ISO/IEC JTC1/SC29/WG1 **N90053**

90th Meeting – Online – 18-22 January 2021

INTERNATIONAL ORGANISATION FOR STANDARDISATION
ORGANISATION INTERNATIONALE DE NORMALISATION

ISO/IEC JTC 1/SC 29/WG1
(ITU-T SG16)

Coding of Still Pictures

JBIG	**JPEG**
Joint Bi-level Image Experts Group	Joint Photographic Experts Group

TITLE:	JPEG Pleno Point Cloud Coding Common Test Conditions v3.5
SOURCE:	Stuart Perry (Editor)
PROJECT:	JPEG Pleno
STATUS:	Approved
REQUESTED ACTION:	For Discussion
DISTRIBUTION:	Public

Contact:
ISO/IEC JTC 1/SC 29/WG1 Convener – Prof. Touradj Ebrahimi
EPFL/STI/IEL/GR-EB, Station 11, CH-1015 Lausanne, Switzerland
Tel: +41 21 693 2606, Fax: +41 21 693 7600, E-mail: Touradj.Ebrahimi@epfl.ch

Editorial Comments

This is a living document that goes through iterations. Proposals for revisions of the text can be delivered to the editor Stuart Perry, by downloading this document, editing it using track changes and sending it to Stuart.Perry@uts.edu.au.

If you have interest in JPEG Pleno Point Cloud, please subscribe to the email reflector, via the following link: http://jpeg-pointcloud-list.jpeg.org

Index

ISO/IEC JTC1/SC29/WG1 N90053

90th Meeting – Online – 18-22 January 2021

JPEG Pleno Point Cloud Coding Common Test Conditions v3.5

January 22nd, 2021

1 Scope

This document describes the Common Test Conditions for the JPEG Pleno Point Cloud Coding experiments.

2 Test Materials

This section describes the test material selected for JPEG Pleno Point Cloud Coding core and exploration experiments.

The JPEG Pleno Point Cloud Coding Data Test Set is diverse in terms of

- Content type
- Number of points
- Spatial density of points
- Additional attributes such as colour

In the following, the JPEG Pleno Point Cloud Coding Test Set will be organized according to the type of content.

- Human beings
- Small inanimate objects

2.1 Human Beings

This section presents the selected human beings point clouds, organized as full bodies and upper bodies.

2.1.1 Full Bodies

This dataset has been provided by 8i Labs [8i17]. There are two voxelised sequences in the dataset, known as *longdress* and *soldier*, pictured in Figure 1. For the JPEG Pleno Point Cloud Coding test set, we have selected a single frame from each sequence as indicated in Table 1. In each sequence, the full body of a human subject is captured by 42 RGB cameras configured in 14 clusters (each cluster acting as a logical RGBD camera), at 30 fps, over a 10 s period. One spatial resolution is provided for each sequence: a cube of 1024x1024x1024 voxels, this results in a geometry precision of 10 bits. For each sequence, the cube is scaled so that it is the smallest bounding cube that contains every frame in the entire sequence. Since the height of the subject is typically the longest dimension, for a subject 1.8 m high, a voxel with a geometric precision of 10 bits would be approximately 1.8 m / 1024 voxels ≈

1.75 mm per voxel on a side. As these are dynamic point clouds, only a single frame specified in Table 1 will be used. Figure 1 shows renderings of the Full Bodies point clouds.

Table 1 – Full Bodies point cloud properties.

Point cloud name	Frame number	Point cloud file name	Number of points	Geometry precision
longdress	1300	longdress_vox10_1300.ply	857,966	10 bit
soldier	0690	soldier_vox10_0690.ply	1,089,091	10 bit

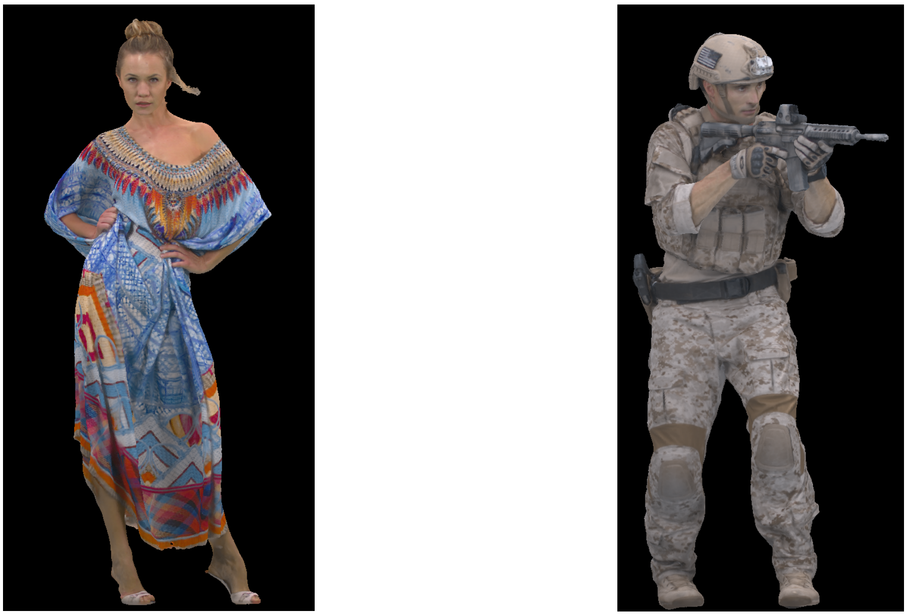

Figure 1 – Renderings of the Full Bodies point clouds. On the left is *longdress* and *soldier* is on the right.

This data set can be found at: https://jpeg.org/plenodb/pc/8ilabs/.

2.1.2 Upper Bodies

The dynamic voxelized point cloud sequences in this dataset are known as the Microsoft Voxelized Upper Bodies (MVUB) [MVUB16]. The JPEG Committee makes use of two subjects in the dataset, known as *phil9*, and *ricardo10* pictured in Figure 2. The heads and torsos of these subjects are captured by four frontal RGBD cameras, at 30 fps, over a 7-10s period for each. For the JPEG Pleno Point Cloud Coding test set, we have selected a single frame from each sequence as indicated in Table 2. Two spatial resolutions are provided for each sequence: a cube of 512x512x512 voxels and a cube of 1024x1024x1024 voxels, indicating a geometry precision of 9 and 10 bits respectively. The JPEG Committee has chosen the 9 bit data for the *phil9* point cloud and the 10 bit data for the *ricardo10* point cloud. Voxels with a precision of 9 bits correspond to volumes of approximately 1.5 mm^3, while voxels with a precision of 10 bits correspond to volumes of approximately 0.75 mm^3. As these are dynamic point clouds, only a single frame from each sequence as specified in Table 2 will be used.

Table 2 - Upper Bodies point cloud properties.

Point cloud name	Frame number	Point cloud file name	Number of points	Geometry precision
ricardo10	82	ricardo10_frame0082.ply	1,414,040	10 bit
phil9	139	phil9_frame0139.ply	356,258	9 bit

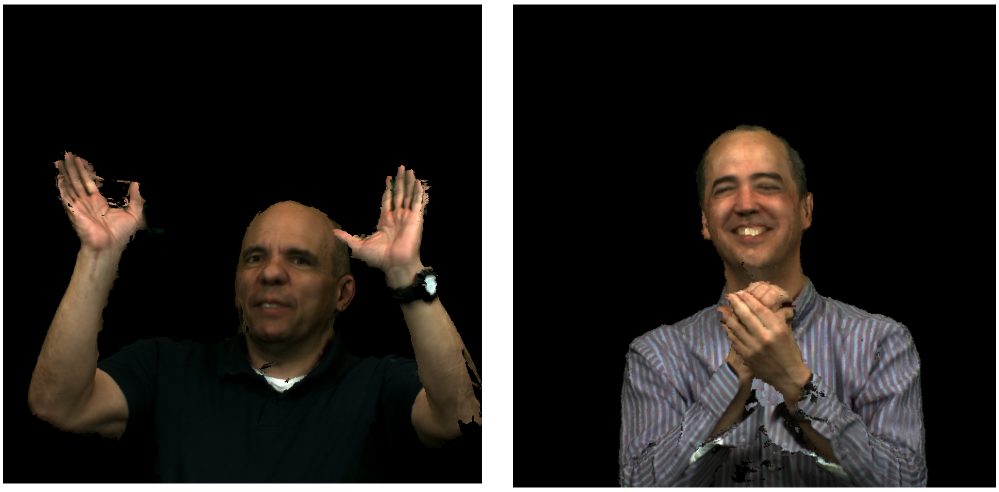

Figure 2 - Renderings of the Upper Bodies point clouds. On the left is *ricardo10* and *phil9* is on the right.

This data set can be found at: https://jpeg.org/plenodb/pc/microsoft/.

2.2 Inanimate Objects

University of San Paulo, Brazil supplied a set of point clouds to JPEG related to cultural heritage. These point clouds have 24 bit RGB colour information associated with each point. The *romanoillamp* and *bumbameuboi* point clouds have been selected for the JPEG Pleno Point Cloud Coding test set to represent cultural heritage applications. Voxelised versions of these point clouds are available in the JPEG Pleno Point Cloud Coding test set. The voxelised version of the *romanoillamp* has been voxelised to a geometry precision of 10 bits. In addition, École Polytechnique Fédérale de Lausanne (EPFL) donated the *guanyin* and *rhetorician* point clouds, which are part of the PointXR dataset [Alexiou, 2020], also representing cultural heritage application. Both of these point clouds have been voxelised to a geometry precision of 10 bits. This set of point clouds are shown in Figure 3 and the properties of these point clouds are described in Table 3.

Table 3 - Small inanimate objects point cloud properties.

Point cloud name	Frame number	Point cloud file name	Number of points	Geometry precision
romanoillamp	-	romanoillamp_Transform_Denoise_vox10.ply	638,071	10 bit
bumbameuboi	-	bumbameuboi_Rescale_Denoise_vox10.ply	113,160	10 bit
guanyin	-	guanyin_vox10.ply	2,286,975	10 bit
rhetorician	-	rhetorician_vox10.ply	1,732,600	10 bit

(a)

(b)

Figure 3 - Rendering of the (a) *romanoillamp,* (b) *bumbameuboi,* (c) *guanyin* and (d) *rhetorician* point clouds.

The small inanimate objects point clouds provided by the University of San Paulo may be found at: http://uspaulopc.di.ubi.pt/.

The small inanimate objects point clouds provided by École Polytechnique Fédérale de Lausanne (EPFL) may be found at: https://www.epfl.ch/labs/mmspg/downloads/pointxr/.

3 Subjective Quality Evaluation

For subjective evaluation, it is important to allow for consistent rendering of the point cloud data. For subjective evaluation, point clouds will be rendered and displayed in the following fashion:

3.1 Rendering

- Point clouds will be rendered to a video using CloudCompare software [CloudCompare19] with the following settings:
  - Default parameters except for point size.
  - Point size needs to be determined for each sequence, by experimentation to render as far as possible the perception of a watertight surface.
- Point clouds will be rendered against a black background without surface reconstruction, in other words as individual points.
- Point clouds should be rendered for, and viewed on, 2D monitors with resolution of at least 1920x1080 and ideally of resolution 3840x2160 and a colour gamut of sRGB or wider. Point clouds should be rendered such that the intrinsic resolution of the point cloud matches the display resolution. This requires as far as possible that point

density of the rendered point cloud be such that no more than one point should occupy a single pixel of the displayed video. This may require the display of cropped sections of the point cloud with suitably adjusted view paths.

  - Customization for resolution of the frames is by adjusting the edge of 3D view window to the desired resolution.
  - In order to get full resolution of 2D monitors, press F11 (full screen 3D view) before choosing view and rendering.

3.2 Video Production

- To create the video, the camera will be rotated around the object to create a set of video frames that allows for viewing of the entire object. The exact path used for the video sequence will vary depending on the content type and is only known to the Chair and Co-chair of the Ad Hoc Group on JPEG Pleno - Point Cloud, Stuart Perry (University of Technology Sydney, Australia) and Luis Cruz (University of Coimbra, Portugal). The view paths are not revealed to proponents to avoid proposals being explicitly tailored according to content view paths.
- Videos will be created such that each video is of 12 seconds duration at a rate of at least 30 fps. As far as possible the video view paths should be adjusted such that the duration of the video is approximately 12 seconds, while the degree of apparent motion between frames is the same for all stimuli videos.
- The video frames will be then visually losslessly compressed (i.e., constant rate factor equal to 17) with the HEVC encoder (using FFmpeg software), producing an animated video of 30 fps with a total duration of 12 seconds.

3.3 Viewing Conditions

- Viewing conditions should follow ITU-R Recommendation BT.500.13 [BT50013]. MPV [MPV19] video player will be used for displaying the videos. Displays used in the subjective testing should have anti-aliasing disabled.

3.4 Training Before Subjective Evaluation

- Prior to subjective evaluation there will be a training period to acclimatize participants to the task to be performed. This training period involves showing participants video sequences similar to the ones used in the test, but not the point clouds used in the subsequent test sequences. The *phil9* point cloud described in Section 2.1.2 will be used for this purpose.
- Participants are requested to score the perceived quality of the rendered point cloud in relation to the uncompressed.

3.5 Subjective Evaluation Protocol

- The DSIS simultaneous test method will be used with a 5-level impairment scale, including a hidden reference for sanity checking. Both the reference and the degraded stimuli will be simultaneously shown to the observer, side-by-side, and every subject asked to rate the visual quality of the processed with respect to the reference stimulus. To avoid bias, in half observer presentations, the reference will be placed on the right and the degraded content on the left side of the screen, and vice-versa for the rest of the evaluations.

3.6 Analysis of Results

- Outlier detection algorithm based on ITU-R Recommendation BT.500-13 [BT50013] will be applied to the collected scores, and the ratings of the identified outliers will be discarded. The scores will then be averaged to compute mean opinion scores (MOS) and 95% Confidence Intervals (CIs) assuming a Student's t-distribution.

4 Rate and Rate-Distortion Evaluation

Rate metrics are concerned with characterising the change in bitrate under various conditions.

4.1 Bit Rate

The bit rates specified in the test conditions detailed in this document and reported for the experiments with the various codecs should account for the total number of bits, *NTotBits*, necessary for generating the encoded file (or files) out of which the decoder can reconstruct a lossy or lossless version of the entire input point cloud.

The main rate metric is the number of bits per point (bpp) defined as:

$$Bits\ per\ point\ =\ NTotBits/NTotPoints$$

where *NTotBits* is the number of bits for the compressed representation of the point cloud and *NTotPoints* is the number of points in the input point cloud. In addition to the total bits per point, the bits per point used to code geometry and the bits per point used to code attributes should also be reported.

4.2 Scalability

Scalability capabilities shall be assessed by encoding the point cloud using scalable and non-scalable codecs at a variety of rate points. The relative compression efficiency will be assessed using Bjontegaard delta metrics (Rate and PSNR) between scalable and non-scalable codecs. The JPEG Committee will compute RD curves using the PSNRD1, PSNRD2, plane to plane and point to point attribute measure metrics described below to allow different scalability requirements to be tested.

Scalable codecs will have rate and distortion values obtained when decoding at each level of scalability. If a scalable codec has a non-scalable mode, then rate and distortion values will also be obtained for the codec in the non-scalable mode.

Reports on scalability should report on bits required to achieve each specific scalability level.

4.2.1 Random Access

The *RoI Random Access* measure shall be measured, for a given quality, by the maximum number of bits required to access a specific Region of Interest (RoI) of the point cloud divided by the total number of coded bits for the point cloud.

$$RoI\ Random\ Access\ = \frac{Total\ amount\ of\ bits\ that\ have\ to\ be\ decoded\ to\ access\ an\ RoI}{Total\ amount\ of\ encoded\ bits\ to\ decode\ the\ full\ point\ cloud}$$

A set of predetermined RoIs will be defined by the JPEG Committee. A codec must provide the entire RoI, but may provide more points. The result should be reported for the specific RoI type that gives the worst possible result for the *RoI Random Access*.

5 Quality Metrics

Quality Metrics are metrics that measure the similarity of the decoded point cloud to a reference point cloud using an objective measure. Ideally the quality measures using should correlate well with subjective testing results.

The quality metrics considered for this activity are:

- Point to Point Geometry D1 Metric
- Point to Plane Geometry D2 Metric
- Plane to Plane Angular Similarity Metric
- Point to Point Attribute Measure

5.1 Point to Point Geometry PSNRD1 Metric

The point to point geometry PSNRD1 metric is based on the geometric distance of associated points between a stimulus under evaluation, **B**, and a reference point cloud, **A**. In particular, for every point of the content under evaluation, a point that belongs to the reference point cloud is identified, through the nearest neighbor algorithm. Then, an individual error is computed based on the Euclidean distance.

This error value is associated to every point, indicating the displacement of the distorted point from the reference position. The error values for each point are then summed to create a final measure. In accordance with the description given in [wg11/n18665], the point to point geometry PSNRD1 metric is computed as follows:

Using the notation of [wg11/n18665], we first determine an error vector $E(i,j)$ denoting the difference vector between the identified point a_j in reference point cloud **A** to the corresponding point b_i (identified by the nearest neighbour algorithm) in point cloud **B**. The length of the error vector is the point to point error, i.e.,

$$e_{B,A}^{D1}(i) = \left| |E(i,j)| \right|_2^2 \qquad (1)$$

The point to point error (D1) for the whole point cloud is then defined as the arithmetic mean, $e_{B,A}^{D1}$, of the errors $e_{B,A}^{D1}(i)$.

This error is expressed as the symmetric PSNRD1 using the following formula:

$$PSNRD1 = 10 \log_{10} \frac{3*peak^2}{max(e_{B,A}^{D1}, e_{A,B}^{D1})} \qquad (2)$$

where *peak* is the geometric resolution of the model (i.e. if voxel bit depth = 10, it is 1024) and $e_{A,B}^{D1}$ is the point to point error when the roles of **A** and **B** are swapped in Equation 1. The denominator of Equation 2 ensures that the resultant PSNR is symmetric and invariant to which of the point clouds compared is considered the reference.

The PSNRD1 metric will be computed by the JPEG Committee using the software supplied by WG07 “MPEG 3D Graphics coding” at:

http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric/tree/master.

The tag "release-v0.13.5" indicates the current released version of the software, which will be used for this work.

5.2 Point to Plane Geometry PSNRD2 Metric

The point to plane geometry PSNRD2 metric is based on the geometric distance of associated points between the content under evaluation, **B**, and the local planes fitted to the reference point cloud, **A**. In particular, for every point of the content under evaluation, b_i, a point that belongs to the reference point cloud, a_j, is identified through the nearest neighbor algorithm and a plane is fitted to the region centred on the identified point on the reference cloud. The normal of this plane is denoted N_j and is computed using quadric fitting in CloudCompare [CloudCompare19] including points within a radius of 20. Where a point cloud has pre-computed available information for the normal at a_j, this information may be used in place of N_j. The distance vector $E(i,j)$ between b_i and a_j is computed and projected via the dot product onto the normal vector, N_j to compute the point to plane error [wg11/n18665]:

$$e_{B,A}^{D2}(i) =< E(i,j), N_j >^2 \qquad (3)$$

Where $< x, y >$ denotes the dot product between vectors x and y.

The point to plane error (D2) for the whole point cloud is then defined as the arithmetic mean, $e_{B,A}^{D2}$ of the errors $e_{B,A}^{D2}(i)$.

The point to plane geometry PSNRD2 metric is expressed as the symmetric PSNR in the same manner as PSNRD1 (Equation 2).

The PSNRD2 metric will be computed by the JPEG Committee using the software supplied by WG07 at:

http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric/tree/master.

The tag "release-v0.13.5" indicates the current released version of the software, which will be used for this work.

The following command line [wg11/n18665]:

```
  ./pc_error   --fileA=pointcloudOrg.ply   --fileB=pointcloudDec.ply   --
inputNorm=normalOrg.ply
```

should be used in the case the normal information for the reference point cloud is available in the file normalOrg.ply.

5.3 Plane to Plane Angular Similarity Metric

The plane to plane metric [Alexiou, 2018] is based on the angular similarity of tangent planes that correspond to associated points between a stimulus under evaluation, **B**, and a reference point cloud, **A**. In particular, for every point of the stimulus under evaluation, b_i, a point that belongs to the reference point cloud, a_j, is identified, using the nearest neighbor algorithm. Then, the angular similarity of tangent planes is computed using the corresponding normal vectors and based on the angle θ, which denotes the minimum out of the two angles that are formed by the intersecting tangent planes [wg1/n81067], as shown in Equation 4.

$$Angular\ Similarity\ =\ 1-\frac{\theta}{\pi} \tag{4}$$

This error value is associated to every b_i and provides an approximation of the dissimilarity between the underlying local surfaces. A total distortion measurement for the stimulus under evaluation is obtained as the mean squared error across all individual error values.

The plane to plane angular similarity is defined as the symmetric value, obtained after setting both point clouds **A** and **B** as the reference, and keeping the maximum error (i.e., minimum similarity).

The computation of the angular similarity requires the presence of the normal vectors for both the original and the distorted contents. To reduce the normal estimation error, which affects the performance of the metric [wg1/m88038], the normal vectors for both the reference and the distorted models should be (re-)estimated using a quadric local surface fitting model with a neighborhood radius of 20, as implemented in CloudCompare [CloudCompare19]. The normal estimation tool in CloudCompare can be accessed either through the Graphical User Interface[1], or through the command line mode, using the following command line:

```
   ./CloudCompare -SILENT -O fileIn.ply -C_EXPORT_FMT 'PLY' -OCTREE_NORMALS
R MODEL QUADRIC
```

The plane to plane angular similarity will be computed by the JPEG Committee based on the normal vectors estimated from the aforementioned procedure, using the script given in:

https://github.com/mmspg/point-cloud-angular-similarity-metric.

5.4 Point to Point Attribute Measure

The point to point attribute metric is based on the error of attribute values of associated points between the reference point cloud and the content under evaluation. In particular, for every point of the content under evaluation, a point that belongs to the reference point cloud is identified, through the nearest neighbor algorithm as described in Section 5.1. Then, an individual error is computed based on the Euclidean distance. For color attributes, the MSE for each of the three color components is calculated. For near-lossless coding, the maximum error should be considered instead of MSE. A conversion from RGB space to YCbCr space is conducted using ITU-R BT.709 [BT709], Since YCbCr space correlates better with human perception. PSNR value is then computed as:

$$PSNR = 10\log_{10}\left(\frac{p^2}{MSE}\right)\ ,$$

A symmetric computation of the distortion is utilized, in the same way as is done for geometric distortions and described in Section 5.1. The maximum distortion between the

[1] https://www.cloudcompare.org/doc/wiki/index.php?title=Normals%5CCompute

two passes is selected as the final distortion. Since the color attributes for all test data have a bit depth of 8 bits per point, the peak value p for PSNR calculation is 255.

This measure may be expressed as separate PSNR values for the Y, Cb and Cr channels are combined by:

$$PSNR_{Colour} = \frac{6PSNR_Y + PSNR_{C_b} + PSNR_{C_r}}{8}$$

where $PSNR_Y$, $PSNR_{C_b}$, and $PSNR_{C_r}$ are the PSNR values for the Y, Cb and Cr channels respectively.

The Point to Point Attribute Measure metric will be computed by the JPEG Committee using the software supplied by WG07 at:

http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric/tree/master

The tag "release-v0.13.5" indicates the current released version of the software, which will be used for this work.

6 Target Bit Rates

Submitters are expected to encode these point clouds at target bitrates ranging from 4 bits per input point down to a bitrate 0.1 bits per input point. It is expected that each test point cloud that will be considered for subjective evaluations will have to be encoded at the target bitrates (within a tolerance of +-10%) shown in Table 4 within the range above.

Table 4 - Target bit rates for submitted codecs (bits per point)

0.1	0.35	1.0	2.0	4.0

7 Anchor Generation

7.1 Anchor Codecs

The anchors will be generated by the G-PCC (TMC13) [G-PCC19], [G-PCC19b], and V-PCC (TMC2) codecs [V-PCC19], [V-PCC19b]. Reference implementations of both of these codecs can be found at [MPEG19]. Before encoding all point clouds will be scaled to a bounding box of size 1 and translated to the origin (0,0,0).

7.2 G-PCC Coding Parameters

The software for G-PCC is available from the MPEG GitLab:

http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc13.git.

Software documentation and usage are described in [G-PCC19b]. The "MPEG PCC tmc13 version release-v7.0-0-g47cf663" software should be used. G-PCC contains in practice 2 geometry encoders (Octree and TriSoup) and 2 color encoders (Predlift and RAHT), which can be combined, leading to a total of 4 variants. At the current time, the JPEG Pleno Point Cloud AhG will restrict attention to Octree+Predlift and TriSoup+Predlift. This is because in previous experimentation, JPEG Experts report that Predlift performance seems superior to RAHT, when subjectively compared [Alexiou, 2019] under the MPEG Common Test

Conditions [wg11/n18665]. Configuration files for the above parameters can be found for every anchor configuration in cfg_ctc.zip in the JPEG PCC working directory at:

https://drive.google.com/drive/folders/1vmlZOJwaUnfa5R2F4LfHz4nf7mY6oRW-?usp=sharing.

7.2.1 Octree Geometry Coding Parameters

G-PCC expects geometry to be expressed with integer precision, so all content must be quantised or voxelized prior to encoding. This should be achieved by octree voxelization to a depth that preserves the number of points in the original point cloud.

The following parameters define the codec configuration to be used in evaluation of octree geometry.

- `mode=0`
- `trisoup_node_size_log2=0`
- `mergeDuplicatedPoints=1`
- `ctxOccupancyReductionFactor=3`
- `neighbourAvailBoundaryLog2=8`
- `intra_pred_max_node_size_log2=6`
- `mergeDuplicatedPoints=1` for lossy geometry conditions.
- `mergeDuplicatedPoints=0` for lossless geometry conditions.

Lossless Geometry

The `positionQuantizationScale` parameter is set to 1.

Near-Lossless Geometry

The `positionQuantizationScale` parameter is set according to Table 5 for each point cloud in the test set.

Table 5 - `positionQuantizationScale` settings for each point cloud for G-PCC Octree geometry coding.

Point Cloud	`positionQuantizationScale` at rate point				
	R5	R4	R3	R2	R1
longdress	15/16	7/8	3/4	1/2	1/4
soldier	15/16	7/8	3/4	1/2	1/4
ricardo10	15/16	7/8	3/4	1/2	1/4
phil9	31/32	15/16	7/8	3/4	1/2
romanoillamp	15/16	7/8	3/4	1/2	1/4

bumbameuboi	15/16	7/8	3/4	1/2	1/4
guanyin	15/16	7/8	3/4	1/2	1/4
rhetorician	15/16	7/8	3/4	1/2	1/4

For point cloud content with colour information (*longdress*, *soldier*, *ricardo10*, *phil9*, *romanoillamp, bumbameuboi, guanyin,* and *rhetorician*) the following settings define the codec configuration in addition to those above.

- `colorTransform=1`
- `transformType=2`
- `numberOfNearestNeighborsInPrediction=3`
- `levelOfDetailCount=12`
- `positionQuantizationScaleAdjustsDist2=1`
- `dist2=3`
- `lodDecimation=0`
- `adaptivePredictionThreshold=64`
- `qpChromaOffset=0`
- `bitdepth=8`
- `attribute=color`

The `qp` parameter is set according to Table 6 for each rate point.

Table 6 - `qp` settings for each rate for G-PCC Octree geometry coding.

R5	R4	R3	R2	R1
22	28	34	40	46

7.2.2 TriSoup Geometry Coding Parameters

The following parameters define the codec configuration to be used in evaluation of TriSoup geometry.

- `mode=0`
- `mergeDuplicatedPoints=1`
- `ctxOccupancyReductionFactor=3`
- `neighbourAvailBoundaryLog2=8`
- `intra_pred_max_node_size_log2=6`
- `mergeDuplicatedPoints=1` for lossy geometry conditions.
- `mergeDuplicatedPoints=0` for lossless geometry conditions.
- `inferredDirectCodingMode=0`

Lossy Geometry

The positionQuantizationScale parameter is set according to Table 7. For a given test sequence, the same value is used for all rate points.

Table 7 - positionQuantizationScale settings for each point cloud for G-PCC TriSoup geometry coding.

Point Cloud	`positionQuantizationScale`
longdress	1
solider	1
ricardo10	1
phil9	1/2
romanoillamp	1
bumbameuboi	1
guanyin	1
rhetorician	1

The `trisoup_node_size_log2` parameter is set according to Table 8.

Table 8 - `trisoup_node_size_log2` settings for each rate for G-PCC TriSoup geometry coding.

R5	R4	R3	R2	R1
0	2	3	3	4

For point cloud content with colour information (*longdress*, *soldier*, *ricardo10*, *phil9*, *romanoillamp, bumbameuboi*, *guanyin* and *rhetorician*) the following settings define the codec configuration in addition to those above.

- `colorTransform: 1`
- `transformType: 2`
- `numberOfNearestNeighborsInPrediction: 3`
- `levelOfDetailCount: 12`
- `positionQuantizationScaleAdjustsDist2: 1`

- `dist2: 3`
- `lodDecimation: 0`
- `adaptivePredictionThreshold: 64`
- `qpChromaOffset: 0`
- `bitdepth: 8`
- `attribute: color`

The `qp` parameter is set according to Table 9 for each rate point.

Table 9 - `qp` settings for each rate for G-PCC TriSoup geometry coding.

R5	R4	R3	R2	R1
22	28	34	46	52

7.3 V-PCC Coding Parameters

The software implementation of V-PCC to be used is available from the MPEG GitLab:

http://mpegx.int-evry.fr/software/MPEG/PCC/TM/mpeg-pcc-tmc2.git.

Details about the V-PCC algorithm can be found in [V-PCC19] and the software documentation and usage description are provided in [V-PCC19b]

Additional software required for running V-PCC (please make sure to use exact software tags):

HEVC HM 16.20 with Screen Content Coding Extension (SCC) 8.8:

https://hevc.hhi.fraunhofer.de/svn/svn_HEVCSoftware/tags/HM-16.20+SCM-8.8/.

HDRTools version 0.18:

https://gitlab.com/standards/HDRTools/tags/v0.18.

Building the encoder and decoder applications should be done following the step-by-step instructions provided in GitLab and documentation listed above.

7.3.1 Coding Parameters and Configuration files

The V-PCC software supports config file handling via a `-c/--config=option` command line parameter. Multiple config files may be specified by repeating the option, with settings in later files overriding those in earlier ones.

The coding parameters and specialized sub-component (HEVC and HDRTools) settings are specified in several configuration files which can be found in the *common*, *rate*, *sequence*, *condition*, *hm* and *hdrconvert* folders found under the /cfg directory of the provided software. For the preparation of these anchors only the files in the rate and sequence may need changing, the first to specify geometry, color and occupancy maps representation precision/quantization and the second to specify the name and location of the input point cloud as well as its voxelization precision/depth expressed in bits.

A rate configuration file could contain the three lines defining three fidelity parameters for the geometry, color and occupancy maps:

```
##

# Rate parameters for R05

geometryQP: 16

textureQP: 22

occupancyPrecision: 2
```

and a sequence configuration file could include the following lines:

```
# longdress PC voxelized with 10 bits precision

uncompressedDataPath: voxelized/fullbodies/longdress_vox10_1300.ply

geometry3dCoordinatesBitdepth: 10
```

Since V-PCC is a point cloud sequence encoder and the anchors consist of single point cloud, the encoding condition to be selected is All Intra, with configuration parameters listed in the ctc-all-intra.cfg file to be found on the condition folder. This file should not be changed.

Similarly, the file ctc-common.cfg from the *common* folder should be used to specify lossy encoding without point cloud partitioning (`enablePointCloudPartitioning: 0`). To prepare the anchors it is not necessary to change the contents of this file.

The remaining sub-folders, *hm* and *hdrconvert* contain HEVC and HDRTools configuration presets which are referred to by the *ctc-common.cfg* and *ctc-all-intra.cfg* files and ordinarily should not be modified.

All these configuration files are specified in the command line invoking the V-PCC encoder, as in the following example:

```
PccAppEncoder            --config=common/ctc-common.cfg            --
config=sequence/longdress_vox10_1300.cfg --config=condition/ctc-all-
intra.cfg  --config=rate/ctc-r5.cfg  --configurationFolder=cfg/  --
uncompressedDataFolder=pcdata/ --colorSpaceConversionPath=HDRConvert
--videoEncoderPath=TAppEncoderStatic        --nbThread=1          --
keepIntermediateFiles=1                                           --
reconstructedDataPath=longdress_vox10_1300_ai_r05.ply             --
compressedStreamPath=longdress_vox10_1300_ai_r05.bin
```

7.3.2 Rate points and corresponding fidelity parameters

The V-PCC encoded anchors were prepared using 6 sets of parameters specifying (indirectly) six different rates, according to Table 10:

Table 10 - Rate settings for V-PCC anchors

Rate	Geometry QP	Texture QP	Occupancy Map Precision
R01	36	47	4
R02	32	42	4

R03	28	37	4
R04	20	27	4
R05	16	22	2

These parameter sets specify encoding configurations with increasing quality (and bitrate) as can be confirmed in the accompanying results Excel files.

8 References

[8i17] Eugene d'Eon, Bob Harrison, Taos Myers, and Philip A. Chou, "8i Voxelized Full Bodies - A Voxelized Point Cloud Dataset," ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Geneva, January 2017.

[MVUB16] C. Loop, Q. Cai, S. Orts Escolano, and P.A. Chou, "Microsoft Voxelized Upper Bodies – A Voxelized Point Cloud Dataset," *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document m38673/M72012*, Geneva, May 2016.

[Alexiou, 2020] E. Alexiou, N. Yang and T. Ebrahimi, "PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality," *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*, Athlone, Ireland, 2020, pp. 1-6.

[CloudCompare19], CloudCompare: 3D point cloud and mesh processing software Open Source Project, https://www.danielgm.net/cc/, 2019.

[BT50013] ITU-R BT.500-13, "Methodology for the subjective assessment of the quality of television pictures," International Telecommunications Union, Jan. 2012.

[MPV19] MPV video player, https://mpv.io, 2019

[wg11/n18665] ISO/IEC JTC1/SC29/WG11 MPEG2016/n18665, "Common Test Conditions for PCC", July 2019, Gothenburg, Sweden.

[wg1/n81067] ISO/IEC JTC1/SC29/WG 1 M81049, "JPEG Pleno - Overview of Point Cloud", October 2018, Vancouver, Canada.

[Alexiou, 2018] E. Alexiou, T. Ebrahimi, "Point cloud quality assessment metric based on angular similarity," *2018 IEEE International Conference on Multimedia and Expo (ICME)*, San Diego, CA, 2018, pp. 1-6.

[wg1/m88038] ISO/IEC JTC1/SC29/WG 1 M88038, "Benchmarking of the plane-to-plane metric", July 2020, Online.

[BT709] ITU-R BT.709, Parameter values for the HDTV standards for production and international programme exchange, Jun. 2015.

[Alexiou, 2019] E. Alexiou, I. Viola, T. M. Borges, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi. "A comprehensive study of the rate-distortion performance in MPEG point cloud compression." *APSIPA Transactions on Signal and Information Processing, 8*, E27(2019).

[G-PCC19] ISO/IET JTC1/SC29/WG11, WG11N18189, "G-PCC codec description v2", Jan 2019.

[G-PCC19b] ISO/IET JTC1/SC29/WG11, WG11N18473, "G-PCC Test Model v6", March 2019, Geneva, Switzerland.

[V-PCC19] ISO/IEC JTC1/SC29/WG11 MPEG2019/N18190, "V-PCC Codec description (TMC2 release v5.0)", Jan 2019.

[V-PCC19b] ISO/IET JTC1/SC29/WG11, WG11N18475, "V-PCC Test Model v6", March 2019, Geneva, Switzerland.

[MPEG19], ISO/IET/JTC1/SC29/WG11, “MPEG Point Cloud Compression”, http://www.mpeg-pcc.org/, 2019.