Flynn, A. Tourapis, and K. Mammou, “[G-PCC][New proposal] Predictive Geometry Coding,” ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. m51012, Oct. 2019.

[2] D. Flynn and K. Mammou, “G-PCC CE13.22 report on predictive geometry coding,” ISO/IEC JTC1/SC29/WG11, 129th meeting, Brussels, Tech. Rep. m52515, Jan. 2020.

[3] 3DG, “Common Test Conditions for PCC,” ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. w18883, Oct. 2019.