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/m52519
January 2020, Brussels, Belgium

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

Abstract

The predictive geometry coder was suggested as a completely independent coding method at the 128th meeting. However, there is considerable reticence to providing another independent method despite its benefits for certain application areas. This contribution investigates a combination of the predictive tree and octree codecs where octree coding is performed first until a predictive geometry node size is reached, and then predictive tree coding is performed within each octree node.

Introduction

Octree coding is a method that represents a sparse geometry by encoding the structure of an octree. The octree has a known depth, and therefore represents a known volume 2^{3d}. By construction, each octree node has between one and eight child nodes. The position of a point is encoded as the path through the octree. Each node of the tree contains a bitmap representing which child nodes are present. This bitmap is called the occupancy map, or occupancy information for a node.

The predictive occupancy coder [1] represents positions as nodes in a tree. The tree is traversed depth-first, 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
- A maximum of three child nodes

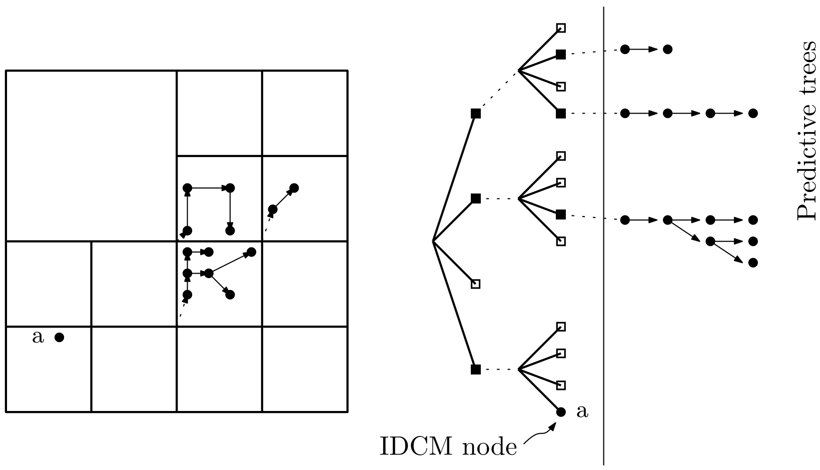

Figure 1 – Illustration of combined octree – predictive tree coding

Necessary modifications to the predictive geometry coder

The originally proposed predictive tree coder uses information about the number of points in a slice to determine how many predictive tree nodes to decode. However, when combined with the octree coding scheme as below, this is insufficient. Other than memory allocation, the main use for this is to determine if there are additional independent predictive trees to decode. In order to avoid signalling a per tree point count, we propose to signal at the end of each tree a flag indicating if it is the last tree in a set or if at least one additional tree follows.

Integration with the octree

The proposed method for a straightforward integration with the geometry octree is to split coding into two distinct phases, similar to the trisoup process, as illustrated in Figure 1.

We introduce a pred_geom_node_size_log2 variable into the high-level syntax that indicates when node size at which coding should transition between octree coding and the predictive tree coding.

At the transition depth, each octree node acts as the seed for one or more predictive geometry trees. The scaled octree node position is used as the local origin of the predictive tree.

In the current encoder, the predictive tree codes points that exist wholly within the octree node. However, the syntax does not preclude a predictive tree coding (stealing) points from adjacent tree.

The octree coding phase may use IDCM to code the exact location of isolated points, independently of the predictive tree coder. This is different to trisoup where IDCM must be disabled. Similarly, any octree subtrees that, due to in-tree quantisation, are coded as leaf-nodes do not act as seeds for a predictive tree.

Results

To evaluate the performance of the proposed method it is necessary to examine the behaviour of varying the predictive geometry node size. An experiment is performed using lossless geometry coding according to the common test conditions [2]. To reduce the size of the experiment, only sequences that appear to perform well in CE13.22 [3] are used. The non-normative encoder search method is the high-latency method.

The performance is illustrated in Figure 2 using a range of node sizes between 2^5 and 2^{18} (again, limited to reduce the experiment size). All compression ratios are relative to the TMC13v8.0 anchor results [4]. Dotted lines illustrate the CE13.22 results.

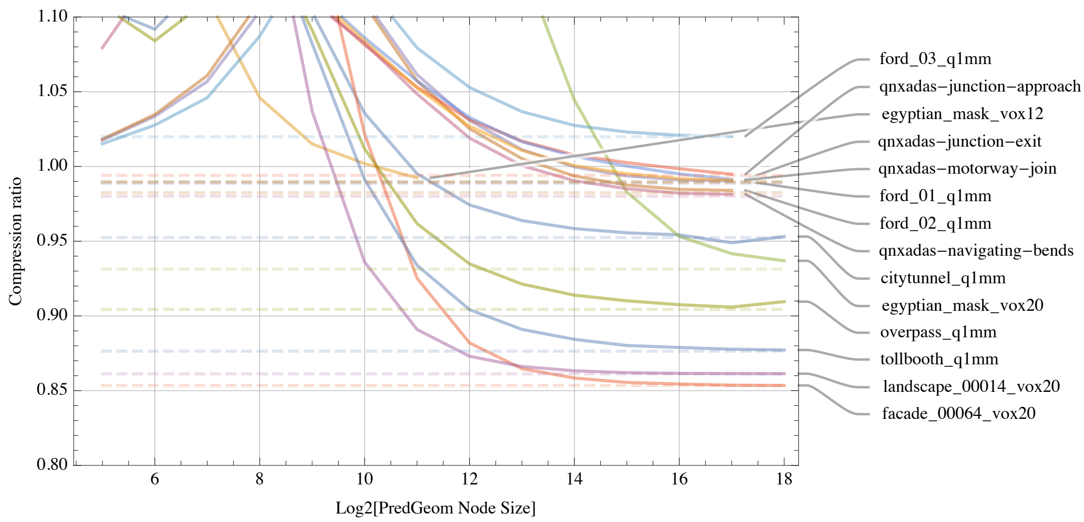

Figure 2 – Compression performance of combined scheme for various predictive geometry node sizes. Dotted lines represent CE result.

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

[3] 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.

[4] 3DG, “G-PCC performance evaluation and anchor results,” ISO/IEC JTC1/SC29/WG11, 128th meeting, Geneva, Tech. Rep. w18885, Oct. 2019.