**INTERNATIONAL ORGANISATION FOR STANDARDISATION**

**ORGANISATION INTERNATIONALE DE NORMALISATION**

**ISO/IEC JTC 1/SC 29/WG 11**

**CODING OF MOVING PICTURES AND AUDIO**

**ISO/IEC JTC 1/SC 29/WG 11 m52503**

**Brussels, BE – January 2020**

|  |  |
| --- | --- |
| **Source:** | **Sony corporation** |
| **Title:** | **[G-PCC] [crosscheck] CE13.22 Predictive Geometry Coding** |
| **Authors:** | **Alexandre Zaghetto, Danillo B. Graziosi, Ali Tabatabai** |

# Introduction

In [1], the proponents present a predictive geometry coding scheme as an alternative to the octree-based approach. The predictive scheme is meant to offer the following advantages:

1. Low latency applications/streaming; and
2. Low complexity decoding.

**The proposed solution targets only Category 3 content.**

Three use cases are discussed:

1. high latency use-case slow mode;
2. high latency use-case fast mode; and
3. low latency use-case.

# Cross-check results

This document reports the crosscheck of Apple’s code [2], [3] provided for CE13.22 [4], which is an implementation of [1]. The source code was obtained from the following branches:

<mpeg128/ce13.22/predgeom-high-latency>

<mpeg128/ce13.22/predgeom-low-latency>

In comparison to [1], only two use cases are discussed in this CE, the high-latency encoder (no distinction between slow and fast mode is presented) and low-latency encoder. For the low-latency encoder, results for search ranges of 128 and 512 points are presented.

It was verified that for both encoders,

1. Apple’s implementation had a perfect match with the provided rate-distortion results.
2. In terms of average encoding and decoding time, there were some differences, but they don’t interfere in the evaluation of the proposed.

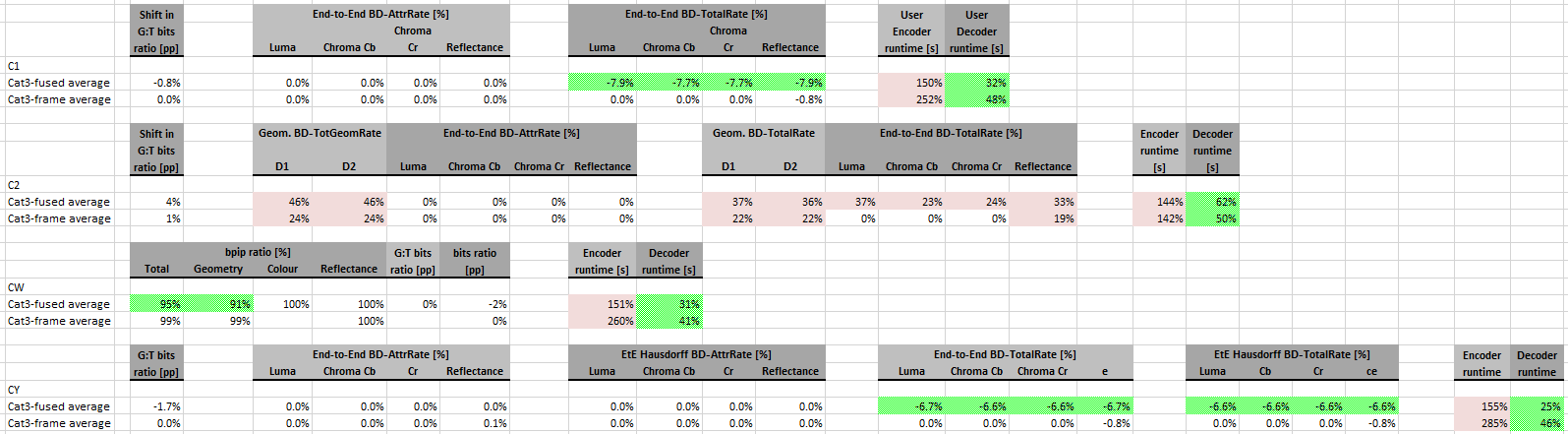
# PowerPoint files with summarized objective comparison and Excel files with detailed cross-checked results are attached to this report:

* CE13.22.Predictive Geometry Coding\_lowlatency\_sr512\_\_vs\_\_octree\_predlift\_summary
* CE13.22.Predictive Geometry Coding\_lowlatency\_sr512\_\_vs\_\_sr128\_summary
* predgeom-lowlatency-sr512\_xcheck\_\_vs\_\_sr128\_xcheck
* predgeom-lowlatency-sr512\_xcheck\_\_vs\_\_tmc13v8.0\_octree\_predlift\_xcheck

# Conclusion

# High-latency encoder (predgeom-high-latency)

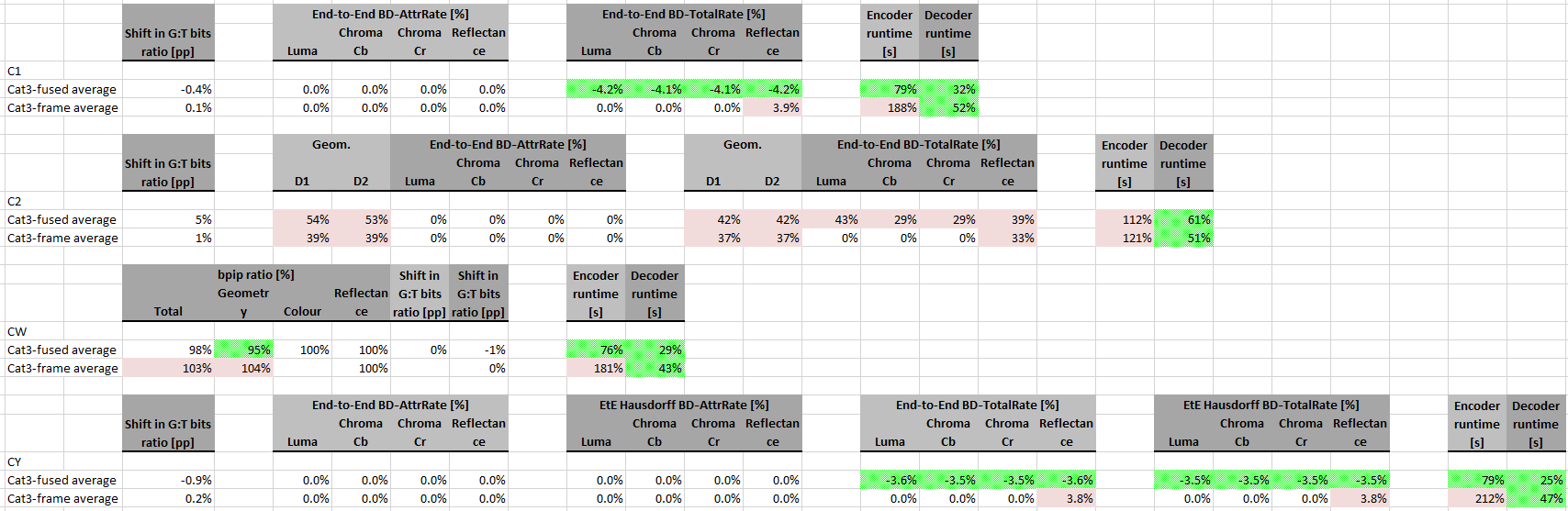
Compared to the TMC13v8.0\_octree-predlift, the average encoding time increases to something ranging from around 152% (C2) to 286% (C1). Decoding time decreases to something between 30% (CW) and 60% (C1). BD-TotalRate or bpip ratio significantly increase for almost all sequences, with the exception of cat3-fused (where gains are observed) and cat3-frame (where no significant changes are observed). The best case scenario is CW, where the average decoding time drops to around 30% and the total overall bpip ratio is 107.2%. The average encoding time, however, increases to 224%. Considering the premise that the proposed method targets only cat3 sequences, objective performance metrics for this category are highlighted below.



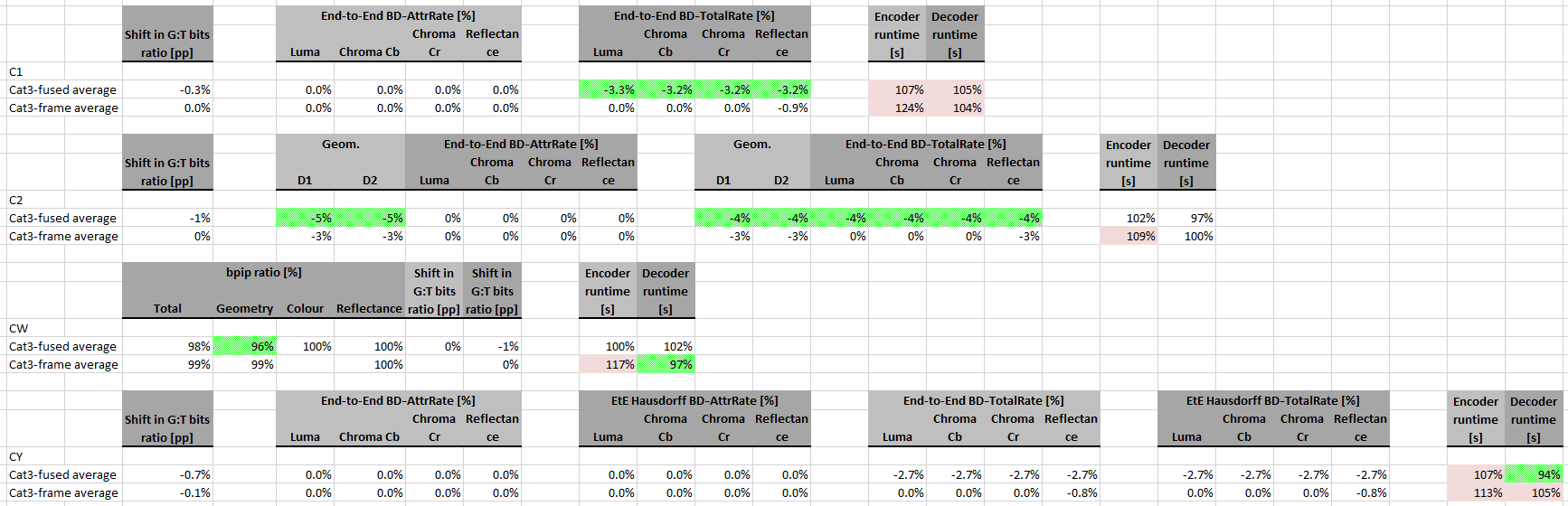
As one can see, the method seems not to be efficient only for C2. Big BD-TotalRate increases significantly.

# Low-latency encoder (predgeom-low-latency)

Regarding the low-latency encoder, two sets of results were provided: (a) low-latency encoder with 512 latency points versus TMC13v8.0\_octree-predlift; and (b) low-latency encoder using 512 versus 128 latency points. For the low-latency encoder configured to use 512 latency points, the results are significantly worse in terms of BD-TotalRate when compared to the high-latency encoder. However, for CW sequences average encoding time dropped to 112%, while decoding time remained around 30%. Furthermore, if only cat3 sequences is analyzed, the drop in encoding time is significant, as shown in the table below.

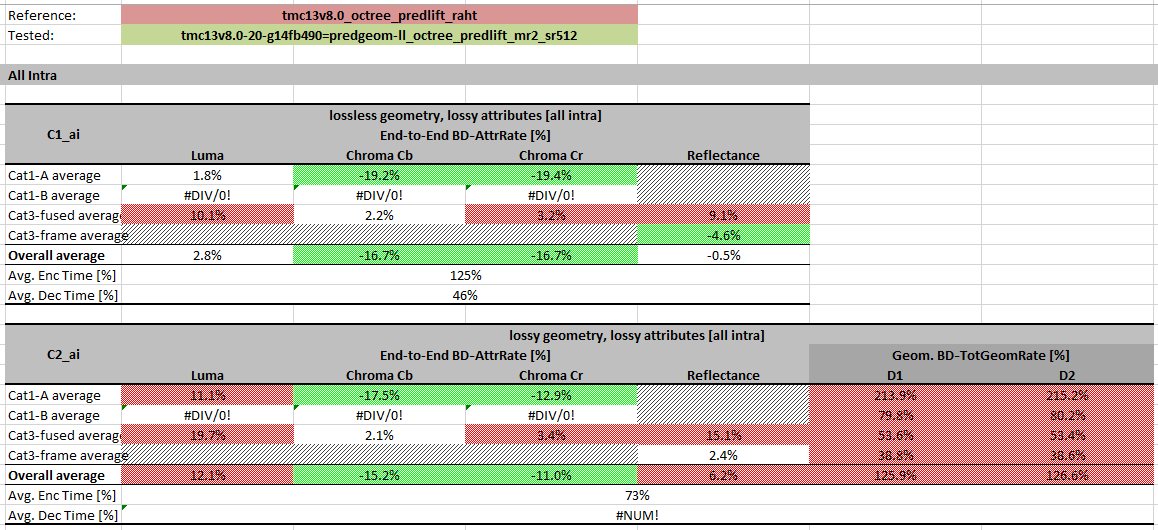


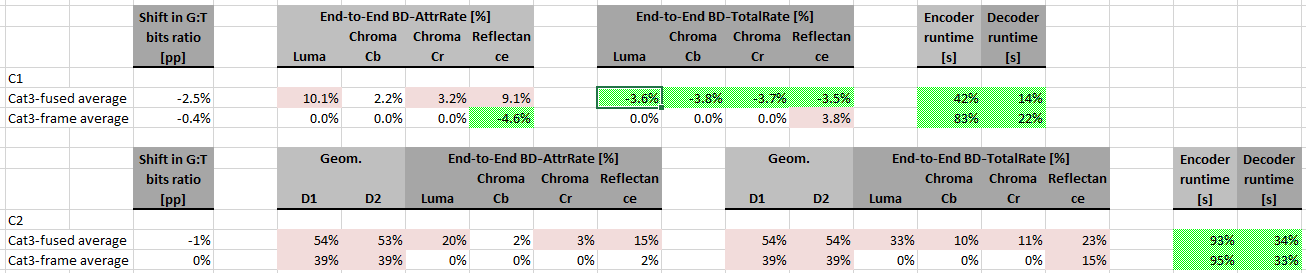
Using 512 latency points yields better rate-distortion results than using 128 points, with a potential cost of encoding time that ranges from 109% (C2) up to 124% (C1) and a negligible change in decoding time (from 94% up to 105%), as show in the table below (reference is sr128, tested is sr512).



Finally, it was confirmed that the techniques here analyzed are restrict to cat3 sequences, more specifically C1, CW and CY, and is limited by a trade-off between complexity and encoding efficiency.

Comparison with octree\_raht if also provided.





# Excel files with the cat3 Summary analysis are attached to this report (need to be crosschecked):

* cat3\_Summary\_predgeom-highlatency\_\_vs\_\_tmc13v8.0\_octree\_predlift
* cat3\_Summary\_predgeom-lowlatency-sr512\_\_vs\_\_tmc13v8.0\_octree\_predlift
* cat3\_Summary\_predgeom-lowlatency-sr512\_\_vs\_\_sr128
* cat3\_Summary\_predgeom-highlatency\_\_vs\_\_tmc13v8.0\_octree\_raht

# Recommendation

* In general, there is a strong compromise between encoding time, decoding time and rate-distortion performance. Decrease in decoding time implies increase in encoding time and decrease of rate-distortion performance.
* For cat1-A and cat1-B sequences, BD-Rates/bpip ratios are prohibitive.
* For cat3 sequences the trade-off can be considered acceptable, with the exception of C2 scenario, where rate-distortion differences are high (unless one wants to pay a RD overhead in exchange of less complexity).
* If low-latency and low-decoding time for cat3 sequences only are desired features, especially for C1, CW and CY scenarios, then the adoption of the **low-latency encoder** is recommended.
* Still considering only cat3 sequences:
  + The differences in decoding time for the low-latency and the high-latency encoders are minor.
  + Encoding time for the low-latency encoder is much lower.
  + RD performance of high-latency encoder is significantly better.
  + If high-latency with low-decoding time and better RD performance then low-latency encoder is desired for cat3 sequences, then the recommendation is to adopt **the high-latency decoder**.

# References

1. [G-PCC][New proposal] Predictive Geometry Coding, ISO/IEC JTC1/SC29 WG11 Doc. m51012, Geneva, CH, October 2019.
2. <http://mpegx.int-evry.fr/software/MPEG/PCC/CE/mpeg-pcc-tmc13/tree/mpeg128/ce13.22/predgeom-high-latency>
3. <http://mpegx.int-evry.fr/software/MPEG/PCC/CE/mpeg-pcc-tmc13/tree/mpeg128/ce13.22/predgeom-low-latency>
4. [G-PCC] CE13.22 Improvements on tree based geometry coding, ISO/IEC JTC1/SC29 WG11 Doc. N18902, Geneva, CH, October 2019.