

**INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC29/WG11
CODING OF MOVING PICTURES AND AUDIO**

ISO/IEC JTC1/SC29/WG11 MPEG/m53389
April 2020, Online

<i>Source:</i>	Apple Inc.	
<i>Status:</i>	Input document	
<i>Title:</i>	CE13.29 report on geometry quantisation	
<i>Author(s):</i>	David Flynn Khaled Mammou	<i>davidflynn@apple.com</i> <i>kmammou@apple.com</i>

Abstract

In-tree geometry quantisation permits the partial quantisation of position information at a particular depth of the geometry tree. The degree of quantisation is controlled by an exponential quantisation step size function.

CE13.29 is tasked with studying both the parameterisation of the step size and an extension of the quantisation feature to directly coded positions (so-called IDCM nodes). This report studies the effects of both aspects.

Overview of existing tools

Geometry quantisation in the current test model is performed as follows: At a given tree depth a QP is signalled for each node in the level. The size of these nodes is called the scaling node size (qNs). The points in each qNs sized node (encoder only) have their yet to be coded LSBs quantised according to the quantisation parameter. Octree coding proceeds as usual, although the height of each sub tree may be reduced.

Reconstruction takes the coded octree positions and scales the position bits corresponding to the scaling node size.

Quantisation step sizes

The current v9 specification [1] and software adopts the conventional exponential mapping of a quantisation parameter to a step size. A doubling of the step size equates to a change in QP of 6. For geometry, where the domain may be larger than for attribute transform coefficients, the fixed point step size precision is increased.

Quantised nodes

The current design signals a QP for internal nodes at a given level. The QP applies to all points spanned by the corresponding subtree. IDCM nodes (terminal nodes) that occur within the subtree are scaled using the given QP. IDCM nodes that occur prior to the QP signalling depth escape quantisation/scaling.

Proposed tools

Integer step sizes

The studied tool [2] proposes a different mapping that has the following properties:

- The exponential mapping is broadly maintained, but is linear between two powers of two.
- The doubling interval is a change in QP of 4.
- A fixed point representation using two fractional bits is used.
- The step sizes have (apart from some initial values) no fractional parts.

The mapping is:

$$qS = (4 + (qP \& 3)) << (qP >> 2)$$

This results in the step size series of 1, 1.25, 1.5, 1.75, 2, 2.5, 3, 3.5, 4, 5, 6, 7, 8, 10, 12, 14, 16, 20, 24, 28, 32, 40, 48, 56, 64, ... Only QPs 1, 2, 3, 5, and 7 result in non-integer step size.

Since the step sizes using this scheme are intended to be integers, it is not necessary to use rounding in the scaling function:

$$\text{scale } x := (x * qS) >> 2$$

Rather than (in line with v9):

$$\text{scale}' x := (x * qS + 2) >> 2$$

Quantisation of IDCM nodes

The studied tool [3] proposes to quantise IDCM nodes that occur before the signalling depth according to a specific QP. Since IDCM nodes cannot contain more than two distinct points, it is inappropriate to signal a per-node QP. Instead, a single QP is signalled in the GPS (or, if desired, slice header) that applies to all qualifying IDCM nodes. NB: IDCM nodes that occur within a quantised subtree do not qualify, since the subtree is already quantised.

At the 129th meeting, the interaction between the planar mode and geometry quantisation was defined. This disables the planar mode for a depth that would conflict with the quantisation. In the studied tool, the same interaction can exist. However, the same solution is not easily implemented. Instead, we elect to reduce the scaling node size when the planar mode is enabled. To illustrate, given a point, where the bit x_5 is coded in the current node and bits $x_4..x_0$ are to be coded using IDCM:

$$pos = ...x_5 | x_4 x_3 x_2 x_1 x_0$$

Ordinarily, quantisation would apply to bits $x_4..x_0$ (the quantisation node size). However, if planar mode is enabled, x_4 may be inferred by the planar mode bits. In this case, the scaling node size is reduced by one, and bits $x_3..x_0$ are quantised.

Similarly, at the 129th meeting, it was decided that QPs should be limited such as to not overquantise the node. This has been applied in the same manner to the IDCM QP by clipping the effective QP at each depth of the tree. For example:

$$\begin{aligned} pos &= ...x_6 | x_5 x_4 x_3 x_2 x_1 x_0, & \text{maxqp} &= 39 \\ pos &= ...x_6 x_5 | x_4 x_3 x_2 x_1 x_0, & \text{maxqp} &= 33 \end{aligned}$$

Furthermore, the clipping is based upon the minimum child node size such that no dimension is over-quantised and that all dimensions are equally quantised. In conjunction with some QTBT configurations, the maximum QP can become quite limited.

NB: the quantisation step sizes used to study this tool are the same as v9. The technique applies equally to the integer step sizes also studied in the CE.

Evaluation methodology

To determine the performance of the quantisation schemes, the studies in this CE evaluate a modified lossy geometry condition. The six usual rate points are replaced by a QP sweep that covers all valid QPs within a given scaling node size. Each node uses the same QP.

Evaluations are performed using scaling node sizes of 2^n , $n \in \{2, 5, 7, 11\}$. In order to permit configuring the scaling node size rather than the tree depth, an new (non-normative) configuration parameter is provided in the anchor.

Results are presented in the form of graphs showing the bitrate vs the D1 geometry PSNR in order of sequential QPs.

Due to the large number of test points, the CE has evaluated only the following test sequences: facade_00064_vox11, facade_00064_vox14, facade_00064_vox20, ford_01_q1mm, longdress_vox10_1300, loot_viewdep_vox12, overpass_q1mm, qnx-junction-approach, queen_0200, stanford_area_2_vox16, and ulb_unicorn_vox20.

To evaluate IDCM, two tests are conducted:

- First that the introduction of IDCM quantisation does not affect the existing codec behaviour when used with a QP=4 (no quantisation).
- Second to examine the effect of IDCM quantisation when the IDCM QP is the same as the node QP.

Since the quantisation of IDCM nodes requires there to be IDCM nodes for there to be any observable effect, planarModeIdcmUse is set to 127 compared to certain CTC conditions where it is set to 0. This change is also applied to the anchor.

Results

Integer step sizes

Figure 1 compares the behaviour of the existing draft quantiser with the integer step size proposal. The orange curve shows the v9.0 CE anchor results. The solid blue curve shows the studied scheme (without rounding). A dashed blue curve (if present) shows the studied scheme with rounding in reconstruction (scale' above).

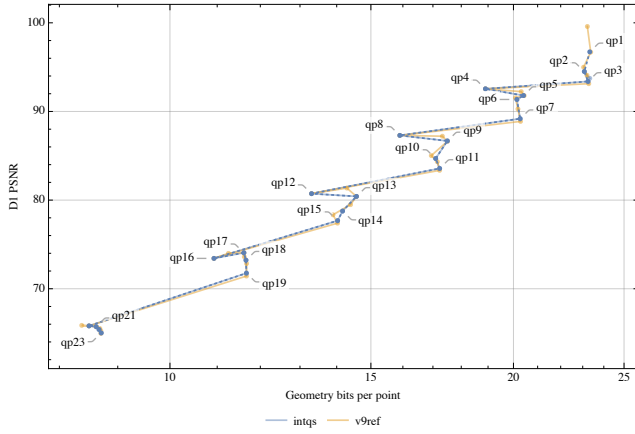
To subjectively assess the effect of integer step sizes, the sequences longdress_vox10_1300 and loot_viewdep_vox12 should be viewed by interested experts.

To study the question of an appropriate QP granularity (number of QPs per step size doubling), Figure 2 shows the results presented in Figure 1 after having been filtered to remove QPs divisible by 2 and 4 (ie, power of two step sizes only).

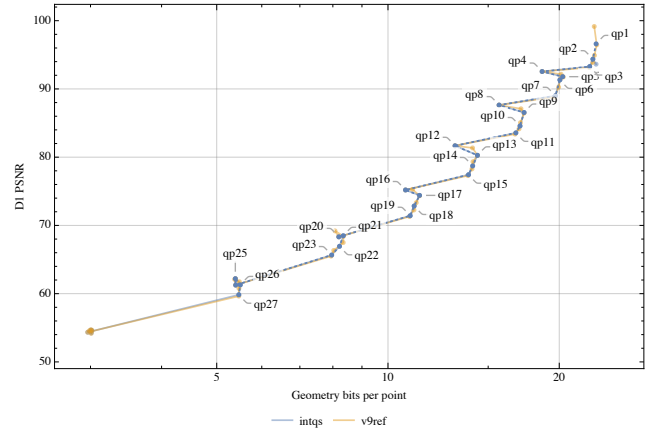
Quantisation of IDCM nodes

To verify that the implementation of idcmqp does not adversely affect the existing quantisation scheme, a qpweep is conducted on both the idcmqp anchor and the idcmqp implementation with idcmqp equal to 4. The results of each test point are compared. Figure 3 summarises the behaviour with the scaling node size of 2^5 . There are no changes in measured distortion, nor of the geometry data lengths. The differences in enc.ext.bits are accounted for by the introduction of signalling the idcmqp in the high level syntax. More detailed reports for all studied node sizes are included with this document¹.

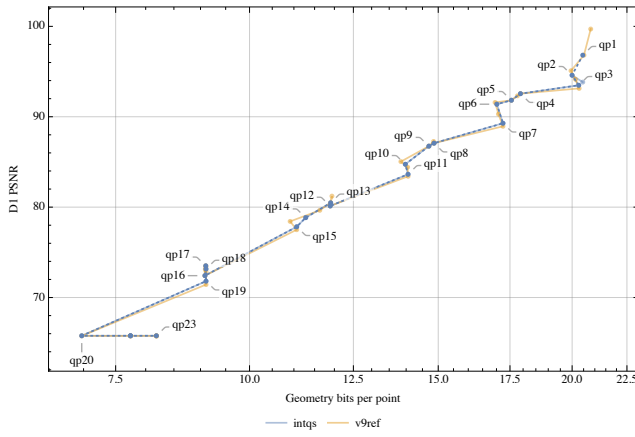
¹report_ns=*_idcmqp=4__vs__v9+idcm=127.txt



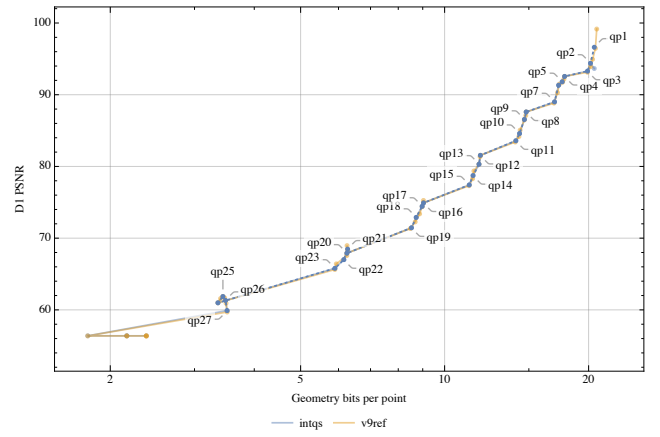
(a) ford_01_q1mm, node size 2^5



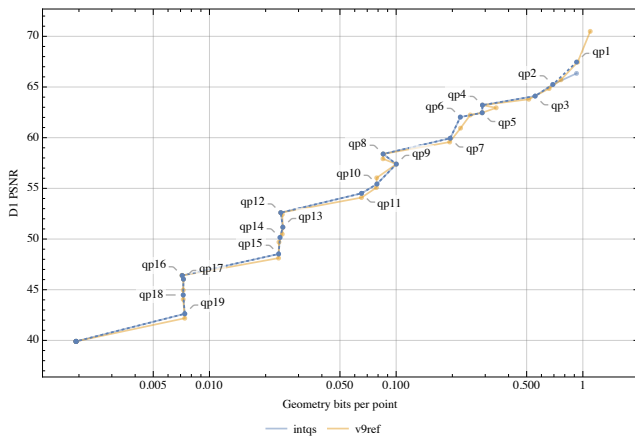
(b) ford_01_q1mm, node size 2^7



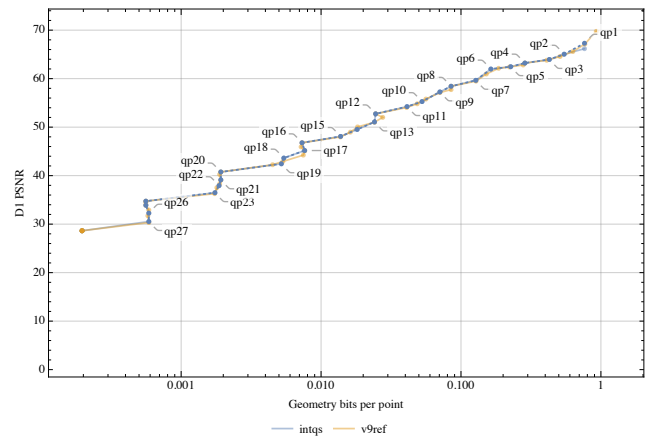
(c) overpass_q1mm, node size 2^5



(d) overpass_q1mm, node size 2^7

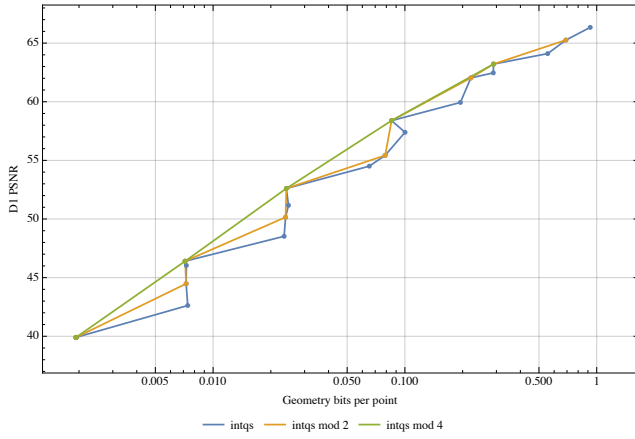


(e) longdress_vox10_1300, node size 2^5

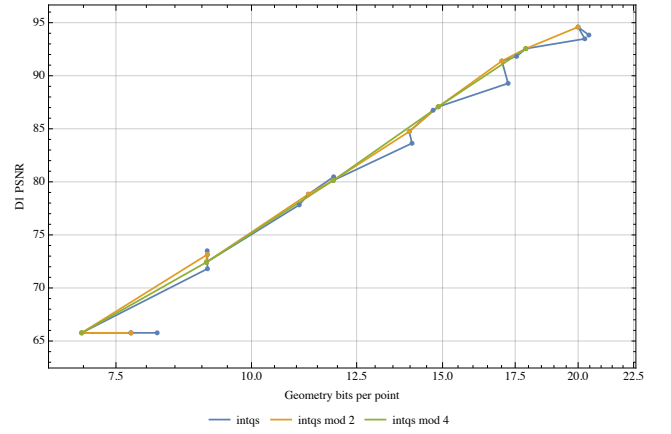


(f) longdress_vox10_1300, node size 2^7

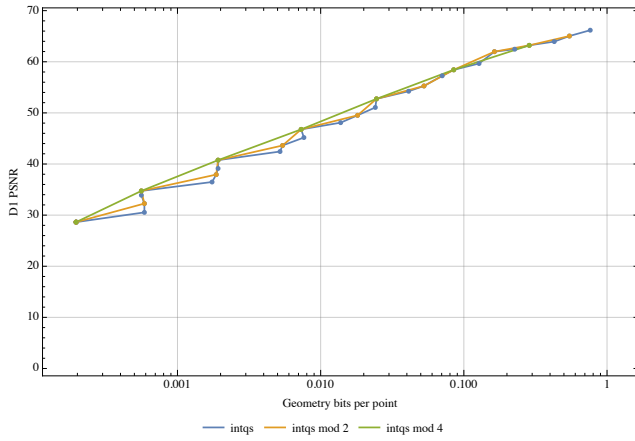
Figure 1 – Rate-distortion plots of quantisation parameter sweep using draft v9 step sizes and proposed integer step size (marked) at quantisation node sizes of 2^5 and 2^7



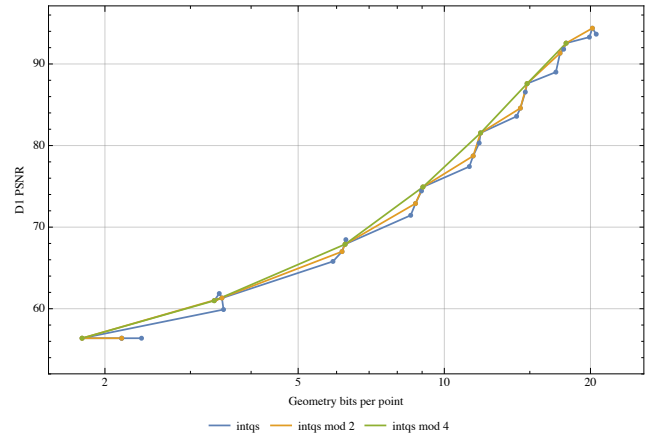
(a) longdress_vox10_1300, node size 2^5



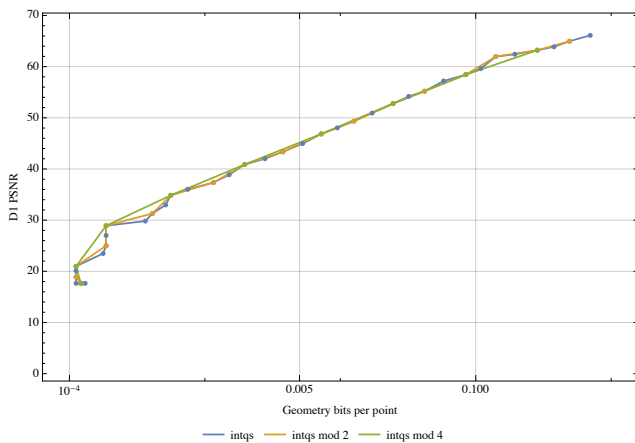
(b) overpass_q1mm, node size 2^5



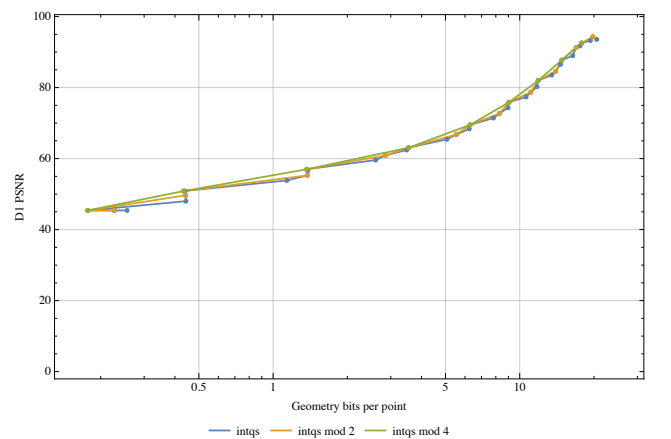
(c) longdress_vox10_1300, node size 2^7



(d) overpass_q1mm, node size 2^7



(e) longdress_vox10_1300, node size 2^9



(f) overpass_q1mm, node size 2^9

Figure 2 – Rate-distortion plots of quantisation parameter sweep using proposed QP mapping for various scaling node sizes, along with coarser alternatives QP modulo 2, and QP modulo 4

Figure 3 – Summary of successful idcmqp = 4 verification for node size = 2^5

```
Summary results:
  omissions: 0
  pass: 28
  failed: 357
    - of which due to partial data: 0
Detail: (pass / fail / missing data)
  enc.ext.bits: 28 / 357 / 0
  enc.bits.geometry: 385 / 0 / 0
  enc.bits.reflectance: 105 / 0 / 0
  enc.bits.colour: 315 / 0 / 0
  dec.numpoints: 385 / 0 / 0
  dec.d1-psnr: 385 / 0 / 0
  dec.d2-psnr: 385 / 0 / 0
  dec.y-psnr: 315 / 0 / 0
  dec.cb-psnr: 315 / 0 / 0
  dec.cr-psnr: 315 / 0 / 0
  dec.reflectance-psnr: 105 / 0 / 0
```

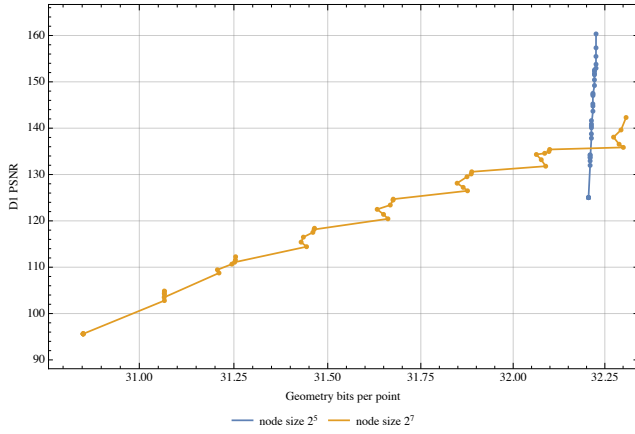
IDCM quantisation control

Figure 4 illustrates the behaviour of IDCM quantisation. In order to avoid any effects of QTBT limiting the range of the IDCM QP (which can be exacerbated by the Angular QTBT twwak), the results presented disable QTBT. Additional results are available that examine the CTC QTBT configuration.

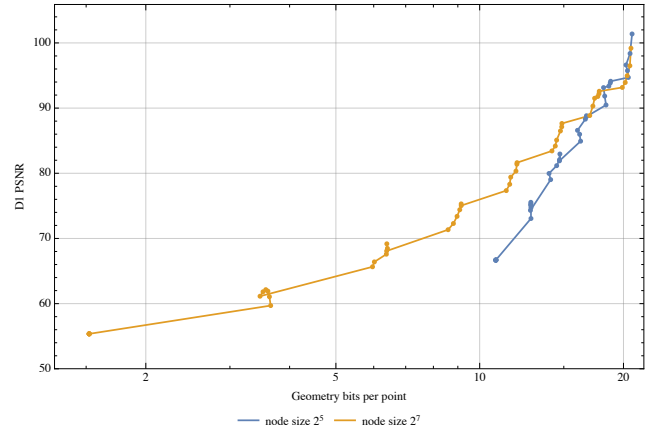
The first row, figs. 4a and 4b, illustrates, using the current test model, the effect observed in the current draft where subtrees that are terminated early by IDCM are not eligible for quantisation. The two curves correspond to different scaling node sizes of the anchor. In the extreme case of ulb_unicorn_vox20, almost all nodes are terminated by IDCM before reaching the scaling node size of 2^5 .

The middle row, figs. 4c and 4d, shows the effect when the IDCM QP is enabled. Here, the comparison is between the anchor and idcmqp. For this test, the IDCM QP is the same as the slice QP. Early terminated IDCM nodes are therefore able to be quantised similarly to non-early terminated nodes.

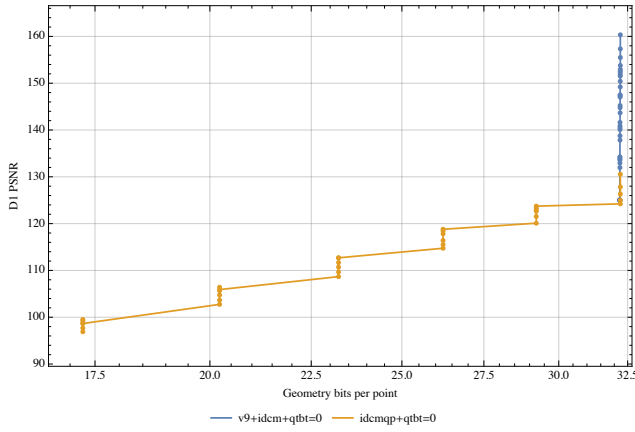
Finally, the last row, figs. 4e and 4f, shows the same comparison as the first row, except the software used for both curves has idcmqp enabled. Again, the difference is just the scaling node size. The disparity between the codec behaviour at different scaling node sizes is eliminated or significantly reduced.



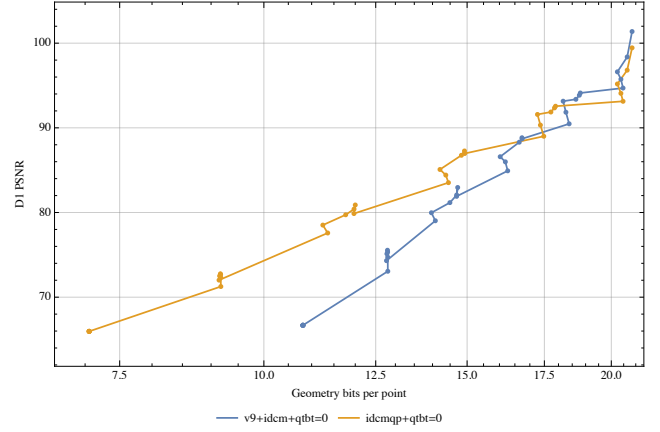
(a) ulb_unicorn_vox20, v9+idcm, node sizes 2^5 , 2^7



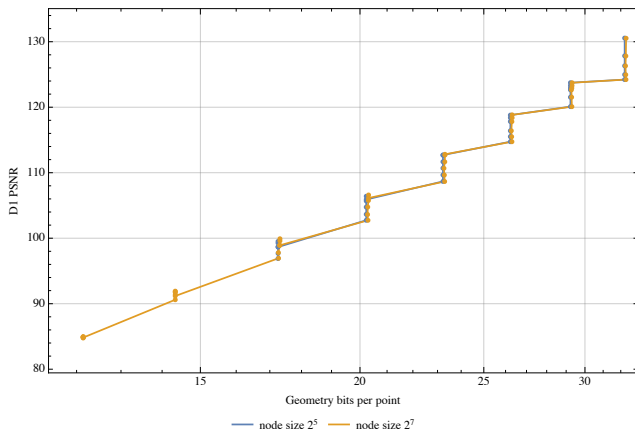
(b) overpass_q1mm, v9+idcm, node sizes 2^5 , 2^7



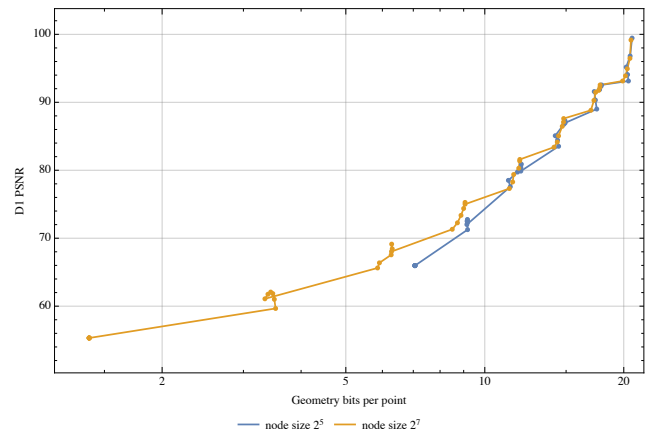
(c) ulb_unicorn_vox20, idcmqp, node size 2^5



(d) overpass_q1mm, idcmqp, node size 2^5



(e) ulb_unicorn_vox20, idcmqp, node sizes 2^5 , 2^7



(f) overpass_q1mm, idcmqp, node sizes 2^5 , 2^7

Figure 4 – Rate-distortion plots illustrating the behaviour of geometry quantisation in the presence of IDCM nodes with and without idcmqp

Repository and configuration

Modifications to TMC13 v9 that provide the anchor configuration and the studied implementation are available in the following branches of the CE repository:

- [mpeg129/ce13.29/geomquant-qpsweep](#), the anchor
- [mpeg129/ce13.29/geomquant-intqs-qpsweep](#), m52522
- [mpeg129/ce13.29/geomquant-idcmqp-qpsweep](#), m52523

NB: the anchor branch has been updated with additional configuration files and bug fixes necessary for this evaluation. This does not affect the anchor generation used for the integer step size evaluation.

The following configuration snippets are provided:

- `cfg-geomquant-size=5.yaml`: sweep of QPs for $32 \times 32 \times 32$ nodes
- `cfg-geomquant-size=7.yaml`: sweep of QPs for $128 \times 128 \times 128$ nodes
- `cfg-geomquant-size=9.yaml`: sweep of QPs for $512 \times 512 \times 512$ nodes
- `cfg-geomquant-size=11.yaml`: sweep of QPs for $2048 \times 2048 \times 2048$ nodes
- `cfg-idcm=127.yaml`: sets `planarModeIdcmUse=127`
- `cfg-qtbt=0.yaml`: disables qtbt

NB: some of these snippets are different in the respective branches due to the different QP mapping.

idcmqp branch only:

- `cfg-idcmquant-sweep=5.yaml`: IDCM QPs matching `cfg-geomquant-size=5.yaml`
- `cfg-idcmquant-sweep=7.yaml`: IDCM QPs matching `cfg-geomquant-size=7.yaml`
- `cfg-idcmquant-sweep=9.yaml`: IDCM QPs matching `cfg-geomquant-size=9.yaml`
- `cfg-idcmquant-sweep=11.yaml`: IDCM QPs matching `cfg-geomquant-size=11.yaml`

NB: the QP sweep uses the `lossy-geom-lossy-attrs` configuration, but replaces the CTC rate points with the QP sweep. This will generate a different set of configuration files to normal. For attributes, a QP corresponding to r04 of the CTC is used.

Configuration files may be generated using the `gen-cfg.sh` script. For example:

```
$ cd cfg
$ ../scripts/gen-cfg.sh -- cfg-geomquant-size=5.yaml
```

The anchor and results of integer step sizes are generated using: `cfg-geomquant-size=${ns}.yaml` and `cfg-idcm=127.yaml`

Results relating to idcmqp are generated using:

- `cfg-geomquant-size=${ns}.yaml`, `cfg-idcm=127.yaml`, and `cfg-qtbt=0.yaml` (anchor).
- `cfg-geomquant-size=${ns}.yaml`, `cfg-idcmquant-sweep=${ns}.yaml`, and `cfg-qtbt=0.yaml` (idcmqp).

NB: `cfg-idcm=127.yaml` is included in `cfg-idcmquant-sweep=${ns}.yaml`

References

- [1] 3DG, “Text of ISO/IEC 23090-9 DIS Geometry-based PCC,” ISO/IEC JTC1/SC29/WG11, 129th meeting, Brussels, Tech. Rep. w19088, Jan. 2020.
- [2] D. Flynn and K. Mammou, “G-PCC: Integer step sizes for in-tree geometry quantisation,” ISO/IEC JTC1/SC29/WG11, 129th meeting, Brussels, Tech. Rep. m52522, Jan. 2020.
- [3] —, “G-PCC: An IDCM specific QP for in-tree geometry quantisation,” ISO/IEC JTC1/SC29/WG11, 129th meeting, Brussels, Tech. Rep. m52523, Jan. 2020.