Quality Evaluation of Machine Learning-based Point Cloud Coding Solutions

Joao Prazeres
joao.prazeres@ubi.pt
Universidade da Beira Interior and
Instituto de Telecomunicacoes
Covilhã, Portugal

Rafael Rodrigues
rafael.rodrigues@ubi.pt
Universidade da Beira Interior and
Instituto de Telecomunicacoes
Covilhã, Portugal

Manuela Pereira
mpereira@di.ubi.pt
Universidade da Beira Interior and
Instituto de Telecomunicacoes
Covilhã, Portugal

Antonio M. G. Pinheiro
pinheiro@ubi.pt
Universidade da Beira Interior and
Instituto de Telecomunicacoes
Covilhã, Portugal

ABSTRACT

In this paper, a quality evaluation of three point cloud coding solutions based on machine learning technology is presented, notably, ADLPCC, PCC_GEO_CNN, and PCGC, as well as LUT_SR, which uses multi-resolution Look-Up Tables. Moreover, the MPEG G-PCC was used as an anchor. A set of six point clouds, representing both landscapes and objects were coded using the five encoders at different bit rates, and a subjective test, where the distorted and reference point clouds were rotated in a video sequence side by side, is carried out to assess their performance. Furthermore, the performance of point cloud objective quality metrics that usually provide a good representation of the coded content is analyzed against the subjective evaluation results. The obtained results suggest that some of these metrics fail to provide a good representation of the perceived quality, and thus are not suitable to evaluate some distortions created by machine learning-based solutions. A comparison between the analyzed metrics and the type of represented scene or codec is also presented.

CCS CONCEPTS

• **HCI design and evaluation methods**; • **Visualization design and evaluation methods**;

KEYWORDS

Point Clouds, Machine Learning, Quality evaluation, Coding

ACM Reference Format:
Joao Prazeres, Rafael Rodrigues, Manuela Pereira, and Antonio M. G. Pinheiro. 2022. Quality Evaluation of Machine Learning-based Point Cloud Coding Solutions. In *Proceedings of the 1st International Workshop on Advances in Point Cloud Compression, Processing and Analysis (APCCPA '22), October 14, 2022, Lisboa, Portugal.* ACM, New York, NY, USA, 9 pages. https://doi.org/10.1145/3552457.3555730

1 INTRODUCTION

In the modern world, the need for 3D data formats is increasing for multiple applications notably virtual, augmented and mixed reality, computer graphics, gaming, 3D printing, construction, manufacturing, robotics, automation, medical applications, cultural heritage, remote sensing and geographical information systems, and also consumer and retail. This applications might strongly benefit from reliable point cloud technology, increasing its effectiveness, and improving the quality and user experiences. The models of data representation, notably models for point clouds representation and associated quality will play an important role, as 3D content usually leads to huge amounts of information.

Point cloud technology has emerged as a popular method for data representation. It consists in a set of Cartesian coordinates (x, y, z), containing a list of attributes associated with each element, such as RGB components, reflectance, physical sensor information or normal vectors. Point clouds allow accurate representation of objects or scenes, from any viewing position or distance, thus making them a very powerful representation model, extremely useful in VR/AR scenarios, environment mapping for autonomous driving and urban and landscape mapping. An accurate point cloud of a city, building or artefact can contain several millions of points, each with one or more associated attributes. Considering this, point cloud compression solutions are needed, in order to efficiently compress and decompress this type of data. MPEG developed two powerful coding solutions, notably the Video-based Point Cloud Compression (V-PCC) [53] and the Geometry-based Point Cloud Compression (G-PCC) [30]. A quality study of these two codecs is provided in [35]. Google developed DRACO [14] for meshes and point cloud representation. Point clouds are coded without losses, although Draco has means to point cloud resolution control, and it can not compete with the MPEG codecs in terms of quality vs bit rate [37].

In recent years, machine learning-based coding solutions have emerged as effective models to encode point clouds [15–17, 39], after revealing to be high performing architectures for image coding.

Multiscale Point Cloud Geometry Compression was proposed in [51, 52] using a Minkowski Engine [10] for sparse convolutions.

APCCPA '22, October 14, 2022, Lisboa, Portugal.

ACM ISBN 978-1-4503-9491-8/22/10…$15.00
https://doi.org/10.1145/3552457.3555730

In [40, 41], the Deep Point Cloud Geometry Compression was presented. Adaptive Deep Learning Point Cloud Compression was presented in [17] and LUT_SR was proposed in [12], as a post processing step for G-PCC. In [43] a point cloud lossy attribute auto encoder is proposed, directly encoding and decoding attributes with the help of geometry.

Up to our knowledge, the quality evaluation of these coding solutions is based on objective metrics. These metrics are even likely to be used in the cost functions used for the training of the coding solution. Moreover, the type of errors caused by the machine learning solutions tends to be quite different of the caused by the common codecs, as typically machine learning based codecs tend to create holes in the point clouds surface. Because of that, a suitable subjective evaluation that compares different machine learning based coding solutions of point clouds is of upmost importance, as it will provide a reliable comparison between different solutions, and will provide a reliable benchmarking for different point cloud quality metrics.

In this work, a quality evaluation of three machine learning-based coding solutions is presented, notably, Adaptive Deep Learning Point Cloud Coding (ADLPCC) [17], Multiscale Point Cloud Geometry Compression (PCGC) [51], Deep Point Cloud Geometry Compression (PCC_GEO_CNN) [40]. Another recent solution, which uses Look-Up Tables built on geometric similarities across scales to resolve low resolution point clouds (LUT_SR) [12] was evaluated. Finally, the MPEG G-PCC [30] was used as benchmarking anchor.

Unlike most of previous studies that only study the quality of small objects, point cloud representatives of landscapes were also included in this paper. Furthermore, the current evaluation maps the color attributes in the coded point cloud. This is needed because the tested codecs only encode geometry. However, the texture created by color attributes plays an important role in the perceived quality.

Point cloud quality evaluation methodologies were several times studied. In [6] geometry only point clouds are considered and quality models are established. Compression artifacts using prior encoding schemes are evaluated in [11, 24, 34]. Current efforts account for a wider range of high-performing codecs, such as the ones reported in [7, 35, 44]. In [35] a subjective quality evaluation using MPEG Point Cloud codecs is presented. This work also considers a set of point cloud metrics, concluding that point to point and point to plane metrics [47] are the best performing ones, and provide a good representation of the subjective evaluation. In [6], a subjective evaluation was conducted in which point clouds were coded with octree based compression, and displayed with Screened Poisson surface reconstruction. In [11], a set of point clouds were coded using octree-pruning and a projection based method, with three levels of degradation. The models were displayed using point sizes large enough to ensure a visualisation of watertight surfaces. In [36], crowd sourcing was employed in subjective evaluation. The participants were given the option of downloading the subjective experiment content, or to access an online server and conduct the evaluation on a web browser. The two types of subjective evaluation reveal a very high level of statistical similarity. In [3], a subjective evaluation using AR is proposed, whereas in [45], a VR environment is used to evaluate point clouds. In [5], a point cloud toolbox was created, in order to aid subjective testing in such environment. In [31], users represent by 3D point clouds were able to interact with a virtual room. In [54], different resolutions and noise types were considered in a VR subjective evaluation. In [22], an evaluation on point cloud denoising algorithms is proposed. In [38], a subjective quality evaluation is conducted in a 3D environment. The results were compared to a previous study using 2D displays, and revealed that there were no statistical differences between the results. Additionally, objective quality evaluation aims to propose algorithms which can accurately predict the visual quality of content representations. Having access to algorithms that can accurately predict the quality of coded contents it's incredibly valuable as they will allow reliable quality estimation without the need of subjective quality evaluations or to easily setup codecs for improved quality of experience. Additionally, the benchmarking of these solutions is facilitated by using the best objective quality metrics, replacing the need to conduct subjective quality evaluations. In [29], a database for point cloud quality assessment is proposed together with a subjective evaluation.

Objective quality metrics can be divided in image-based and model based [27]. The first exploits high-performing solutions, applied afterwards on the selected representative views of the model, while the second one relies on geometric error, curvature or statistics measures [26], among others. The most common model-based approaches mainly assess geometry and rely on Euclidean distances, or projected errors along normal vectors [48]. In [1], an algorithm based on local surface approximations was proposed and in [2], an algorithm based on geometry normals, curvature statistics and color was introduced. In [32], color errors based on MSE and PSNR are applied on either RGB or YCbCr color space. In [50], histograms, representing color statistics are used to predict texture distortion of point cloud contents. A broad study of objective quality metrics was conducted in [28]

In the following section, a short description of the tested codecs is presented. In section 3, data preparation is described. The subjective and objective quality evaluation are described in section 4 and 5, respectively, along with a discussion of the obtained results.

2 CODEC DESCRIPTION

In this section, the codecs considered for this study are shortly described.

2.1 Adaptive Deep Learning Point Cloud Coding

The Adaptive Deep Learning Point Cloud Coding (ADLPCC) [17] initialially creates a partition of the point cloud into regular-sized 3D blocks, which are encoded and reconstructed separately by several models. The common architecture of the encoding models consists of an autoencoder (AE) and a variational autoencoder (VAE), each with three convolutional layers for encoding and three for reconstruction, with sigmoid and ReLU activations, respectively. The resulting latent representations of the AE and VAE are both entropy coded.

Each coding model is evaluated on a given block using objective PC distortion metrics and bit rate measurements. The point cloud decoding uses the decoding counterparts of the selected model for a given block, before being reconverted into 3D coordinates and merging. ADLPCC uses a loss function that takes into account both the distortion (D) and the block coding rate (R), according to:

$$J = D + \lambda R \tag{1}$$

Varying the hyperparameter λ allows to obtain different rate-distortion trade-offs. The block distortion in measured using a focal binary cross-entropy function (Eq. 2), which incorporates two training hyperparameters α and γ to adapt to block sparsity characteristics.

$$FL(v, u) = \begin{cases} -\alpha(1-v)\gamma \log(v) & u = 1 \\ -(1-\alpha)v^{\gamma} \log(1-v) & u = 0 \end{cases} \tag{2}$$

In Eq. 2, u and v refer to the original voxel occupancy binary value and the reconstructed voxel probability, respectively.

2.2 Multiscale Point Cloud Geometry Compression

The Multiscale Point Cloud Geometry Compression (PCGC) model [51] performs block-wise multiresolution encoding. Each encoding module consists of two simple convolutional layers for downscaling, followed by three residual feature extraction blocks, each containing three instances of the Inception Residual Network [46]. The downsampling is performed three times, resulting in different representations (X', X", and Y). At the bottleneck, the geometry coordinates (C_Y) and feature attributes (F_Y) of the latter are encoded, using the octree codec [30] and entropy coding, respectively. The decoding branch mirrors the encoding part, with the decoded C and F components as input, and includes a hierarchical classification at each scale.

2.3 Deep Point Cloud Geometry Compression

Deep Point Cloud Geometry Compression (PCC_GEO_CNN) [40] aims at reducing the blocking effect introduced by other deep learning-based codecs, by taking the original point cloud as input. PCC_GEO_CNN learns an encoding function from three sequential convolutional layers. While the first two layers use ReLU activation, the latent representation from the third layer ($y = f_a(x)$) is quantized $\hat{y} = Q(y)$, using element-wise integer rounding. $\hat{y}$ is compressed through range coding and the Deflate algorithm, which is a combination of LZ77 and Huffman coding [20] with shape information on the reference cloud x and latent representation y added before compression.

The decoding function (f_s) mirrors f_a, with three transposed convolutional layers, all using a ReLU activation function. The last layer has only 1 filter and its output is converted into the distorted point cloud ($\tilde{x}$) using element-wise minimum, maximum and rounding functions. The global loss function is similar to Eq. 1, but with the λ parameter associated to the distortion. The distortion is, in turn, computed using the focal loss defined in Eq. 3, to compensate the larger number of empty voxels:

$$FL(p_z^t) = -\alpha_z (1 - p_z^t)^{\gamma} \log(p_z^t) \tag{3}$$

In this loss function, if the voxel z is occupied, p_z^t and α_z are defined as p_z (reconstructed voxel probability) and α, respectively. Otherwise, they are defined as $1 - p_z$ and $1 - \alpha$. The parameters α and γ are hyperparameters of the model training.

2.4 Look-Up Tables

This solution creates a hierarchical tree-like dictionary, named Look-Up Table (LUT), which maps the occupancy relationships between downsampled geometries (V_d) and its originating counterparts (V) [12]. The method performs a second downsampling step taking V_d as input geometry, and a fractional scale factor s, resulting in the parent geometry V_{d^2}. Then, the child occupancy for each parent voxel, $\sigma(v_{d^2}(k))$, may be defined and stored in the LUT. The neighborhood configuration $\varphi_M(v_{d^2}(k))$, where M defined a M^3 cude around $v_{d^2}(k)$ is also stored in the LUT. Using a fractional scale avoids dealing with irregular grids. However, the resulting downsampled geometries may then have different configurations/classifications, because each coordinate (x,y,z) in V_d could be uniparous or multiparous, according to the number of corresponding children. The LUT will have m entries, one for each possible geometry configuration, that estimates the most likely child occupancy for the neighborhood $\varphi_M(m)$:

$$\overline{\sigma}(m) = E\{\sigma(v_{d^2}(k)) \mid \varphi_M(m)\} \tag{4}$$

The upsampling stage to obtain a resolved point cloud (V_{sr}) first applies the nearest-neighbor interpolation to find all the possible child nodes of the input point cloud V_d. The resulting geometry is carved to obtain V_{sr} following the respective LUT entries, i.e., for the corresponding φ_M, to know what points to remove. Color interpolation for the resolved point cloud takes the weighted average of the adjacent neighbors, whose weights are scale-dependent.

2.5 Geometry Point Cloud Compression

The Geometry Point Cloud Compression (G-PCC) codec is one of the normalized MPEG point cloud codecs [30], and is used as an anchor in this study. Geometry encoding in G-PCC may follow one of two methods of point cloud compression, notably the octree based method, and the trisoup method, which is based on surface reconstruction using triangular primitives, after octree decomposition. For this study, only the octree method was considered as it usually presents a more stable behaviour, and the performance of both methods is not significantly different, as shown in [35].

The G-PCC loss is controlled by the positionQuantizationScale (pQs) parameter, that controls the number of divisions of the octree, from the root to each leaf node, leading to a regular downsampling of the input point clouds. Geometry is first encoded using the octree method, and then decoded to define the shape over which the color will lie. The color attributes are assigned to output points through a re-colouring step, which uses the color values of the original model. G-PCC uses one of two different approaches to encode the color information, namely RAHT [13], based on the 3D Haar transform, and Prediction-plus-Lifting [4], based on prediction of a color value from its neighbours. For this study, only the Prediction-plus-Lifting method was considered, as it was shown in [7] that users tend to prefer the Lifting codec over RAHT. It uses a QP parameter that controls the losses on the texture information. The point clouds were encoded with the parameters shown in table 1.

Table 1: G-PCC encoding parameters.

Rate	R01	R02	R03	R04	R05
QP	46	40	34	28	22
pQS	0.25	0.5	0.75	0.875	0.9375

3 DATA SELECTION AND PREPARATION

A set of six point clouds available at JPEG Pleno database[1] was selected, depicting three objects and three landscapes. The objects are the *Romanoillamp* from Univ. of Sao Paulo Database[2], the *Guanyin* from the EPFL dataset, and frame 1300 of the *Longdress* dynamic point cloud. The point clouds *Citiusp*, *IpanemaCut* and *Ramos*, from Univ. of Sao Paulo Database were selected. Figure 1 represents the chosen point clouds.

Prior to the test, the point clouds were voxelized [1, 5] by quantizing the coordinates of the models and blending the colors of points in the same voxel. A voxel depth of 10 was empirically chosen, to ensure that each point could be represented by one pixel of the 4K display used for evaluation, and used for all the point clouds in the subjective test dataset.

Regarding video preparation, several point cloud views were captured using PCLVisualizer [42], each representing a 1° rotation. A complete rotation about the vertical axis was depicted, and the full sequence was rendered at 30 fps, thus resulting in 12 second videos. These video sequences were created with FFMPEG [3], using the H.264 codec[18].

The Constant Rate Factor (CRF) and q parameters were set to 0, so that no compression was applied. Furthermore, to prevent any RGB to YUV colorspace conversion the `libx264rgb` option was also used. In some cases, the point size was changed to provide an improved visual representation (Table 2), maintaining surfaces continuous as much as possible. If transparency appears in the point cloud, the subjects would likely see the opposite part of the point cloud, leading to a very bad quality perception [6, 11].

The machine learning-based codecs used in this study only encode geometry information. Thus, to prepare the distorted contents for the subjective evaluation, the color information of the reference point clouds was mapped onto the corresponding distorted point cloud using *Meshlab* [4]. Texture is very important in the definition of the perceived quality. It was also observed for the definition of perceived quality, to balance the quality of texture with the quality of the geometry. Because of that, the point clouds were encoded with G-PCC using the `lossless-geometry-lossy-atts` mode. For each rate in the subjective test (R01-R05), the QP value is set to the corresponding value in table 1, with a fixed pQs value of 1.

4 SUBJECTIVE QUALITY EVALUATION

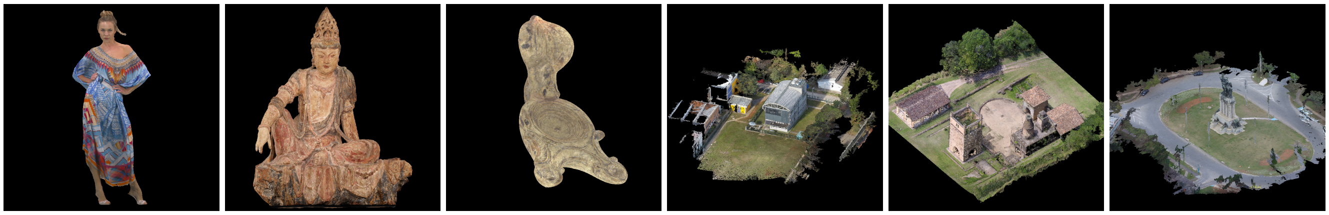
A total of 20 participants were involved in the subjective study, with ages between 18 and 58 (24.7±8.3), from which 15 were male and 5 were female. The subjective test setup used a 31.1 inch Eizo ColorEdge CG318-4K, with a full resolution of 4096x2160, and followed the specifications in [9]. During the test, each participant was shown a randomized sequence of videos containing both the quality reference and a distorted version, side by side. To avoid biases, half of the subjects were shown videos with the reference on the right and the distorted content on the left, and the other half vice-versa. A Double Stimulus Impairment Scale was used, with the subjects being prompted to evaluate the quality of the distorted point cloud, in comparison to the provided reference, according to a five-level rating scale (1 - very annoying, 2 - slightly annoying, 3 - annoying, 4 - perceptible, but not annoying, 5 - imperceptible). After the subjective test, the Mean Opinion Score (MOS) for every content was computed, by taking the average of their obtained scores. The coding bit rates, measured in bits per point (bpp), were computed as $bpp = n\_bits_d / n\_points_o$, where n_bits_d is the number of bits of a distorted content, and n_points_o is the number of points of the original point cloud.

Five quality levels were considered for each codec-content pair (Table 2), giving a total of 150 videos. Moreover, hidden reference-reference pairs for every content were also included in the test sequence, raising the final total to 156 videos. Distorted versions of the same content were never shown back to back. Before the proper subjective test, subjects were shown a training sequence with eight videos, to become familiarized with the distortion artifacts typically created by the codecs. These included four degradation levels for two different point clouds, namely *Airplane* (after conversion from a mesh) from the PointNet Database, and *Villalobospark*, both from the University of São Paulo dataset.

Figure 2 shows the MOS obtained in the subjective tests plotted against the respective coding bit rates (bpp), for each point cloud in the test dataset. Although the opinion scores do not have a gaussian distribution, the 95% Confidence Interval (CI) was computed, assuming a Student's t-distribution. The horizontal green line at the top of each plot refers to the MOS for the hidden references, whereas the green bar around it represents its 95% CI. The vertical black bar at the right side of each plot, represents the lossless encoding with G-PCC. This was computed to assure that the tested bit rates were not greater than the lossless bit rate of G-PCC. It should be noted here that G-PCC and LUT_SR were tested for similar bit rates, allowing a more direct comparison. For learning-based codecs, i.e., ADLPCC, PCC_GEO_CNN, and PCGC, the resulting bit rates are highly dependent on their training, thus the same was not possible. Nevertheless, these are directly comparable within their selected range of bit rates, and are simultaneously comparable to the higher bit rates of G-PCC and LUT_SR.

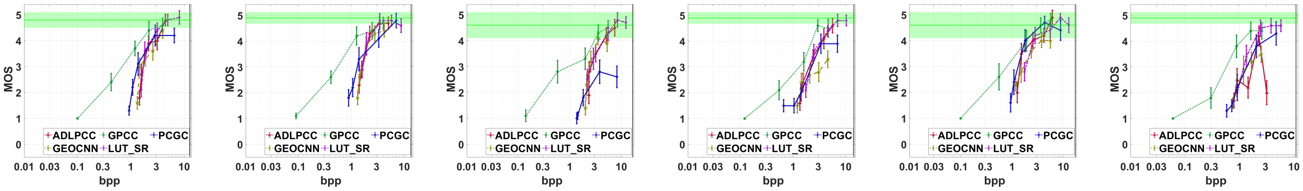
None of the learning-based codecs, was able to reach the performance of the anchor G-PCC. Globally, all three codecs showed a very similar performance, regardless of the content, with only a few exceptions. PCGC only reaches a MOS similar to the reference for the *IpanemaCut* (R04 and R05) and the *Guanyin* (R05) point clouds. However, it seems to have a slighty better performance for low bit rates than ADLPCC and PCC_GEO_CNN for some content, namely *Longdress*, *Guanyin*, and *IpanemaCut*. The *Romanoillamp* point cloud is an outlier to the general behavior. PCGC performed quite poorly for the *Romanoillamp* point cloud, as the MOS never reached 3 for any bit rate, which may be related to a lack of suitable data in the training set. Excerpts of the resulting coded point cloud may be seen in Figure 6. In the case of ADLPCC, the *Ramos* point cloud even reveals a strange behavior, where the higher bit rate

[1] http://plenodb.jpeg.org/pc/8ilabs
[2] http://uspaulopc.di.ubi.pt
[3] https://ffmpeg.org/
[4] https://www.meshlab.net/



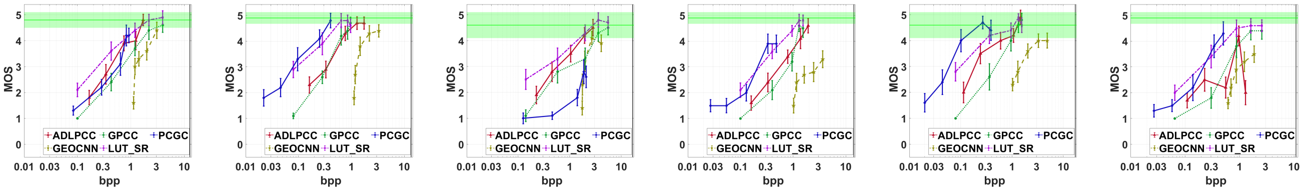
(a) *Longdress* (b) *Guanyin* (c) *Romanoillamp* (d) *Citiusp* (e) *IpanemaCut* (f) *Ramos*

Figure 1: Point Cloud test set.



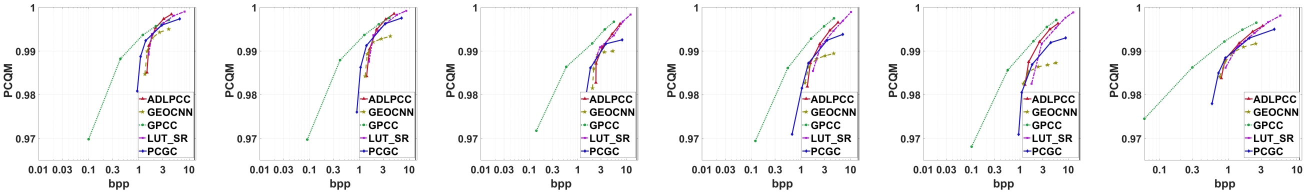
(a) *Longdress* (b) *Guanyin* (c) *Romanoillamp* (d) *Citiusp* (e) *IpanemaCut* (f) *Ramos*

Figure 2: MOS vs bpp with 95% confidence interval considering both texture (encoded with G-PCC) and geometry.



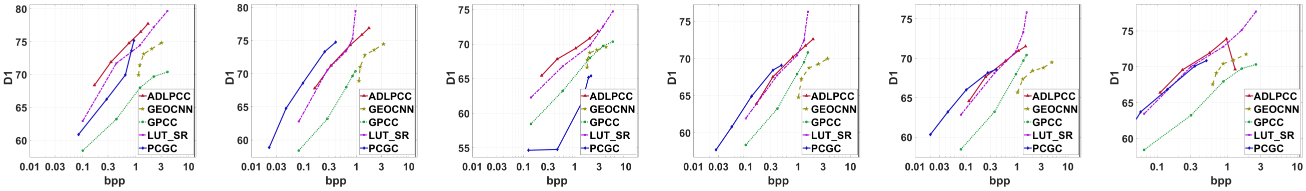
(a) *Longdress* (b) *Guanyin* (c) *Romanoillamp* (d) *Citiusp* (e) *IpanemaCut* (f) *Ramos*

Figure 3: MOS vs bpp with 95% confidence interval considering the geometry information only.



(a) *Longdress* (b) *Guanyin* (c) *Romanoillamp* (d) *Citiusp* (e) *IpanemaCut* (f) *Ramos*

Figure 4: PCQM vs bpp.



(a) *Longdress* (b) *Guanyin* (c) *Romanoillamp* (d) *Citiusp* (e) *IpanemaCut* (f) *Ramos*

Figure 5: D1 vs bpp.

Table 2: Point size of each point cloud for visualization in the subjective test.

<table>
<tr><th></th><th colspan="5">ADLPCC</th><th colspan="5">PCC_GEO_CNN</th><th colspan="5">LUT_SR</th><th colspan="5">G-PCC</th><th colspan="5">PCGC</th></tr>
<tr><th>Content</th><th>R01</th><th>R02</th><th>R03</th><th>R04</th><th>R05</th><th>R01</th><th>R02</th><th>R03</th><th>R04</th><th>R05</th><th>R01</th><th>R02</th><th>R03</th><th>R04</th><th>R05</th><th>R01</th><th>R02</th><th>R03</th><th>R04</th><th>R05</th><th>R01</th><th>R02</th><th>R03</th><th>R04</th><th>R05</th></tr>
<tr><td><i>Longdress</i></td><td>7</td><td>4</td><td colspan="3">3</td><td>8</td><td>6</td><td>5</td><td colspan="2">3</td><td colspan="5">3</td><td>8</td><td>4</td><td colspan="3">3</td><td>12</td><td>10</td><td>9</td><td colspan="2">3</td></tr>
<tr><td><i>Guanyin</i></td><td>7</td><td>4</td><td colspan="3">3</td><td>8</td><td>6</td><td colspan="2">5</td><td>3</td><td colspan="5">3</td><td>8</td><td>4</td><td colspan="3">3</td><td>12</td><td>10</td><td>9</td><td colspan="2">3</td></tr>
<tr><td><i>Romanoillamp</i></td><td>15</td><td colspan="2">6</td><td colspan="2">5</td><td>10</td><td colspan="2">8</td><td>7</td><td>6</td><td colspan="5">5</td><td>10</td><td>6</td><td colspan="3">5</td><td>15</td><td>14</td><td>13</td><td>8</td><td>7</td></tr>
<tr><td><i>Citiusp</i></td><td colspan="5">3</td><td colspan="5">3</td><td colspan="5">3</td><td>12</td><td>6</td><td>5</td><td colspan="2">3</td><td colspan="5">3</td></tr>
<tr><td><i>IpanemaCut</i></td><td colspan="5">3</td><td colspan="5">3</td><td colspan="5">3</td><td>12</td><td>6</td><td>5</td><td colspan="2">3</td><td colspan="5">3</td></tr>
<tr><td><i>Ramos</i></td><td colspan="5">3</td><td colspan="5">3</td><td colspan="5">3</td><td>12</td><td>6</td><td>4</td><td colspan="2">3</td><td colspan="5">3</td></tr>
</table>

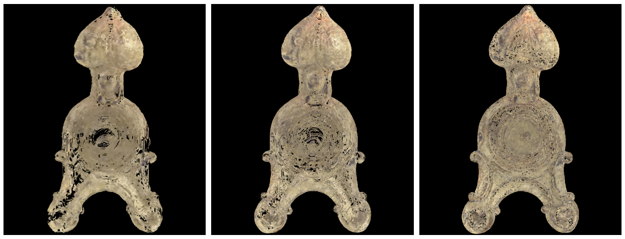
results in very low performance. The reason can be observed in Figure 7, where a significant part of the point cloud disappeared. This might also be caused by a lack of suitable training data.

Fig. 3 shows the plots of the MOS obtained as a function of the bit rate of the geometry only, without the texture influence. This plots are important because the tested machine learning-based codecs only encode the geometry.

Apart the PCC_GEO_CNN codec, in general all the encoders lead to a better MOS than the anchor G-PCC, when considering the geometry only. Their performance only becomes worst when the texture is added. In particular, the codec LUT_SR successfully improved the performance of G-PCC on geometry, but that improvement is lost when the texture is added.

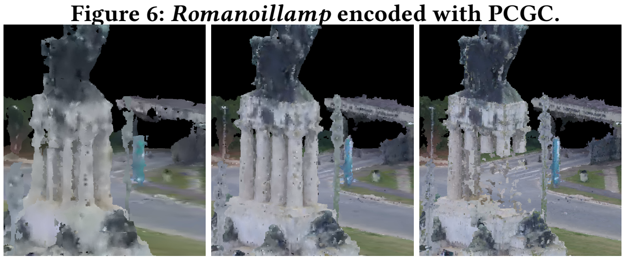
Another unusual behaviour was observed for the higher bit rate of ADLPCC (R05), which yielded a very low perceived quality for the *Ramos* point cloud, following a MOS above 4 with R04. This might also be caused by a lack of appropriate training data. An excerpt of the resulting coded point may be seen in Figure 7. ADLPCC also revealed some blocking artifacts, due to its initial point cloud partition, that may have further influenced its scores.

As shown in Table 2, for the content representing landscapes, only G-PCC required as increase of the point size for the lower bit rates. All other codecs kept a good integrity of the surfaces in visualization using the standard point size, without a major decay in the experienced quality. In the case of content representing objects, most codecs required adjustment, especially in lower bit rates, with the exception of LUT_SR. Fig. 6 and 8 represent an example for PCGC and PCC_GEO_CNN.



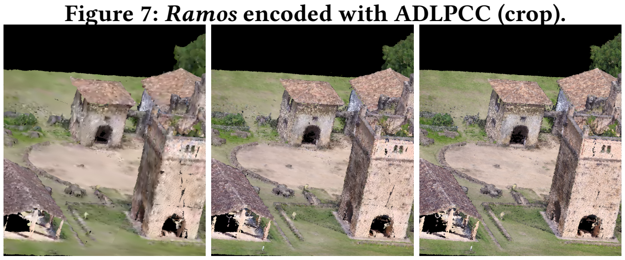
(a) *R01* **(b) *R03*** **(c) R05**

Figure 6: *Romanoillamp* encoded with PCGC.



(a) R01 **(b) R03** **(c) *R05***

Figure 7: *Ramos* encoded with ADLPCC (crop).



(a) R01 **(b) R03** **(c) *R05***

Figure 8: *IpanemaCut* encoded with PCC_GEO_CNN (crop).

5 OBJECTIVE QUALITY EVALUATION

Subjective quality assessment provides a ground truth for the validation of objective quality metrics, in the presence of the distortions produced by these codecs. In this paper, the performance of a selected set of objective quality metrics are described, notably the D1 and D2 metrics [48], Point cloud Structural Similarity metric (PSSIM) using color attributes and the covariance estimator [5], Point 2 Distribution metric [23, 25], Point Cloud Metric - Reduced Reference (PCM-RR) [49], and Point Cloud Quality Metric (PCQM) [32], by correlating their predicted MOS with the subjective MOS. The metrics selection is based on past experiences.

The MOS predictions of a given metric were computed after logistic regression on the objective scores, as is commonly done when benchmarking objective metrics [19, 33]. Then, the Pearson Correlation Coefficient (PCC), the Spearman Rank Order Correlation Coefficient (SROCC), the Root-Mean Squared Error (RMSE), and the Outlier Ratio (OR) were computed to measure the correlation, as specified in [33]. The Cloud Compare Quadric Fitting with a radius of 5 [8] was used to compute the normals [21] in the cases it was needed.

Table 3 shows the correlation outcomes of each metric, either for the entire dataset (*Global*) or by type of content (*Landscapes* or *Objects*). PCQM achieved the best results by a large margin, PCC = 0.911 and SROCC = 0.912 when considering the entire test set. It was the only metric reaching correlation coefficients above 0.9, even when considering types of content separately (PCC/SROCC of 0.932/0.928, and 0.910/0.906, for landscapes and objects, respectively). PSSIM and Point 2 Distribution provided the next best

Table 3: Correlation of the objective metrics with the subjective MOS. Results under *Global* refer to the MOS for all codecs and content types, while results under *Landscapes* and *Objects* refer to each content type separately.

	Global				Landscapes				Objects			
Metric	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR
PCQM	**0.911**	**0.912**	**0.127**	**0.547**	**0.932**	**0.928**	**0.114**	**0.140**	**0.910**	**0.906**	**0.124**	**0.570**
D1 MSE PSNR	0.820	0.797	0.175	0.780	0.843	0.841	0.165	0.600	0.822	0.783	0.171	0.660
D2 MSE PSNR	0.824	0.801	0.173	0.787	0.835	0.829	0.169	0.560	0.817	0.784	0.173	0.750
Point 2 Distribution	0.853	0.847	0.160	0.673	0.864	0.851	0.155	0.520	0.850	0.849	0.159	0.570
PSSIM	0.855	0.858	0.159	0.687	0.836	0.831	0.169	0.560	0.862	0.867	0.152	0.620
PCM-RR	0.831	0.834	0.171	0.855	0.887	0.885	0.142	0.440	0.814	0.826	0.175	0.750

Table 4: Correlation of the objective metrics with the subjective MOS for each codec.

	ADLPCC				PCC_GEO_CNN				LUT_SR				G-PCC				PCGC			
Metric	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR	PCC	SROCC	RMSE	OR
PCQM	0.854	0.843	0.146	0.300	0.798	0.774	0.141	0.767	**0.950**	**0.895**	**0.075**	**0.200**	**0.984**	**0.956**	**0.063**	**0.100**	0.889	0.874	0.147	0.233
D1 MSE PSNR	0.788	0.763	0.173	0.533	0.683	0.644	0.171	0.633	0.939	0.850	0.083	0.267	**0.984**	**0.938**	**0.064**	**0.100**	0.915	0.899	0.128	0.300
D2 MSE PSNR	0.826	0.814	0.159	0.367	0.819	0.820	0.134	0.433	0.946	0.850	0.079	0.017	0.980	0.941	0.071	0.100	0.834	0.817	0.716	0.400
Point 2 Distribution	**0.894**	**0.903**	**0.126**	**0.233**	**0.925**	**0.927**	**0.088**	**0.267**	**0.950**	**0.903**	**0.075**	**0.167**	0.758	0.660	0.240	0.400	**0.933**	**0.925**	**0.114**	**0.233**
PSSIM	0.869	0.875	0.139	0.300	0.906	0.908	0.099	0.300	0.915	0.876	0.097	0.167	0.935	0.929	0.127	0.200	0.907	0.896	0.135	0.300
PCM-RR	0.753	0.784	0.185	0.533	0.719	0.707	0.163	0.533	0.807	0.803	0.142	0.433	0.967	0.938	0.091	0.133	**0.934**	**0.917**	**0.113**	**0.300**

performances, and quite similar between them, considering the entire test set, with PCC/SROCC of 0.855/0.858, and 0.853/0.847, respectively.

When considering the correlations using only point clouds of landscapes, PCM-RR shows an improvement of its performance, reaching a PCC of 0.887 and SROCC of 0.885, which makes it the second best metric for landscape point clouds. Indeed, the same occurred with D1 and D2, albeit on a smaller scale, as these performed better with point clouds of landscapes. In the case of objects, D1 and D2 (PCC of 0.822 and 0.817) yielded similar PCC values to PCM-RR (0.814). However, their rank correlation coefficient even dropped below 0.8 (SROCC of 0.783 and 0.784).

Table 4 shows the performance of the objective metrics for each codec, while Fig. 9 presents the corresponding normalized objective metric vs. normalized MOS plots. For machine learning-based codecs i.e., ADLPCC, PCC_GEO_CNN, and PCGC, Point 2 Distribution is consistently the best performing metric, with PCC/SROCC of 0.894/0.903, 0.925/0.927, and 0.933/0.925, respectively. Interestingly, PCM-RR, which had a poor performance for ADLPCC and PCC_GEO_CNN, similar to Point 2 Distribution for PCGC, obtaining PCC = 0.934 and SROCC = 0.917. PSSIM held the second best performance for ADLPCC and PCC_GEO_CNN, and also achieved a good performance for PCGC. PCQM and D1 also had reasonable performances with PCGC.

Point 2 Distribution was again the best metric for LUT_SR (PCC/SROCC = 0.950/0.903), close to PCQM (PCC/SROCC = 0.950/ 0.895). However, it was the worst performing metric for the G-PCC anchor codec, with a PCC below 0.8. As was expected, PCQM, D1 and D2 performed well with G-PCC, always reaching coefficients above 0.9.

Figure 4 and 5 show the PCQM [32] and D1 metric [47] performances, respectively, plotted against the coding bit rates, for each point cloud on the dataset. Figure 4 show the geometry plus texture bitrates, as this metric considers both geometry and color.

Figure 4 shows the GPCC codec obtains the best results. The ADLPCC seems to perform the best different between deep learning coding solutions. However, PCGC seems to perform better in the low bit rates. Moreover, PCQM does not reveal the bad behavior of the ADLPCC for *Ramos* point cloud.

Figure 5 does not use texture information. As such, the plots show the geometry bitrates only. This metric reveals different performance for the different codecs. Although this metric reveals to be quite reliable for the analysis of some individual codecs, it can not be used to compare different coding solutions.

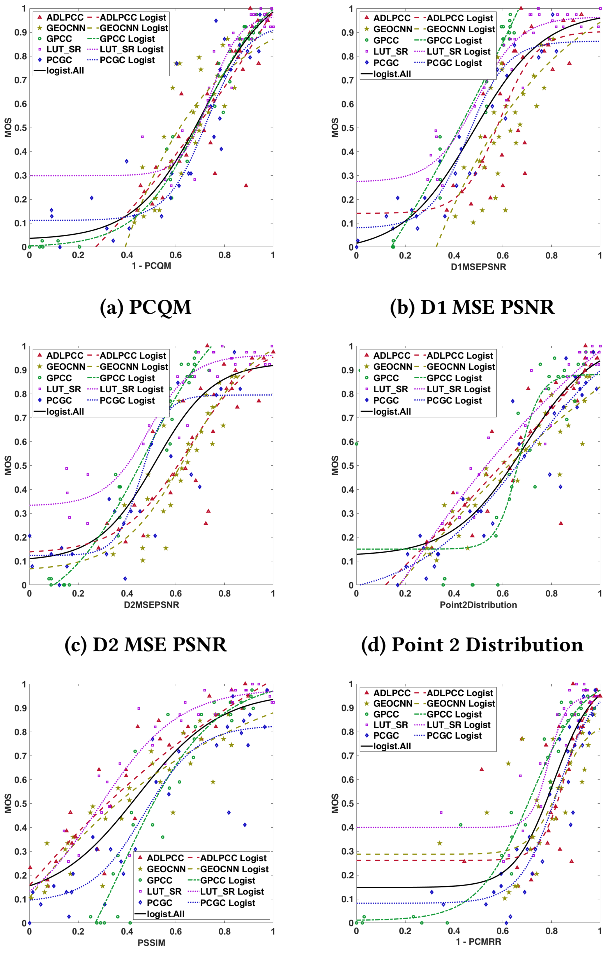
6 CONCLUSIONS

A study on the performance of static point clouds machine learning-based codecs was reported, notably, ADLPCC, PCC_GEO_CNN, and PCGC. This new generation of point cloud codecs is seen as a possible way to provide efficient compression beyond state-of-the-art codecs. Moreover, a very recent coding framework, based on self-similarities across different resolution's, i.e., LUT_SR was also tested. The G-PCC codec was also used as anchor.

From the presented subjective study, it can be concluded that most of the machine learning-based codecs have very similar performances. The PCGC seems to have some advantage in the lower bit rates, but it performs worst in the higher bit rates. Furthermore, they are more efficient than G-PCC in representing the geometry, with the exception of the PCC_GEO_CNN.

As texture is very important in any subjective evaluation, we made the option of encoding texture with G-PCC, keeping the same attribute quality parameter, as it is typically done with G-PCC. When the texture is incorporated, the anchor G-PCC has a much better performance than the other codecs. This indicates that coding solutions incorporating texture are also required for the future.

The tested machine learning-based codecs tend to create surface regions without points, which seem to produce a very strong



(a) PCQM

(b) D1 MSE PSNR

(c) D2 MSE PSNR

(d) Point 2 Distribution

(e) PSSIM

(f) PCM-RR

Figure 9: Objective metric vs. MOS plots, with logistic regression curves (global and for each codec).

bad quality perception. This was specially noted in the *Romanoillamp* point cloud, encoded with PCGC and the *Ramos* point cloud, encoded with ADLPCC.

Regarding the objective quality evaluation, it was observed that the tested metrics have some problems in predicting the perceived quality of point clouds encoded with machine learning-based codecs.

ACKNOWLEDGMENTS

This research was funded by the Portuguese FCT-Fundação para a Ciência e Tecnologia under the project UIDB/50008/2020, PLive X-0017-LX-20, and by operation Centro-01-0145-FEDER-000019 - C4 - Centro de Competencias em Cloud Computing.

REFERENCES

[1] E. Alexiou and T. Ebrahimi. 2018. Point Cloud Quality Assessment Metric Based on Angular Similarity. In *2018 IEEE International Conference on Multimedia and Expo (ICME)*.

[2] Evangelos Alexiou and Touradj Ebrahimi. 2020. Towards a Point Cloud Structural Similarity Metric. In *2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*.

[3] Evangelos Alexiou, Evgeniy Upenik, and Touradj Ebrahimi. 2017. Towards subjective quality assessment of point cloud imaging in augmented reality. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*.

[4] Evangelos Alexiou, Irene Viola, and Pablo Cesar. 2021. PointPCA: Point Cloud Objective Quality Assessment Using PCA-Based Descriptors. arXiv:2111.12663 [cs.MM]

[5] Evangelos Alexiou, Nanyang Yang, and Touradj Ebrahimi. 2020. PointXR: A Toolbox for Visualization and Subjective Evaluation of Point Clouds in Virtual Reality. *2020 Twelfth International Conference On Quality Of Multimedia Experience (Qomex)*.

[6] Evangelos Alexiou et al. 2018. Point Cloud Subjective Evaluation Methodology based on 2D Rendering. In *2018 Tenth International Conference on Quality of Multimedia Experience (QoMEX)*.

[7] E. Alexiou et al. 2019. A comprehensive study of the rate-distortion performance in MPEG point cloud compression. (2019).

[8] Pengbo Bo, Ruotian Ling, and Wenping Wang. 2012. A revisit to fitting parametric surfaces to point clouds. *Computers & Graphics* (2012). Shape Modeling International (SMI) Conference 2012.

[9] ITU-R BT.500-13. 2012. Methodology for the subjective assessment of the quality of television pictures,.

[10] Christopher Choy, JunYoung Gwak, and Silvio Savarese. 2019. 4D Spatio-Temporal ConvNets: Minkowski Convolutional Neural Networks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[11] L. A. da Silva Cruz et al. 2019. Point cloud quality evaluation: Towards a definition for test conditions. In *2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX)*.

[12] Ricardo de Queiroz, Diogo Garcia, and Tomas Borges. 2021. Fractional Super-Resolution of Voxelized Point Clouds.

[13] Ricardo L. de Queiroz and Philip A. Chou. 2016. Compression of 3D Point Clouds Using a Region-Adaptive Hierarchical Transform. *IEEE Transactions on Image Processing* (2016).

[14] Google. 2020. Draco PCC Software. Retrieved February 28, 2021 from https://github.com/google/draco

[15] André F. R. Guarda, Nuno M. M. Rodrigues, and Fernando Pereira. 2019. Point Cloud Coding: Adopting a Deep Learning-based Approach. In *2019 Picture Coding Symposium (PCS)*.

[16] André F. R. Guarda, Nuno M. M. Rodrigues, and Fernando Pereira. 2020. Deep Learning-based Point Cloud Geometry Coding with Resolution Scalability. In *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*.

[17] André F. R. Guarda, Nuno M. M. Rodrigues, and Fernando Pereira. 2021. Adaptive Deep Learning-Based Point Cloud Geometry Coding. *IEEE Journal of Selected Topics in Signal Processing* (2021).

[18] ITU H.264. 2021. Recommendation H.264.

[19] Philippe Hanhart et al. 2015. Benchmarking of objective quality metrics for HDR image quality assessment. *EURASIP Journal on Image and Video Processing* (2015).

[20] David A. Huffman. 1952. A Method for the Construction of Minimum-Redundancy Codes. *Proceedings of the IRE* 40, 9 (1952), 1098–1101. https://doi.org/10.1109/JRPROC.1952.273898

[21] ISO/IEC JTC1/SC29/WG1. 2020. JPEG Pleno PC Exploration Study 4 Results. WG1M89044, 89th Meeting.

[22] Alireza Javaheri, Catarina Brites, Fernando Pereira, and João Ascenso. 2017. Subjective and objective quality evaluation of 3D point cloud denoising algorithms. In *2017 IEEE International Conference on Multimedia Expo Workshops (ICMEW)*.

[23] Alireza Javaheri, Catarina Brites, Fernando Pereira, and João Ascenso. 2021. A Point-to-Distribution Joint Geometry and Color Metric for Point Cloud Quality Assessment.

[24] A. Javaheri et al. 2017. Subjective and objective quality evaluation of compressed point clouds. In *2017 IEEE 19th International Workshop on Multimedia Signal Processing (MMSP)*.

[25] Alireza Javaheri et all. 2020. Mahalanobis Based Point to Distribution Metric for Point Cloud Geometry Quality Evaluation. *IEEE Signal Processing Letters* (2020).

[26] Guillaume Lavoué. 2011. A Multiscale Metric for 3D Mesh Visual Quality Assessment. *Computer Graphics Forum* (2011).

[27] Guillaume Lavoué, Mohamed Chaker Larabi, and Libor Váša. 2016. On the Efficiency of Image Metrics for Evaluating the Visual Quality of 3D Models. *IEEE Transactions on Visualization and Computer Graphics* (2016).

[28] Davi Lazzarotto, Evangelos Alexiou, and Touradj Ebrahimi. 2021. Benchmarking of objective quality metrics for point cloud compression. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*. 1–6. https://doi.org/10.1109/MMSP53017.2021.9733538

[29] Qi Liu, Honglei Su, Zhengfang Duanmu, Wentao Liu, and Zhou Wang. 2022. Perceptual Quality Assessment of Colored 3D Point Clouds. *IEEE Transactions on Visualization and Computer Graphics* (2022), 1–1. https://doi.org/10.1109/TVCG.2022.3167151

[30] K. Mammou, P. A. Chou, D. Flynn, and M. Krivokuća. 2019. G-PCC codec description v2. *ISO/IEC JTC1/SC29/WG11 N18189* (Jan 2019).

[31] Rufael Mekuria, Kees Blom, and Pablo Cesar. 2017. Design, Implementation, and Evaluation of a Point Cloud Codec for Tele-Immersive Video. *IEEE Transactions on Circuits and Systems for Video Technology* (2017).

[32] Gabriel Meynet et al. 2020. PCQM: A Full-Reference Quality Metric for Colored 3D Point Clouds. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*.

[33] ITU-T P.1401. 2012. International Telecommunication Union,. In *Methods, metrics and procedures for statistical evaluation, qualification and comparison of objective quality prediction models*.

[34] S. Perry et al. 2019. Study of Subjective and Objective Quality Evaluation of 3D Point Cloud Data by the JPEG Committee. *Electronic Imaging* 2019, 10 (2019).

[35] Stuart Perry et al. 2020. Quality Evaluation Of Static Point Clouds Encoded Using MPEG Codecs. In *2020 IEEE International Conference on Image Processing (ICIP)*.

[36] Stuart Perry et al. 2021. Comparison of Remote Subjective Assessment Strategies in the Context of the JPEG Pleno Point Cloud Activity. In *2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP)*.

[37] João Prazeres, Manuela Pereira, and António M. G Pinheiro. 2022. Quality analysis of point cloud coding solutions. In *2022 Electronic Imaging Symposium*.

[38] João Prazeres, Manuela Pereira, and António M. G Pinheiro. 2022. Subjective Quality Evaluation Of Point Clouds With 3D Stereoscopic Visualization. In *IEEE International Conference on Image Processing (ICIP)*.

[39] Maurice Quach, Giuseppe Valenzise, and Frédéric Dufaux. 2019. Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression. *CoRR* (2019).

[40] Maurice Quach, Giuseppe Valenzise, and Frédéric Dufaux. 2019. Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression. In *2019 IEEE International Conference on Image Processing, ICIP 2019, Taipei, Taiwan, September 22-25, 2019*.

[41] Maurice Quach, Giuseppe Valenzise, and Frederic Dufaux. 2020. Improved Deep Point Cloud Geometry Compression. arXiv:2006.09043 [cs.CV]

[42] Radu Bogdan Rusu and Steve Cousins. 2011. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, Shanghai, China.

[43] Xihua Sheng et al. 2021. Deep-PCAC: An End-to-End Deep Lossy Compression Framework for Point Cloud Attributes. *IEEE Transactions on Multimedia* (2021).

[44] H. Su et al. 2019. Perceptual Quality Assessment of 3d Point Clouds. In *2019 IEEE International Conference on Image Processing (ICIP)*.

[45] Shishir Subramanyam, Jie Li, Irene Viola, and Pablo Cesar. 2020. Comparing the Quality of Highly Realistic Digital Humans in 3DoF and 6DoF: A Volumetric Video Case Study. In *2020 IEEE Conference on Virtual Reality and 3D User Interfaces (VR)*.

[46] Christian Szegedy, Sergey Ioffe, and Vincent Vanhoucke. 2016. Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning. *CoRR* (2016).

[47] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro. 2017. Geometric distortion metrics for point cloud compression. In *2017 IEEE International Conference on Image Processing (ICIP)*.

[48] Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, and Anthony Vetro. 2017. Geometric distortion metrics for point cloud compression. In *2017 IEEE International Conference on Image Processing (ICIP)*.

[49] Irene Viola and Pablo Cesar. 2020. A Reduced Reference Metric for Visual Quality Evaluation of Point Cloud Contents. *IEEE Signal Processing Letters* (2020).

[50] Irene Viola, Shishir Subramanyam, and Pablo Cesar. 2020. A Color-Based Objective Quality Metric for Point Cloud Contents. In *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*.

[51] Jianqiang Wang, Dandan Ding, Zhu Li, and Zhan Ma. 2020. Multiscale Point Cloud Geometry Compression.

[52] Jianqiang Wang, Hao Zhu, Zhan Ma, Tong Chen, Haojie Liu, and Qiu Shen. 2019. Learned Point Cloud Geometry Compression. *CoRR* (2019).

[53] V Zakharchenko. 2018. “Algorithm description of mpeg-pcc-tmc2”. *ISO/IEC JTC1/SC29/WG11 MPEG2018/N17767* (Jul 2018).

[54] Juan Zhang et all. 2014. A subjective quality evaluation for 3D point cloud models. In *2014 International Conference on Audio, Language and Image Processing*.