TITLE: JPEG Pleno Point Cloud Coding Common Training and Test Conditions v1.2

SOURCE: Luis Cruz (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 Luis Cruz, by downloading this document, editing it using track changes and sending it to lcruz@deec.uc.pt

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
# Table of Contents

1 Scope ......................................................................................................................... 4

2 Point Cloud Datasets .................................................................................................. 4
   2.1 Training dataset ....................................................................................................... 4
   2.2 Test dataset ........................................................................................................... 5

3 Training and Validation Processes .................................................................................. 5

4 Coding Conditions ......................................................................................................... 5
   4.1 Lossy coding target coding rates ........................................................................... 5
   4.2 Lossless coding conditions ....................................................................................... 6
   4.3 Scalable coding rates ............................................................................................... 6

5 Performance Assessment ............................................................................................... 6
   5.1 Rate metrics ........................................................................................................... 6
   5.2 Objective quality metrics ....................................................................................... 7
   5.3 Objective rate-distortion metrics .......................................................................... 7
   5.4 Subjective quality assessment .............................................................................. 7
   5.5 Scalability cost assessment .................................................................................. 8
   5.6 Random access cost assessment .......................................................................... 9

6 Complexity Evaluation .................................................................................................. 9

7 References ..................................................................................................................... 11

Annex A Point Cloud Datasets ......................................................................................... 14
   A.1 Training dataset ...................................................................................................... 14
   A.2 Test dataset ........................................................................................................... 18

Annex B Anchor Coding Configurations .......................................................................... 18
   G-PCC Encoder Configuration and Coding Parameters ............................................... 19
   V-PCC Encoder Configuration and Coding Parameters ............................................... 20

Annex C Objective Quality Measures .............................................................................. 21
   C.1 Geometry-only coding metrics .............................................................................. 21
   C.2 Texture-only quality metrics ................................................................................ 22
   C.3 Geometry and texture quality metric ...................................................................... 23

Annex D Subjective Quality Evaluation Protocols .......................................................... 25
Scope

This document describes the Common Training and Test Conditions (CTTC) to be used to train and evaluate the point cloud coding solutions submitted to the Call for Proposals on JPEG Pleno Point Cloud Coding (CfP) [wg1/N100097]. As defined in the CfP the coding technologies sought shall be learning-based solutions and target the coding of geometry-only point clouds as well as point clouds with geometry and associated texture information.

All coding solutions submitted to the CfP should be trained using the data described in Section 2, targeting the coding modes and rates specified in Section 4. The submitted codecs will be evaluated following the procedures, tools and datasets described later in these CTTC.

Point Cloud Datasets

Reflecting the fact that the CfP asks for coding solutions developed using learning-based techniques, this CTTC document defines and provides access information to a training dataset to be used in the codec design. An additional dataset to be used in the test and performance assessment of the codecs will also be made available, at a later stage. The point clouds included in the datasets will be chosen to ensure diversity in terms of content type, geometry and texture complexity, spatial point density and inclusion of texture information.

2.1 Training dataset

The training dataset is an ensemble of 3D point clouds collected by WG1 experts and will be provided to the CfP submitters to train their coding solutions. The dataset contents are representative of point cloud data commonly used in the most important use cases specified in [wg1/N100096] involving human observers and machine processing. The dataset comprises a mix of point clouds with variable number of points, with and without texture. The contents of this dataset were gathered from several sources like public access repositories of point clouds and meshes, repositories accompanying academic and industrial research works published in the literature and datasets created by standardization groups from bodies like ISO/IEC JTC 1/SC29.

Details about this dataset, like information about the sources of the data, rights-of-use information, file formats, geometry and texture representation formats and access information can be consulted in Annex A.1.
2.2 Test dataset

This dataset is a compilation of 3D point clouds to be used in evaluating the codecs submitted to the CfP. To guarantee fair testing of the submitted codecs, an effort will be made to ensure the test dataset contents will be similar enough to those of the training dataset. This dataset will be compiled by the PCC AhG chairs (Stuart Perry and Luis Cruz), from sources that will remain undisclosed, and will be accessible to the submitters only after the proposal’s submission period lapses and the submissions are sealed.

3 Training and Validation Processes

Training of codecs must be based on the dataset described in Annex A.1.

However, alternative or complementary training datasets may be used for codecs submitted to be evaluated. In this case, the CfP outlines conditions wherein other datasets may be used.

Acknowledging that there are different ways to train and validate deep learning-based processing systems, training using subsets of the dataset described in Annex A.1 is allowed. However, detailed information about the process followed to train the models has to be disclosed to WG1, to allow independent verification in the context of the codec evaluation procedures. This information includes details such as whether the entire dataset was used in the training and validation, how it was partitioned between training and validation datasets, what hyper-parameters were used, what validation procedure was followed, as well as any other information needed to reproduce the training and validation processes.

4 Coding Conditions

4.1 Lossy coding target coding rates

For the testing of learning-based codecs, a test dataset will be supplied by the JPEG Committee. To ensure fairness of evaluation, this test dataset is not disclosed in this document, however conditions of encoding of the test dataset such as the target bit rates will be the same as described here for the training dataset.

Test point clouds coded by learning-based codecs should be encoded at four bitrates. The four target rates are listed in Table 1. Each of the encoded rates may be below, but should not be greater than the respective target rates listed in Table 1 by more than 10%.

For solutions coding both texture and geometry, the target total rates (geometry and texture) are shown in the first row of Table 1. For solutions coding only the geometry information, the target bitrates are shown in the second row of Table 1. In all plots and comparisons, the bitrates of solutions that code only geometry should be compared to the bit rate of the geometry component of the anchors.
Table 1 - Target bit rates for submitted codecs (bpip - bits per input point)

<table>
<thead>
<tr>
<th>Components coded</th>
<th>Rate 1</th>
<th>Rate 2</th>
<th>Rate 3</th>
<th>Rate 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>Geometry and Texture</td>
<td>0.1</td>
<td>0.3</td>
<td>1.0</td>
<td>3.0</td>
</tr>
<tr>
<td>Geometry</td>
<td>0.05</td>
<td>0.15</td>
<td>0.5</td>
<td>1.5</td>
</tr>
</tbody>
</table>

4.2 Lossless coding conditions

For codecs that support only lossless or both lossy and lossless coding, two lossless specific coding conditions will have to be used: i) lossless geometry-only coding; and ii) lossless geometry and texture coding. Naturally, there are no target bitrates for these two coding conditions.

4.3 Scalable coding rates

For codecs under evaluation offering scalability, there will be no predefined number of layers or per-layer bitrates. However, if the coding technology allows it, codecs supporting scalable coding will be evaluated at the rates listed in Table 1 where the first rate will be the base layer rate, the second rate is the sum of the base layer and first enhancement layer rate and so on. Please see Figure 1 for a graphical description of the scalability rates definitions. If the codec supports multiple types of scalability, e.g. resolution scalability and quality scalability, then the submitters should provide data for all scalability types.

5 Performance Assessment

The codecs under evaluation will be compared to two reference codecs, henceforth called anchors. The comparisons will take into account four key performance indicators: coding rate, objective quality, subjective quality and computational complexity. Additionally, scalability and random access performance will be measured, as well as memory requirements, as detailed in the following sections.

One of the anchor codecs is a state-of-the-art coder based on 3D spatial hierarchical partitioning, the MPEG G-PCC standard in the Octree coding mode [G-PCC19], [G-PCC19b]. The other state-of-the-art anchor codec to be used is the MPEG V-PCC standard [V_PCC], a codec based on projections onto 2D planes that leverages legacy 2D video encoders like HEVC and VVC to encode the projections.

The most recent versions of the MPEG G-PCC and V-PCC standards reference software available should be used wherever possible.

Reference codings should use the configurations and parameters described in Annex B.

5.1 Rate metrics

The basic metric to measure coding performance in the bit expenditure dimension is the ratio of the number of bits produced by the encoder when coding a given point cloud, to the number of points of that (original) point cloud. Values of these measures will be loosely designated by “rate”.
The rates specified in this document and reported during codec evaluation should account for the total number of bits, \( \text{TotNBits} \), of the file (or files) containing the bitstream and all auxiliary information required by the decoder to reconstruct a lossy or lossless version of the entire input point cloud.

Formally, the rate metric is the number of bits per input point (bpip) defined as:

\[
\text{Rate} = \frac{\text{TotNBits}}{\text{TotNPoints}}
\]

where \( \text{TotNBits} \) is the number of bits for the compressed representation of the point cloud and \( \text{TotNPoints} \) is the number of points in the input point cloud. In addition to the total bits per input point, the bits per input point used to code the geometry and the bits per input point used to code the texture (if coded) should also be reported.

### 5.2 Objective quality metrics

The objective quality of decoded point clouds will be measured by mathematically defined similarities between the decoded point cloud and the corresponding original point cloud taken as reference. The following objective quality metrics will be used:

- **Geometry-only**: Two different metrics will be used to measure the geometry quality, PSNR\(_D1\) computed from the average point-to-point distance a.k.a. D1 measure and PSNR\(_D2\) computed from the average point-to-plane distance a.k.a. D2 measure. These measures are defined in Annex C.1.
- **Texture**: The texture quality will be measured by PSNR\(_Y\) and PSNR\(_\text{Color}\) as defined in Annex C.2.
- **Joint geometry and texture**: For point clouds with both geometry and texture, besides the previous geometry and texture measures (PSNR\(_D1\), PSNR\(_D2\), PSNR\(_Y\), and PSNR\(_\text{Color}\)) and PCQM measures will also be computed, where the latter is defined in Annex C.3.

All the metrics should be computed following the procedures and mathematical formulations described in Annex C or using the implementations listed therein.

### 5.3 Objective rate-distortion metrics

The submitted codecs will be compared to the MPEG G-PCC (Octree) and V-PCC anchors by computing Bjontegard Deltas (BD) based on the rate-distortion pairs for all rate points defined in Section 4.2 and the objective quality metrics selected in Section 5.2.

The RD performance for geometry-only codecs should be compared to the anchor RD performance where the rate accounts for all the information needed to reconstruct the geometry component.

### 5.4 Subjective quality assessment

Subjective evaluation of the submitted learning-based point cloud codecs will be limited to proposals that code both the geometry and texture. For subjective evaluation, it is important to ensure consistent criteria for rendering the point cloud data. Content preparation and rendering, presentation, grading and statistical grade analysis should be done following the protocols described in Annex D.
Mean opinion scores (MOS) for each test content coded at each rate together with the respective rate will be used to compute BD-Rate and BD-MOS figures of merit for each codec submission.

In the case of point cloud codecs addressing only geometry information, subjective evaluation is not conducted, and quality assessment relies solely on the geometry-only objective metrics described in sections 5.2 and 5.3, namely PSNRD1, and PSNRD2.

5.5 Scalability cost assessment

If a codec submitted for evaluation supports scalability, the submitters should provide information about the cost of supporting scalability. Computing this cost may depend on how the submitted codec implements scalability but if possible the following protocol should be observed. The scalability cost shall be measured empirically using the test dataset to be provided. For each supported scalability type (e.g. resolution and quality), the test data shall be scalably encoded with a number of layers corresponding to the number of rates listed in Table 1 and using those rates as targets for the scalable encoding as explained in section 4.3. The rates listed in Table 1 will also be used as target coding rates for coding the same point clouds using the same codec in non-scalable mode. After decoding the scalable bitstreams and computing the quality of the reconstructed point-cloud for each point cloud and each rate, it is straightforward to compare the scalable and non-scalable encoding mode and compute the scalability cost.

For codecs that support geometry and texture this cost should be measured by Bjontegaard Deltas (BD) between the two sets of rate-distortion pairs. For each scalable encoding, two sets if BD values will have to be provided:

- Two BD values measuring the R-D performance in the PSNRD1, geometry bitrate plane, namely BD-RateD1 measuring the geometry bitrate increase with respect to PSNRD1 distortion and BD-PSNRD1 measuring the PSNRD1 increase with respect to geometry bitrate,
- Two BD values measuring the R-D performance in the PCQM, total bitrate plane, namely BD-Rate1-PCQM measuring the total bitrate increase with respect to 1-PCQM (i.e. one minus the value of PCQM) and BD-1PCQM measuring the 1-PCQM increase with respect to total bitrate.

Codecs that support only geometry coding shall use the rates indicated in Table 1 for geometry-only encoding and the scalability costs to be computed are BD-RateD1 and BD-PSNRD1.

Figure 1 shows a graphical representation of the coder operating points in scalable and non-scalable modes, for the case of 1-PCQM quality objective quality measure and total bitrate.
Reports on scalability should report on the types of scalability supported, number of scalability layers used, bits required to achieve each specific scalability level, and the resulting BD values. If the scalability cost is computed in a different way, the submitters are requested to provide detailed information about the procedure followed to compute the cost(s) reported and the scripts needed to compute those costs.

5.6 Random access cost assessment

The cost to access a specific Region of Interest (RoI) of a point cloud, the RoI Random Access Cost shall be measured, for a given quality, by the ratio between the number of bits that need to be decoded to reconstruct the RoI points and the total number bits of the compressed representation of the point cloud.

\[
\text{RoI RA Cost} = \frac{\text{Total amount of bits to be decoded to access an RoI}}{\text{Total amount of bits of the coded point cloud}}
\]

To assess the random access cost, the codecs to be evaluated must provide at least all the decoded points belonging to the requested RoIs, but can also provide extra points, depending on the coding algorithm and coded data organization. Submitters should report the maximum (worst) value for the RoI RA Cost computed for RoIs that correspond to meaningful regions of the point cloud (e.g. the head in a point cloud depicting a full human body).

6 Complexity Evaluation

The complexity of the proposed coding solutions will be measured in two major dimensions:

- Resources memory/footprint
- Computational complexity

In the first dimension, the metric of interest will be the size of the model as measured by the number of coefficients/weights/bias and their precision. These values are related to
the memory footprint of the model and provide a rough indication of the model computational complexity as more coefficients imply more arithmetic operations. More precise measures of the model computational complexity will be obtained by recording the running times of the model, during inference, both for encoding and decoding, with and without GPU assistance. Table 2 provides more details of the metrics to be measured to estimate the model complexity.

Table 2 - Complexity Estimation.

<table>
<thead>
<tr>
<th>Type</th>
<th>Metric</th>
<th>Definition</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Resource Footprint</td>
<td>Number of Parameters</td>
<td>Total number of parameters of the model</td>
<td>The model file size can be reported as a proxy for the model memory footprint.</td>
</tr>
<tr>
<td></td>
<td>Parameter Precision</td>
<td>Precision of the parameters (e.g. integer 8 bits, float 32 bits) for both the model weights and the inter-layer activation/buffers.</td>
<td></td>
</tr>
<tr>
<td>Computational Complexity</td>
<td>Number of MAC ops per Thousand of Input Point Cloud Points (MACs/kip)</td>
<td>(MACs/kip) required for encoding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Number of MAC ops per Thousand of Input Point Cloud Points (MACs/kip)</td>
<td>(MACs/kip) required for decoding</td>
<td></td>
</tr>
<tr>
<td></td>
<td>Encoding running time</td>
<td>Average, minimum and maximum running times of the encoder stages, up to and including bitstream generation.</td>
<td>1. Encoding and decoding should be run ten times.</td>
</tr>
<tr>
<td></td>
<td>CPU only</td>
<td></td>
<td>2. Report values for each point cloud, as well as</td>
</tr>
</tbody>
</table>
Decoding running time
CPU only

Average, minimum
and maximum running
times of the decoder
stages, from
bitstream to point
cloud reconstruction.

minimum and
maximum values
and weighted
average for all
point clouds with
weights being the
number of points
of the point
clouds.

3. Ensure no other
are processes
running, except
for the OS.

Encoding running time
with GPU

Average, minimum
and maximum running
times of the encoder
stages, up to and
including bitstream
generation.

Decoding running time
with GPU

Average, minimum
and maximum running
times of the decoder
stages, from
bitstream to point
cloud reconstruction.

The resource (memory) footprint and computational complexity of the anchor will also be
measured. The computational complexity will be measured by the encoding and decoding
running times using the same method as for the submitted codecs, and the memory use will
be estimated using operating system process monitoring tools.

To allow an informed analysis of the complexity results, the machine learning framework,
OS, CPU make, model and frequency, RAM size and GPU details should be reported as well
as the details and versions of any supporting software required to reproduce the results.

7 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.

Bodies - A Voxelized Point Cloud Dataset,” ISO/IEC JTC1/SC29 Joint WG11/WG1
(MPEG/JPEG) input document m38673/M72012, Geneva, May 2016.

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


Annex A  Point Cloud Datasets

A.1 Training dataset

The codecs submitted to the CfP should be trained with the point clouds from the datasets listed in Table A.1.I. In the cases where the original dataset includes meshes, an alternative dataset consisting of versions of the meshes sampled to point clouds will be provided on request. The remaining datasets released already as point clouds should be previously voxelized to 10 bits prior to the training. This voxelization step is already included in the sampling script for the previous category.

Table A.1.I - Datasets

<table>
<thead>
<tr>
<th>Dataset name</th>
<th>Number of models</th>
<th>Average points per model</th>
<th>Attributes</th>
<th>Type</th>
<th>Original file extension</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShapeNetCore</td>
<td>52'472</td>
<td></td>
<td>Geometry + colour</td>
<td>Synthetic mesh</td>
<td>.obj and .mtl</td>
</tr>
<tr>
<td>MPEG and JPEG Pleno</td>
<td>47</td>
<td>10'096'925</td>
<td>Geometry + colour</td>
<td>Real-world point cloud</td>
<td>.ply</td>
</tr>
</tbody>
</table>

A description of each dataset including instruction for access as well as licensing information can be found here below:

**ShapeNetCore:** ShapeNet [Chang, 2015] is a large dataset produced as a collaborative effort between researchers at Princeton University, Stanford University and TTIC. ShapeNetCore is a subset of ShapeNet containing CAD models of single objects obtained from 3D public repositories with associated category labels. The original dataset can be accessed at [https://shapenet.org/](https://shapenet.org/) for users with a verified account. In order for the user to be able to download the dataset, he should agree with the terms of use which states that the use of the database is restricted to non-commercial research and educational purposes. JPEG will provide a version of this dataset with the meshes sampled to point clouds.

**MPEG and JPEG Pleno:** Both these datasets contain several high resolution point clouds from multiple sources, which were already provided to assist standardization process. While the JPEG Pleno dataset can be freely accessed at [http://plenodb.jpeg.org/](http://plenodb.jpeg.org/), credentials are needed for accessing the MPEG dataset at [https://mpegfs.int-evry.fr/mpegcontent/](https://mpegfs.int-evry.fr/mpegcontent/). A complete list of the point clouds considered for the training dataset, as well as other relevant information is included at Table A.1.II. When considering dynamic point clouds,
the number of frames employed for each point cloud model in the training set is a choice of the proponent.

<table>
<thead>
<tr>
<th>Point cloud name</th>
<th>Source</th>
<th>Number of points</th>
<th>Type</th>
<th>Access provided by</th>
</tr>
</thead>
<tbody>
<tr>
<td>Longdress</td>
<td>8ilabs</td>
<td>857'966</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Loot</td>
<td>8ilabs</td>
<td>805'285</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Redandblack</td>
<td>8ilabs</td>
<td>757'691</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Soldier</td>
<td>8ilabs</td>
<td>1'089'091</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Thaidancer</td>
<td>8ilabs</td>
<td>3'130'215</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Boxer</td>
<td>8ilabs</td>
<td>3'493'085</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Andrew10</td>
<td>MVUB</td>
<td>1'276'312</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>David10</td>
<td>MVUB</td>
<td>1'492'780</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Phil10</td>
<td>MVUB</td>
<td>1'660'959</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Ricardo10</td>
<td>MVUB</td>
<td>960'703</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Sarah10</td>
<td>MVUB</td>
<td>1'355'867</td>
<td>Dynamic</td>
<td>JPEG</td>
</tr>
<tr>
<td>Location</td>
<td>Provider</td>
<td>Size (Bytes)</td>
<td>Type</td>
<td>Format</td>
</tr>
<tr>
<td>------------------------</td>
<td>---------------</td>
<td>--------------</td>
<td>-----------</td>
<td>--------</td>
</tr>
<tr>
<td>ArcoValentino</td>
<td>GTI-UPM</td>
<td>1'530'939</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>PalazzoCarignano</td>
<td>GTI-UPM</td>
<td>4'260'757</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>VillaLaTesoriera</td>
<td>GTI-UPM</td>
<td>741'966</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>BumbaMeuBoi</td>
<td>EmergIMG</td>
<td>150'388</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>CITIUSP</td>
<td>EmergIMG</td>
<td>5'929'878</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>DouradoSite</td>
<td>EmergIMG</td>
<td>26'137'113</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>ErasmusMill</td>
<td>EmergIMG</td>
<td>4'412'779</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>Ipanema</td>
<td>EmergIMG</td>
<td>15'028'108</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>ItanguaCanyon</td>
<td>EmergIMG</td>
<td>5'456'996</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>RamosDeAzevedo</td>
<td>EmergIMG</td>
<td>64'153'694</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>RomanOilLight</td>
<td>EmergIMG</td>
<td>1'286'052</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>Labrador</td>
<td>EmergIMG</td>
<td>55'158</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>VillaLobosPark</td>
<td>EmergIMG</td>
<td>7'288'489</td>
<td>Static</td>
<td>JPEG</td>
</tr>
<tr>
<td>Name</td>
<td>Company</td>
<td>Duration</td>
<td>Type</td>
<td>Format</td>
</tr>
<tr>
<td>-----------------------</td>
<td>-------------</td>
<td>----------</td>
<td>----------</td>
<td>--------</td>
</tr>
<tr>
<td>The20sMaria</td>
<td>HHI</td>
<td>10'383'094</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>UlliWegner</td>
<td>HHI</td>
<td>879'709</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Queen</td>
<td>Technicolor</td>
<td>1'000'993</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Basketball_player</td>
<td>Owlii</td>
<td>2'880'057</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Dancer</td>
<td>Owlii</td>
<td>2'592'758</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Exercise</td>
<td>Owlii</td>
<td>2'391'718</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Model</td>
<td>Owlii</td>
<td>2'458'429</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Mitch</td>
<td>Volucap</td>
<td>2'289'640</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Thomas</td>
<td>Volucap</td>
<td>2'172'540</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>XDPDProd_football</td>
<td>XDPD</td>
<td>1'021'107</td>
<td>Dynamic</td>
<td>MPEG</td>
</tr>
<tr>
<td>Façade 00009</td>
<td>Buildings and Façades</td>
<td>1'602'990</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Façade 00015</td>
<td>Buildings and Façades</td>
<td>8'929'532</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Façade 00064</td>
<td>Buildings and Façades</td>
<td>19'714'629</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>House without roof</td>
<td>Buildings and Façades</td>
<td>5’001’077</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>-------------------------</td>
<td>-----------------------</td>
<td>-----------</td>
<td>--------</td>
<td>------</td>
</tr>
<tr>
<td>Egyptian_mask</td>
<td>Inanimate objects</td>
<td>272’689</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Frog_00067</td>
<td>Inanimate objects</td>
<td>3’630’907</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Head_00039</td>
<td>Inanimate objects</td>
<td>14’025’710</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Landscape_00014</td>
<td>Inanimate objects</td>
<td>72’145’549</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Shiva_00035</td>
<td>Inanimate objects</td>
<td>10’105’91</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Statue_klimt</td>
<td>Inanimate objects</td>
<td>499’886</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Stanford_Area_2</td>
<td>Large-Scale Static Scenes</td>
<td>54’989’822</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>Stanford_Area_4</td>
<td>Large-Scale Static Scenes</td>
<td>47’485’046</td>
<td>Static</td>
<td>MPEG</td>
</tr>
<tr>
<td>ULB_Unicorn_HiRes_Point_Cloud</td>
<td>ULB Unicorn</td>
<td>63’864’713</td>
<td>Static</td>
<td>MPEG</td>
</tr>
</tbody>
</table>

**A.2 Test dataset**

To be divulged after the submission trained encoder and decoder software implementations of codecs to be evaluated.

**Annex B Anchor Coding Configurations**

As specified before, the anchor codecs are MPEG G-PCC and MPEG V-PCC in intra coding mode. **In terms of JPEG activities, only anchors encoded by the JPEG Committee will be considered officially valid.**
Taking as reference implementations TMC13 v.13 for G-PCC and TMC2 v.14 for V-PCC with VVC 2D video encoder, the anchor encodings can be obtained as detailed next.

G-PCC Encoder Configuration and Coding Parameters

All G-PCC encodings should use Octree-Predlift coding modes for lossy geometry and colour coding.

Since TMC13 does not perform rate allocation, users often have to do multiple encodings with different \((\text{positionQuantizationScale}, \text{qp})\) parameters pairs to achieve rates as close as possible to those desired. Parameter \text{positionQuantizationScale} controls the geometry encoding fidelity and the parameter \text{qp} the encoding quality of the colour components.

These parameters are defined in the file “encoder.cfg” which has a structure similar to that in the following box,

```
mode: 0
trisoupNodeSizeLog2: 0
…
positionQuantizationScale: 0.875
…
adaptivePredictionThreshold: 64
qp: 28
```

Preparing and calling the encoder will be done according to the instructions provided at


As an example, the G-PCC TMC13 encoder can be run with a command line similar to

```
make -f $SCRIPTS_DIR/Makefile.tmc13-step -C ${OUTPUT_BASE_DIR}/${pcloud}/${rate}/ \
  VPATH=${OUTPUT_BASE_DIR}/${pcloud}/${rate}/ ENCODER=${BUILD_DIR}/tmc3/tmc3 \
  DECODER=${BUILD_DIR}/tmc3/tmc3 PCERROR=${PCERROR_DIR}/pc_error \
  SRCSEQ=${INPUT_PC_DIR}/${pcloud}.ply
```
V-PCC Encoder Configuration and Coding Parameters

The second anchor encodings have to be prepared with MPEG V-PCC TMC2 reference implementation (with VVC 2D encoder). In computing V-PCC anchors, JPEG follow MPEG configurations for lossy-geometry-lossy-attibute (intra only) and change only the three most important encoder parameters, the geometry and colour quantization parameters (QP), respectively geometryQP and attributeQP and the occupancy map precision, occupancyPrecision. All lossy encoding encodings used the configurations embodied in the common-ctc.cfg, vtm-all-intra.cfg, and ctc-{rate}.cfg files that accompany the TMC2 package found at

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

Since TMC2 does not have a rate allocation mechanism, users have to do multiple encodings with different combinations of the geometryQP, attributeQP and occupancyPrecision parameters to obtain encoding rates as close as possible to those desired.

Preparing and calling the encoder will be done according to the instructions provided at:

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

The V-PCC TMC2 encoder can be run with a command line similar to

```
${BIN_DIR}/PccAppEncoder \  
--config=${CFG_DIR}/common/ctc-common.cfg \  
--config=${CFG_DIR}/condition/vtm-all-intra.cfg \  
--config=${CFG_DIR}/inputpccfg.cfg \  
--config=${CFG_DIR}/ratecfg.cfg \  
--configurationFolder=${CFG_DIR}/ \  
--profileCodecGroupIdc=3 --profileToolsetIdc=1 --profileReconstructionIdc=2 \  
--mapCountMinus1=0 --pointLocalReconstruction=1 --pbfEnableFlag=1 \  
--useEightOrientations=1 --flagColorSmoothing=1 --additionalProjectionPlaneMode=5\  
--keepIntermediateFiles=1 \  
--computeMetrics \  
--reconstructedDataPath=${OUTPUT_BASE_DIR}/{pcname}.rec.vvc.ply \  
--compressedStreamPath=${OUTPUT_BASE_DIR}/{pcname}.bitstream.vvc.bin
```
Annex C  Objective Quality Measures

The objective quality metrics employed for the evaluation of the proposals can be classified into geometry-only, texture-only and geometry and texture. While only metrics from the first category can be used for proposals without texture coding, all metrics described in this section are applied for proposals supporting both geometry and texture coding. This annex describes the working principle of all employed metrics as well as instructions on how to apply them to distorted point cloud models.

C.1 Geometry-only coding metrics

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 neighbour 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 [WG11N18665], the point to point geometry PSNRD1 metric is computed as follows:

Using the notation of [WG11N18665], we first determine an error vector $E(i,j)$ denoting the difference vector between the identified point $a_i$ 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.,

$$e_{BA}^{D1}(i) = \| E(i,j) \|_2^2$$

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

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

$$\text{PSNRD1} = 10 \cdot \frac{3 \cdot \text{peak}^2}{\max(e_{BA}^{D1}, e_{AB}^{D1})}$$

where peak is the geometric resolution of the model (i.e. if voxel bit depth = 10, it is 1024) and $e_{BA}^{D1}$ 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:


The tag "release-v0.13.5" indicates the current released version of the software, which will be used for this work.
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 \[WG11N18665\]:

\[
e_{B,A}^{D2}(i) = \langle E(i, j), N_j \rangle^2
\]  
(3)

Where \( \langle x, y \rangle \) 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:


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 \[WG11N18665\]:

```
./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.

C.2 Texture-only quality metrics

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 neighbour algorithm as described in Section C.1. Then, an individual error is computed based on the Euclidean distance. For colour attributes, the MSE for each of the three colour components is calculated. 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:
\[ PSNR = 10 \left( \frac{p^2}{MSE} \right), \]

A symmetric computation of the distortion is utilized, in the same way as is done for geometric distortions and described in Section C.1. The maximum distortion between the two passes is selected as the final distortion. Since the colour 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 which are then combined by:

\[ PSNR_{\text{Colour}} = \frac{6PSNR_Y + PSNR_{Cb} + PSNR_{Cr}}{8} \]

where \( PSNR_Y, PSNR_{Cb}, \) and \( PSNR_{Cr} \) 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.

C.3 Geometry and texture quality metric

PCQM

The PCQM [Meynet, 2020] is based on the linear combination of multiple geometry-based and colour-based features computed between a distorted point cloud \( D \) and the corresponding reference \( R \). The steps to calculate the final metric value are described here below.

Initially, a correspondence between both point clouds \( D \) and \( R \) is established. For each point \( p \) in \( R \), the coordinate and colour of its corresponding point \( \hat{p} \) in \( D \) is obtained. Note that the inverse procedure is not considered, i.e. finding the correspondence of points from \( D \) into \( R \). While previous metrics considered the nearest neighbor for this operation, PCQM searches for the projection \( \hat{p} \) of \( p \) into the local underlying surface in \( D \).

This surface is obtained using quadric fitting in the set of nearest neighbours \( p_i^D \) in the distorted point cloud \( D \). The tangent plane is first estimated using Principal Component Analysis (PCA), yielding an orthonormal base \((u_x, u_y, u_z)\) from which \( u_x \) and \( u_y \) are contained in the tangent plane and \( u_z \) is aligned with its normal. Another coordinate system is then defined using this base, with \( p \) as its origin. Given that the coordinates of each point \( p_i \) in the set of the nearest neighbours \( p_i^D \) is now expressed by \((x_i, y_i, z_i)\) in this new system, the quadric surface \( Q(x, y) = ax^2 + by^2 + cxy + dx + ey + f \) is computed by minimizing \( \sum_i |z_i - Q(x_i, y_i)|^2 \).
The projection $\hat{p}$ is then found by the coordinates $(0,0,Q(0,0))$ according to the local frame with base $(u_x, u_y, u_z)$, which is then translated to the original coordinate system of the point cloud. The colour triplet corresponding to $\hat{p}$ is assigned as the same as the nearest neighbour of $\hat{p}$ in $D$.

The second step consists in determining local neighbourhoods $N(p, h)$ for each point $p$, which is done by selecting all the points in $R$ within a search radius of $h$ from $p$. The set $N(\hat{p}, h)$ is equally gathered from $D$ using the same distance for each point $\hat{p}$.

As a pre-processing stage for the geometry-based features computation, the curvature $\rho$ is calculated for each point $p$ and $\hat{p}$ using the coefficients for the local quadric surface fitted to the neighborhoods $N(p, h/2)$ and $N(\hat{p}, h/2)$, respectively. Similarly, the RGB values of $p$ and $\hat{p}$ are converted to the LAB2000HL [Lissner, 2012] colour space previously to the computation of the colour-based features, yielding a lightness value $L_p$ and two chroma values $a_p$ and $b_p$. The two latter are then combined into $c_p = \sqrt{a_p^2 + b_p^2}$.

The geometry-based and colour-based features are then computed for each point $p$ according to the equations:

Curvature comparison

$$f_1^p = \frac{\left| \mu_p^\rho - \mu_{\hat{p}}^\rho \right|}{\max(\mu_p^\rho, \mu_{\hat{p}}^\rho) + k_1}$$

Curvature contrast

$$f_2^p = \frac{\left| \sigma_p^\rho - \sigma_{\hat{p}}^\rho \right|}{\max(\sigma_p^\rho, \sigma_{\hat{p}}^\rho) + k_2}$$

Curvature structure

$$f_3^p = \frac{\left| \sigma_p^\rho \sigma_{\hat{p}}^\rho - \sigma_p^\rho \right|}{\sigma_p^\rho + k_3}$$

Lightness comparison

$$f_4^p = 1 - \frac{1}{k_4(\mu_p^L - \mu_{\hat{p}}^L)^2 + 1}$$

Lightness contrast

$$f_5^p = 1 - \frac{2\sigma_p^L \sigma_{\hat{p}}_L + k_5}{\sigma_p^L + \sigma_{\hat{p}}^L + k_5}$$

Lightness structure

$$f_6^p = 1 - \frac{\sigma_p^L + k_6}{\sigma_p^L + k_6}$$
Chroma comparison

\[ f_7^P = 1 - \frac{1}{k_7 (\mu_p^c - \mu_p^C)^2 + 1} \]

Hue comparison

\[ f_8^P = 1 - \frac{1}{k_B \cdot \Delta H_{p\beta}^2 + 1} \]

For the first three features based on curvature values, the values \( \mu_p^p \) and \( \mu_{\hat{p}}^p \) represent the gaussian-weighted averages of the curvature on the set of neighbours \( N(p, h) \) and \( N(\hat{p}, h/2) \), respectively, while \( \sigma_p^p, \sigma_{\hat{p}}^p \) and \( \sigma_{p\beta}^p \) are the standard deviations and covariance over these neighborhoods. Similarly, \( f_4 \) to \( f_7 \) use these same statistical variables for lightness and chroma, and \( f_8 \) uses \( \Delta H_{p\beta} = \sqrt{(a_p - a_{\hat{p}})^2 + (b_p - b_{\hat{p}})^2 - (c_p - c_{\hat{p}})^2} \) with \( \Delta H_{p\beta} \) being the gaussian-weighted average of this variable at the neighbourhood \( N(p, h) \). The constants \( k_1 \) to \( k_8 \) are used to avoid instabilities when denominators are close to zero, being set to one for the computation of the features.

For each feature, a global value is then obtained by pooling together \( f_i^P \) for the whole point cloud using the average operator, according to the following equation.

\[ f_i = \frac{1}{|R|} \sum_{p \in R} f_i^P \]

The final value for the PCQM metric is obtained by a weighted average of the computed features. We here employ the weights provided at [Meynet, 2020], which were optimized by logistic regression on a subset of the IST Rendering Point Cloud dataset [Javaheri, 2019]. The released model is composed only by the features \( f_3 \) (Curvature structure), \( f_4 \) (Lightness comparison) and \( f_6 \) (Lightness structure) and is given by Equation:

\[ PCQM = 0.18 f_3 + 0.44 f_4 + 0.38 f_6 \]

The employed implementation, as well as the instructions for building and executing the code, can be found at the following link:

[https://github.com/MEPP-team/PCQM](https://github.com/MEPP-team/PCQM)

**Annex D  Subjective Quality Evaluation Protocols**

The procedure defined in this section for the subjective evaluation of point cloud solutions is designed for point clouds with colour attribute information associated with each point.
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:

D.1 Rendering

- For rendering of Point clouds to a video, the CloudCompare software [CloudCompare19] will be used. The default parameters of CloudCompare software is kept the same for rendering.
- However, adjusting the point size of each viewport of the point cloud is necessary. This step is important because different point clouds will have different point size. Moreover, the same point clouds with different degradation levels can also have a separate point size. Therefore, the point size needs to be determined for each sequence. This is done by experimentation to render as far as possible the perception of a watertight surface.
- A black background is chosen to render the point clouds.
- Point clouds are rendered and viewed on 2D monitors. Resolution of the monitor is kept at 4096 x 2160. The colour gamut is considered as sRGB or wider.
- While rendering the point cloud, it is important to match the intrinsic resolution of the point clouds with the display resolution. Point density is ensured by maintaining that no more than one point should occupy a single pixel of the displayed video.
- Resolution of the frames can also be customized by adjusting the edge of 3D view window to the desired resolution.
- A full screen resolution 2D view of the point clouds can be achieved by pressing F11, before choosing the view and rendering properties.

D.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 Animation plugin of CloudCompare software is used here. The following essential set of inputs are required for exploiting the plugin; select the necessary video frames, zoom function, frame rate, duration of video. Video duration will be set at 12 seconds at 30 fps with 2X zoom.
- The smooth trajectory feature of the plugin would be turned off. This is due to the fact that the exact path used for the video sequence will vary depending on the content type. The degree of apparent motion between frames is always kept the same for all stimuli videos.
- The frames will be exported to a specified folder, so that they can be processed together using the FFmpeg tool. The HEVC encoder present in the FFmpeg software is used for lossless compression of the frames. The rate factor for compression will be fixed at 17. Finally, the HEVC compressed frames are combined to produce an animated video of the point cloud of duration 12 seconds.
D.3 Viewing Conditions

- Viewing conditions should follow ITU-R Recommendation BT.500.13 [BT50013].
- The MPV video player will be used for displaying the videos over a 30 inch flat display with a resolution of 4096 x 2160.
- The displays used in the subjective testing should have anti-aliasing disabled. Additionally, the background of the environment will be set to black and the colour of the point cloud will be set to white. Ambient light sources will be provided in the subjective test environment and spotlight sources would be avoided.
- The subjects will use a mouse attached with the display to provide their score.

D.4 Subjective Evaluation Protocol

- The observers will be screened for visual acuity and colour vision (using for example Snellen chart and Ishihara chart tests respectively)
- A brief description of the evaluation session will be provided to the observer, either in written form or orally.
- Prior to or at the beginning of each evaluation session, test contents will be shown to the observer to acquaint him/her with the type and magnitude of degradations that will be present in the “real” contents to be evaluated next.
- 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.

D.5 Analysis of Results

- Outlier detection algorithm based on ITU-R Recommendation BT.500-14 [BT50014] 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.

Firstly, kurtosis $B_i$ and standard deviation $s_i$ is calculated according to the Equation (D.1) and (D.2), for all video sequences $i \in \{1, n\}$, where $x_{i,j}$ are grades from all observers $j \in \{1, m\}$ for video sequence $i$ and $\bar{x}_j$ is the mean value of all grades for video sequence $i$. 
ISO/IEC JTC1/SC29/WG1 N100170
95th Meeting – Online – 25-29 April 2022

\[ \beta_i = \frac{1}{m} \sum_{j=1}^{m} (x_{i,j} - \bar{x}_i)^4 \left( \frac{1}{m} \sum_{j=1}^{m} (x_{i,j} - \bar{x}_i)^2 \right)^{-1} \]  
(D.1)

\[ s_i = \frac{1}{\sqrt{m-1}} \sum_{j=1}^{m} (x_{i,j} - \bar{x}_i)^2 \]  
(D.2)

Afterwards, a screening rejection algorithm is applied, as described in (D.3).

for every video sequence \( i \in \{1, n\} \)
for every observer \( j \in \{1, m\} \)

if \( 2 \leq \beta_i \leq 4 \)

\[ \text{if } x_{i,j} \geq \bar{x}_i + 2s_i \text{ then } P_{i,j} = P_{i,j} + 1 \]

\[ \text{if } x_{i,j} \leq \bar{x}_i - 2s_i \text{ then } Q_{i,j} = Q_{i,j} + 1 \]

else

\[ \text{if } x_{i,j} \geq \bar{x}_i + \sqrt{20}s_i \text{ then } P_{i,j} = P_{i,j} + 1 \]

\[ \text{if } x_{i,j} \leq \bar{x}_i - \sqrt{20}s_i \text{ then } Q_{i,j} = Q_{i,j} + 1 \]

for every observer \( j \in \{1, m\} \)

\[ \text{if } \frac{P_{i,j} + Q_{i,j}}{n} > 0.05 \text{ and } \frac{P_{i,j} + Q_{i,j}}{P_{i,j} - Q_{i,j}} < 0.03 \text{ then reject observer } j \]  
(D.3)

Finally, confidence interval can be calculated according to the Equation (D.4), where \( m \) is the number of gathered scores per video sequence after screening rejection algorithm, \( t(m-1) \) is Student’s \( t \) inverse cumulative distribution function (defined for the 95% confidence interval, two-tailed test) with \( m-1 \) degrees of freedom, and \( s_i \) is standard deviation for all gathered scores for video sequence \( i \).

\[ CI_i = t(m - 1) \frac{s_i}{\sqrt{m}} \]  
(D.4)