 **ISO/IEC JTC 1/SC 29/ WG 11 N19543**

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

**Coding of moving pictures and audio**

**Convenorship: Japan (JISC)**

|  |  |
| --- | --- |
| **Document type:** | Approved WG 11 document |
| **Title:** | Description of Exploration Experiment 13.42 for G-PCC: Decoupling parsing and reconstruction for predicting transform |
| **Status:** | Approved |
| **Date of document:** | 2020-07-03 |
| **Source:** | Convenor, ISO/IEC JTC 1/SC 29/WG 11 |
| **No. of Pages:** |  |
| **Email of acting convenor** | ostermann@tnt.uni-hannover.de |
| **Committee URL:** | <http://isotc.iso.org/livelink/livelink/open/jtc1sc29> |

**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 N19543**

**Online – July 2020**

|  |  |
| --- | --- |
| **Source:** | **3DG** |
| **Title:** | **Description of Exploration Experiment 13.42 for G-PCC: Decoupling parsing and reconstruction for predicting transform** |

# Abstract

This document provides the description of the exploration experiment 13.42 on decoupling parsing and reconstruction for predicting transform.

Currently, for predicting transform, parsing of predIndex requires attribute reconstruction [1][2]. Such reconstruction dependency removal is beneficial for a codec, as also pointed out by national body comments. In this EE, methods that remove such dependency are investigated as specified in [4].

# EE13.42 Decoupling parsing and reconstruction dependency for predicting transform

## Mandates

* Study the impact on compression efficiency for the different methods decoupling parsing and reconstruction, including the proposal in [4] (on TMC13v11 [1]) for lossless and near-lossless coding configurations [3].
* Investigate alternative methods to decouple parsing and reconstruction.

Related changes to the G-PCC Specification Text [2] shall be reported.

## Participants, description of tools, and implementation notes

The following people are participating in this EE. Their specific roles are detailed in the next section. Proposal is based on the following input contribution:

1. m54701, [G-PCC] [new] An alternative method for EE13.42 for predicting transform.

Proponents and cross checkers are as follows:

| **Name** | **Company** | **E-mail address** | **Type** |
| --- | --- | --- | --- |
| Bappaditya Ray | Qualcomm Inc. | [bray@qti.qualcomm.com](mailto:bray@qti.qualcomm.com) | Proponent (m54701) |
| Toshiyasu Sugio | Panasonic | sugio.toshiyasu@jp.panasonic.com | Crosschecker |

## Information on proposed tool

### Attribute reconstruction dependency for parsing predIndex

Currently, for predicting transform, the attribute of a point in the point cloud is predicted from the attribute values of neighbouring points. Four different predictors are available: weighted predictor, and its three neighbors, and the predictor Index (referred to as predIndex) can be signalled for coding the attribute of a point. To reduce the associated signalling, the choice of four different predictors are only made available when the variation of the attribute values is higher than a threshold signaled in APS; in such a scenario predIndex is signalled. When the variation is less than the threshold, weighted predictor (weighted average in Fig. 1) is used by default and thus predIndex is not signalled.

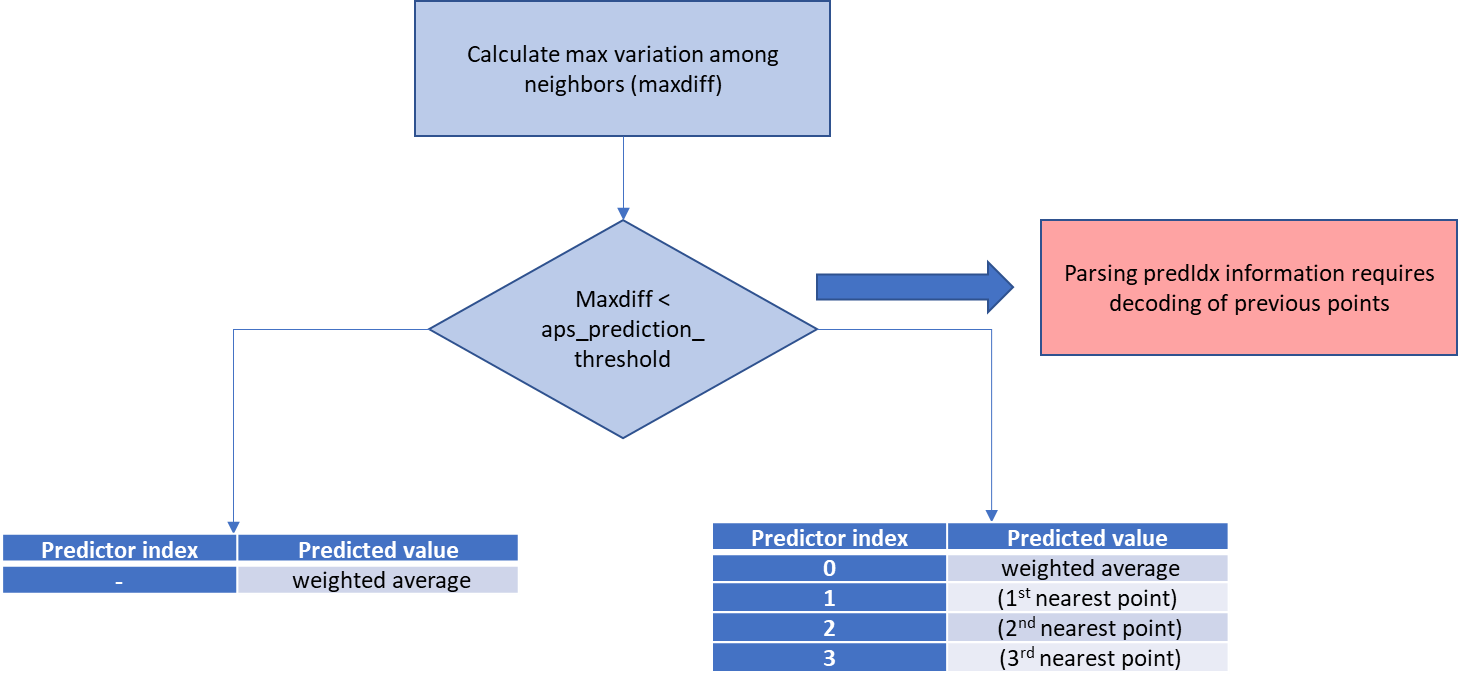


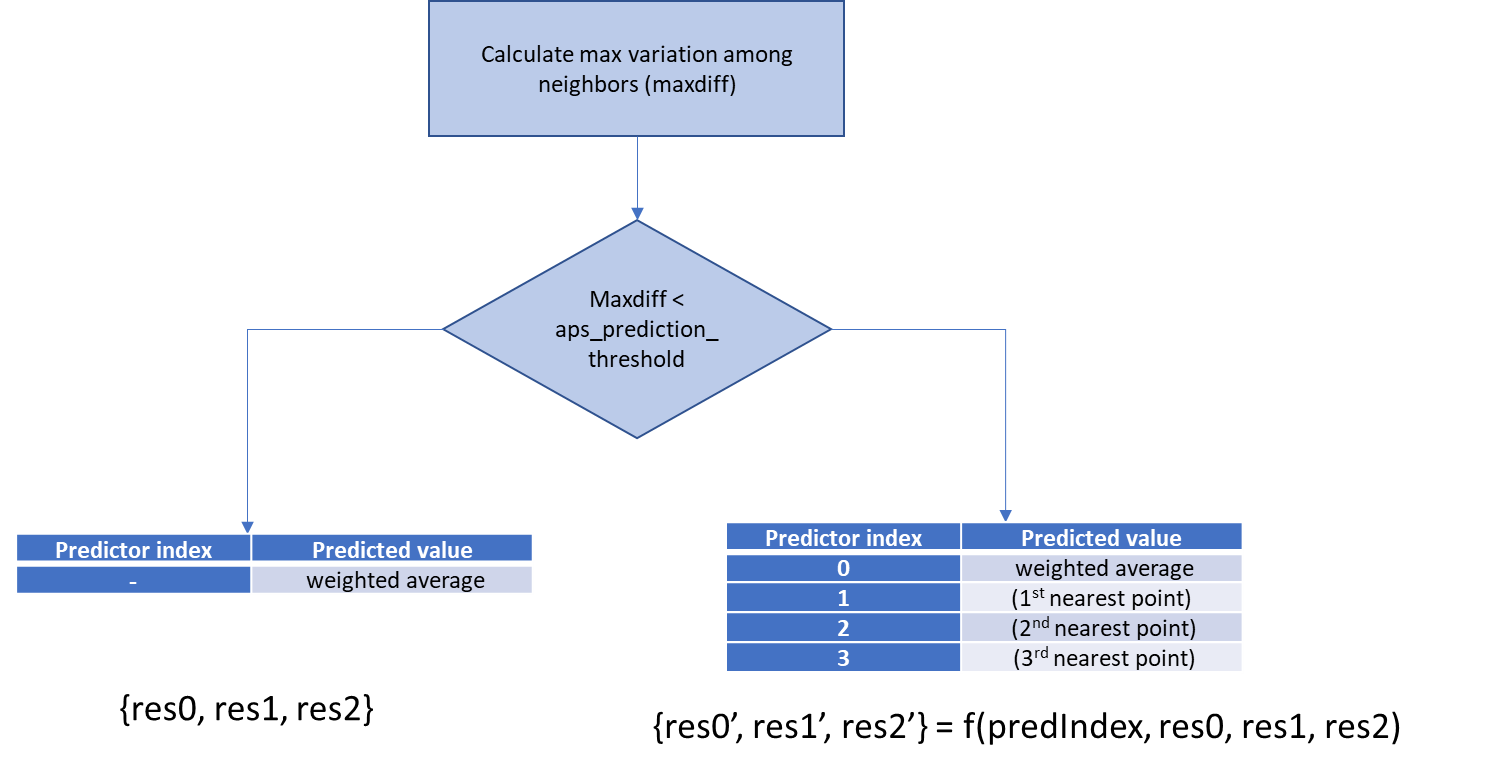
Fig. 1: Prediction process in predicting transform.

Fig. 1 illustrates the prediction strategy. It can be observed that the variation (referred to as maxDiff) is computed using reconstructed attribute values. Thus, the parsing of predIndex requires the reconstruction of the attribute.

Such kind of reconstruction dependency on the parsing is not welcome, as it does not allow to decouple the parsing and the reconstruction process. Decoupling parsing and reconstruction is beneficial for the codec. The reason being CABAC parsing is complex and highly serialized, i.e., symbols need to be parsed in order. If attribute reconstruction is needed for the parsing, that will significantly impact the latency of the overall processing chain. In video coding standards, such dependencies are always avoided. Consideration of removing such dependency was also mentioned in a national body comment recently.

### Proposed methods in m54701 [4]

It is proposed to decouple parsing and predIndex by jointly coding the predIndex and residual, and signal this resultant code as residual.



As described in the figure above, when the multiple predictor process is invoked (right branch), given the residual (res0, res1 and res2) – assuming three components of the residual for example color attribute – and predictor index, the residual and predictor index (predindex) is jointly coded using a predetermined/fixed or signalled/adaptive function “f” and the jointly coded values, i.e., res0’, res1’, and res2’ are signalled as residuals in the bitstream. On the left branch, where predictor index is not applicable, the residuals are directly signaled in the bitstream.

At the decoder side, residuals are parsed and decoded in a normal way. During the attribute reconstruction, if the process invokes multiple predictor branch (in right), inverse function of “f”, let say “invf” is used to recover residual (res0, res1, and res2) and predindex from jointly coded residual (res0’, res1’, and res2’). Note “f” should be an invertible function, i.e., invf should exist. It can also be observed that this process can be applied to both lossless and lossy coding, as the original residuals (res0, res1, and res2) can be recovered always during the process. For number of predictors N the following function can be used

(Here, we jointly code two chroma residuals (below example, for res1 and res2) can be jointly coded (considering four predictors: weighted + 3 direct predictors), and luma residual is untouched.)

f(res0, res1, res2, predindex) =

{res0

sign(res1) \* (|res1|\*2 + predindex/2),

sign(res2) \* (|res2|\*2 + predindex%2)}

= {res0’, res1’, res2’}

invf(res0’, res1’, res2’) =

{(|res1’|%2))\*2 + (|res2’| % (2)),

res0’,

sign(res1’)\*((|res1’| - (|res1’| % 2))/2),

sign(res2’)\*((|res2’| - (|res2’| % 2))/2}

= {predIndex, res0, res1, res2}

For reflectance, we use first and second neighbor as predIdx = 0 and predIdx = 1 respectively.

res0’ = f(res0, predindex) = sign(res0) \* (|res0|\*2 + predindex%2)}

invf(res0’) = {|res0’| % 2, sign(res0’)\*((|res0’| - (|res0’| % 2))/2)} = {predindex, res0}.

## Information for conducting tests

### Software

TMC13v11 shall be used for these experiments. The proposed tools shall be implemented on top of TMC13v11.

### Test configurations

Experiments are to be carried out under CTC condition with configurations: CW\_ai (lossless geometry, lossless attribute), and CY\_ai (lossless geometry, near lossless attribute), as the proposal only impacts the performance of predicting transform.

## EE 13.42 Coordinator

Bappaditya Ray ([bray@qti.qualcomm.com](mailto:bray@qti.qualcomm.com))

# Timeline:

* **2020-07-17**: Expected date for TMC13v11 release;
* **2020**-**07-17**: Expected date for release of finalized CE description;
* **2020-08-14 [TMC13v11 + 4 weeks]**:Deliver source code and results for cross-check;
* **2020-08-21 [TMC13v11 + 5 weeks]**:Deliver cross-check results;
* **2020-xx-xx**: MPEG document upload deadline (refer to updates from 3DG for document upload deadline)

# References

1. “*G-PCC Test Model v11*”, ISO/IEC JTC1/SC29/WG11 Doc. N19517, Online, July 2020
2. “*G-PCC Future Enhancements*”, ISO/IEC JTC1/SC29/WG11 MPEG2020 Doc. w19522, Online, July 2020
3. “*Common Test Conditions for PCC*” ISO/IEC JTC1/SC29/WG11 N19324, Alpbach, Austria, April 2020
4. “*[G-PCC] [new] An alternative method for EE13.42 for predicting transform*”*,* ISO/IEC JTC1/SC29/WG11 MPEG2020 Doc. M54701, Online, July 2020