# Color and Geometry Texture Descriptors for Point-Cloud Quality Assessment

Rafael Diniz*, Pedro Garcia Freitas* and Mylène C.Q. Farias†
*Department of Computer Science, †Department of Electrical Engineering,
University of Brasília, Brazil
Email: *rafaeldiniz@aluno.unb.br, *pedrogarcia@ieee.org, †mylene@ieee.org

***Abstract*—Point Clouds (PCs) have recently been adopted as the preferred data structure for representing 3D visual contents. Examples of PC applications range from 3D representations of small objects up to large scenes, both still or dynamic in time. PC adoption triggered the development of new coding, transmission, and display methodologies that culminated in new international standards for PC compression. Along with these, in the last couple of years, novel methods have been developed for evaluating the visual quality of PC contents. This paper presents a new objective full-reference visual quality assessment metric for static PC contents, named BitDance, which uses color and geometry texture descriptors. The proposed method first extracts the statistics of color and geometry information of the reference and test PCs. Then, it compares the color and geometry statistics and combines them to estimate the perceived quality of the test PC. Using publicly available PC quality assessment datasets, we show that the proposed PC quality assessment metric performs very well when compared to state-of-the-art quality metrics. In particular, the method performs well for different types of PC datasets, including the ones where both geometry and color are not degraded with similar intensities. BitDance is a low complexity algorithm, with an optimized C++ source code that is available for download at github.com/rafael2k/bitdance-pc_metric.**

***Index Terms*—Quality Assessment, Point Clouds, Color Texture Analysis, Geometric Texture Analysis**

## I. Introduction

RECENT technology advancements have driven the production of plenoptic devices that capture and display visual contents, not only as texture information (as in 2D images) but also as 3D texture-geometric information. These devices represent the visual information using an approximation of the plenoptic illumination function, which can describe visible objects from any point in the 3D space [1]. Depending on the capturing device, this approximation can correspond to holograms, light fields, or PC imaging formats. Among these, PC formats have recently become one of the first choices to represent still and dynamic 3D visual contents. These formats consist of a collection of points in a 3D space, with their corresponding position and visual attribute information. To accurately describe a 3D scene, PCs require a large number of points, which limits their use in current multimedia applications. As a consequence, new technologies are being developed to capture, process, transmit, and display this type of media. For example, MPEG Immersive Media (MPEG-I) presented two standards for PC coding. One of them is the V-PCC (ISO/IEC 23090-5), which relies on traditional video encoding techniques, and the other one is the G-PCC (ISO/IEC 23090-9), which encodes the geometry and color information as separate entities [2].

The development of coding algorithms and transmission protocols for PC contents have triggered the development of quality assessment methods specifically designed for PC contents. Some subjective quality experiments have been performed with the goal of understanding how humans perceive immersive media in 6 Degree-of-Freedom (6DoF) environments and what are the impacts of different rendering and compression techniques on the perceived visual quality [3]. Following these studies, Point Cloud Quality Assessment (PCQA) objective metrics based on point distance measurements have been proposed [4]. These point-based PCQA metrics can be divided into the following types: Point-to-Point (Po2Point), Point-to-Plane (Po2Plane), Point-to-Surface (Po2Surface), and Plane-to-Plane (Pl2Plane). These metrics establish correspondences between the reference and (possibly) degraded PCs and measure the distances between the corresponding points/surfaces/planes to estimate the PC quality. In the case of Pl2Plane metrics [5], the angular similarity between the corresponding tangent planes of reference and distorted PC is computed to quantify their quality differences. Point-based PCQA metrics are also known as MPEG metrics because MPEG has made available their reference implementation [6]. Until recently, point-based metrics were considered the state-of-the-art in PCQA [7].

More recently, different PCQA metric approaches have been proposed. Javaheri *et al.* [8], [9] proposed metrics based on the Hausdorff and Mahalanobis distances. Viola *et al.* proposed a metric that combines color and geometry information to obtain a global quality score. Their metric takes into account the color statistics by analyzing the color histograms and the correlograms [10]. Meynet *et al.* [11] proposed a metric that also takes into consideration geometry and color features, using a logistic regression function to combine these features and produce a quality estimate. Alexiou *et al.* [12] also proposed a PCQA metric that extracts local color and geometry features. Yang *et al.* [13] uses graph-based relations among points in the PC to estimate quality. Other works include 2D projection-based approaches [14] and machine learning-based

R. Diniz and P.G. Freitas are with the Department of Computer Science, University of Brasília, Brazil..

M.C.Q. Farias is with the Department of Electrical Engineering, University of Brasília, Brazil.

Manuscript received XXX XX, XXXX; revised YYY YY, YYYY.

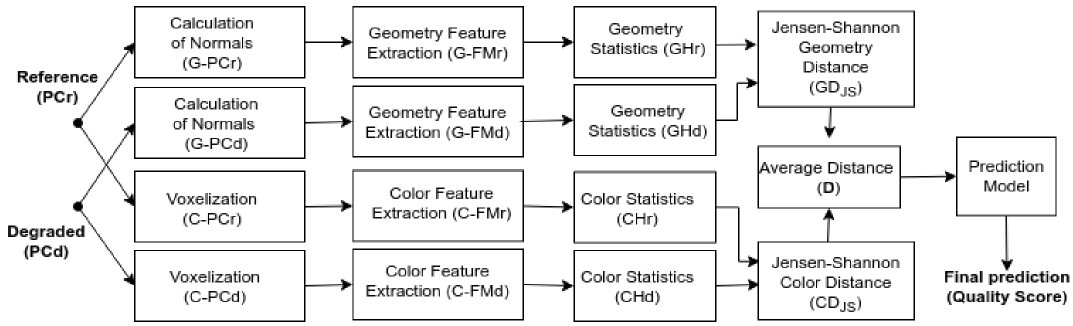


Fig. 1: Block diagram of the proposed PCQA metric.

approaches [15]. Finally, our previous work consisted of using color-based descriptors to estimate PC quality [16], [17], [18].

In this paper, we target the issue of estimating the quality of PCs degraded by generic distortion types by proposing new texture descriptors that are able to better capture color and geometry distortions. The main contribution of this work is the design of a full-reference PCQA metric that independently extracts geometry and color PC features with these proposed descriptors. The extracted color and geometry features of reference and distorted PCs are then compared to estimate the perceived quality. To test the proposed metric, we used four recent public PC quality datasets. Therefore, another contribution of this work is the performance evaluation of not only the proposed PCQA metric but also of other PCQA metrics, which is carried out over a much more diverse set of PC contents and degradations than what is currently found in the literature. The C++ source code of BitDance is available for download at github.com/rafael2k/bitdance-pc_metric.

## II. Proposed Method

Considering that PCs have at least two types of information (color and geometry) per point, the main idea of the proposed PCQA method - BitDance - is to use color and geometry descriptors to independently extract PC features from the reference and test PCs. Figure 1 shows the block diagram of the proposed method, which is divided into the following stages: (1) color feature extraction, (2) geometry feature extraction, (3) computation of feature map histograms and distances, and (4) quality model. In this section, we describe each of the stages depicted in this figure.

### A. Color Feature Extraction

The color feature extraction stage includes voxelization and descriptor application steps. Voxelization is the process of spatially discretizing the original PC points from a continuous 3D space to a discrete 3D space. The elements in the discrete 3D space grid are known as *voxels*, which can be either 'empty' or 'occupied' by a color value. The voxel size (VS) is obtained computing the Edge Size cube (ES):

$$\text{ES} = \frac{k}{S} \cdot \sum_{n=1}^{S} \left( \frac{1}{k_{nn}} \cdot \sum_{i=1}^{k_{nn}} \mathbf{d}\left(N_i(P_n), P_n\right) \right), \quad (1)$$

where $S$ is the number of points of the PC, $k$ is a constant that can take different values (an ES multiplier), $P_n$ is the $n$-th point of the PC, $N_i(P_n)$ are the coordinates of the $i$-th point nearest to $P_n$, and $k_{nn}$ is the total number of nearest neighbors. The function $\mathbf{d}\left(P_a, P_b\right)$ computes the Euclidean distance between points $P_a$ and $P_b$.

In the voxelization step, eq. 1 is used to extract the PC geometry parameters and find the voxel size. As discussed in previous works [18], [17], considering that the PC rendering process does some kind of voxelization, this voxelization step may affect the color texture descriptor and, consequently, the metric performance. Therefore, to choose the best voxelization parameters for the proposed metric, we tested different neighborhood sizes ($k_{nn}$) and multiplier values ($k$), and opted for the combination that resulted in better and more reliable results, which is $k = 6.0$ and $k_{nn} = 8$. A previous analysis of the influence of voxelization parameters on color-based PC descriptors was performed in a previous work [17].

After the voxelization step, the color texture descriptor is applied to the voxelized PC. The proposed color descriptor takes into consideration the perceptual color differences between the voxel point and its neighbors. For this, we use the CIEDE2000 (CIELab $\Delta E$ 2000) [19] color distance metric, which is more advanced than its predecessor color-difference metrics CIELAB $\Delta E$*ab and CIE944, providing perceptually uniform color distances. For each voxel $P_n$, we compute the CIEDE2000 distances between this voxel and each of its $N$-nearest neighbors voxels $P_i$. Then, based on these distances, we compute a label $L$ of $B$ bits for each PC voxel.

The label $L$ is calculated by computing, for all $N$ neighbors of the voxel $P_n$, its CIEDE2000 distance $C[i]$ to each $i$-th neighbor $P_i$ ($1 \leq i \leq N$). Initially, we set $L$ equal to zero. Then, for each of the $N$ neighbors, the following equation is applied iteratively:

$$L = \begin{cases} L \vee (1 \ll \lfloor \frac{C[i]-2.5}{2.5} \rfloor), & \text{if } 2.5 \leq C[i] < 20.0; \\ L \vee (1 \ll 7), & \text{if } C[i] \geq 20. \end{cases} \quad (2)$$

where the symbol $\vee$ is a bitwise OR and $\ll$ is a bitwise left shift. After all neighbors are analyzed, a final 8-bits (binary) label $L$ is obtained.

This process generates binary frequency values for the color distance intervals, which indicate if there is at least one neighboring voxel at this interval distance. If the label $L$ corresponding to a particular voxel is a small number, this means most neighboring voxels have similar a color to the central voxel. On the other hand, if $L$ is a large number there are neighboring voxels that generate big color differences.

In this work, we use $N = 12$ and $B = 8$ bits. It is worth pointing out that we have tested different $N$ (6 to 12) and $B$ values (8, 12, and 16 bits), but the results of these tests are not presented here for lack of space. As mentioned earlier, a previous analysis of similar parameters was performed in a previous work [17]. We chose the best combination by testing the different parameters of the PC color-based distances averaged over different variations of geometry-based distances, as detailed next.

### B. Geometry Feature Extractor

The goal of the geometry feature extractor is to extract information about the geometry of the PC points. This descriptor uses the PC normal vectors, which are vectors orthogonal to the local surface where the PC point is located. Since typical PC acquisition apparatus do not capture normal vectors, with only depth-plus-color information being generally available, the method has to compute the normal values from the eigenvectors of the local neighborhood 3D coordinates. To compute the normal vectors for each PC point, the method considers a local neighborhood with at most 16 points, which are located inside a radius of 6 times the average distance of the 8 closer neighbors. To overcome the fact that each PC point has 2 normal vectors that correctly represent a tangent plane normal, we oriented all PC normals to the direction $(0, 0, 1)$ and normalized the magnitude normal values to '1'.

For each point $P_n$ in a PC, we define the distance between $P_n$'s normal and each of the $N$-nearest neighbors $P_i$'s normals, as the distance between two 3D normal vectors:

$$G = \sqrt{\sum_{d=1}^{3}(v_{n_d} - v_{i_d})^2}$$

where $v_{n_d}$ is the normalized normal vector of point $P_n$, $v_{i_d}$ is the normal vector of a neighbor $P_i$, and $d$ represents each of the 3 dimensions $(x, y, z)$ of a normal vector. Considering that the normalized normals range from 0 to 1, the maximum possible distance between normals is 2.

After the normal distances are computed, we create a label of $B$ bits for each point. We adopted $B = 16$ bits and $N = 6$ in this work, after tests with different values (as mentioned earlier). For a given point $P_n$, its label $L$ is computed through the iteration of the distances $G[i]$ of each $i$-th nearest neighbors of $P_n$, as follows:

$$L = \begin{cases} L \vee 1, & 0.05 \leq G[i] < 0.10; \\ L \vee 1 \ll 1, & 0.1 \leq G[i] < 0.175; \\ L \vee 1 \ll 2, & 0.175 \leq G[i] < 0.275; \\ L \vee 1 \ll \lfloor \frac{G[i]-0.275}{0.125} + 3 \rfloor, & 0.275 \leq G[i] < 1.65; \\ L \vee 1 \ll 14, & 1.65 \leq G[i] < 1.80; \\ L \vee 1 \ll 15, & 1.65 \leq G[i] \leq 2.0. \end{cases} \tag{3}$$

### C. Histogram Distance Measurement

As described in previous sections, for each target point $P_n$ of the PC we compute both color and geometry features, obtaining two labels for each point in a PC. After computing the color and geometry labels associated with all PC points, histograms of the labels are computed independently for color and geometry, as follows:

$$h = \{h[l_0], h[l_1], h[l_2], h[l_3], \cdots\}, \tag{4}$$

where $h[l_j]$ corresponds to the frequency of the label $l_j$, which is computed as follows:

$$h[l_j] = \sum_{n=0}^{S-1} \delta(L(P_n), l_j), \tag{5}$$

where $S$ is the number of PC points and $\delta$ is an impulse function. The histograms are calculated for color and geometry features, both for reference ($CH_r$ and $GH_r$) and degraded ($CH_d$ and $GH_d$) PCs. Then, these histograms are compared using the Jensen-Shannon distance [20], obtaining separate color distance ($C_{JS}$) and geometry distance ($GD_{JS}$) values. To obtain a single distortion measure **D** that represents both color and geometry PC degradations, we simply average these two distances, as shown in Figure 1. This combined distance value represents how degraded $PC_d$ is when compared to the reference $PC_r$.

### D. Quality Regression Model

After obtaining the single distortion measure **D**, by simply averaging the color and geometry histogram distances, we use a regression model to estimate the perceived quality. In quality assessment methodologies, the regression model is often used to adjust the subjective quality scores provided by the different quality datasets. In this work, we use the least-squares to fit the data into the Logistic function. This model models how the human visual system perceives the different levels of distortions and, therefore, how the distance metrics are mapped into subjective quality scores [17].

## III. EXPERIMENTAL SETUP

We used four datasets, with associated subjective scores, in our simulations [14], [21], [22], [23]. Follows a description of the contents and distortions contained in these datasets.

- D1 (Torlig 2018 [14]): This dataset includes human bodies and inanimate objects. Distortions were produced using an octree-based codec, with color attributes encoded using the JPEG at different quantizer levels.
- D2 (Alexiou 2019 [21]): This dataset contains objects, full-bodies, and also a human head. The distortions were generated by the MPEG PC codecs, namely the video-based point cloud codec (V-PCC) and four variants of the geometric-based point cloud codec (G-PCC).
- D3 (Stuart 2020 [22]): This dataset contains human full-bodies and upper bodies. Distortions were created with the MPEG encoders, by the variants V-PCC and G-PCC.
- D4 (Yang 2020 [23]): The dataset contains human full-bodies, objects, and small scenes with many objects.

TABLE I: Performance of our metric proposal and other metrics the different datasets.

| Metrics | Data Sets | | | | | | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | D1 | | | D2 | | | D3 | | | D4 | | | AVG | | |
| | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE | PCC | SROCC | RMSE |
| po2point_MSE | 0.270 | 0.250 | 1.122 | 0.808 | 0.835 | 1.095 | *0.941* | 0.920 | *0.534* | 0.418 | 0.350 | 3.857 | 0.609 | 0.589 | 1.652 |
| PSNR-po2point_MSE | 0.518 | 0.484 | *0.953* | 0.494 | 0.430 | 1.352 | 0.538 | 0.549 | 1.025 | 0.470 | 0.376 | 3.832 | 0.505 | 0.460 | 1.791 |
| po2point_Haus | 0.270 | 0.215 | 1.122 | 0.627 | 0.421 | 1.282 | 0.496 | 0.446 | 1.024 | 0.261 | 0.224 | 3.900 | 0.414 | 0.327 | 1.832 |
| PSNR-po2point_Haus | 0.512 | 0.469 | 0.968 | 0.454 | 0.396 | 1.379 | 0.549 | 0.527 | 1.008 | 0.481 | 0.455 | 3.833 | 0.500 | 0.462 | 1.797 |
| Color-YCbCr_MSE | 0.383 | 0.367 | 1.039 | 0.553 | 0.571 | 1.333 | 0.755 | 0.682 | 0.921 | 0.500 | 0.512 | 3.822 | 0.548 | 0.533 | 1.779 |
| PSNR-Color-YCbCr_MSE | 0.368 | 0.337 | 1.097 | 0.536 | 0.565 | 1.351 | 0.793 | 0.801 | 0.797 | 0.504 | 0.503 | *3.805* | 0.550 | 0.552 | 1.763 |
| Color-YCbCr_Haus | 0.147 | 0.172 | 1.131 | 0.413 | 0.375 | 1.380 | 0.377 | 0.306 | 1.122 | 0.191 | 0.095 | 3.955 | 0.282 | 0.237 | 1.897 |
| PSNR-Color-YCbCr_Haus | 0.386 | 0.320 | 1.059 | 0.435 | 0.391 | 1.417 | 0.445 | 0.449 | 1.100 | 0.344 | 0.270 | 3.875 | 0.403 | 0.358 | 1.863 |
| po2plane_MSE | 0.270 | 0.275 | 1.122 | *0.845* | 0.858 | **1.031** | **0.958** | *0.945* | **0.492** | 0.432 | 0.370 | 3.859 | 0.626 | 0.612 | *1.626* |
| PSNR-po2plane_MSE | 0.484 | 0.421 | 0.984 | 0.499 | 0.495 | 1.361 | 0.542 | 0.579 | 1.021 | 0.380 | 0.390 | 3.893 | 0.476 | 0.471 | 1.815 |
| po2plane_Hausdorff | 0.270 | 0.247 | 1.122 | 0.604 | 0.427 | 1.267 | 0.586 | 0.418 | 0.981 | 0.223 | 0.188 | 3.990 | 0.421 | 0.320 | 1.840 |
| PSNR-po2plane_Haus | 0.440 | 0.408 | 1.016 | 0.428 | 0.367 | 1.394 | 0.497 | 0.463 | 1.034 | 0.464 | 0.451 | 3.836 | 0.457 | 0.422 | 1.820 |
| PCQM | 0.797 | **0.898** | 2.656 | 0.607 | *0.915* | 2.899 | 0.738 | **0.970** | 3.123 | 0.271 | 0.708 | 5.786 | 0.603 | **0.873** | 3.616 |
| PointSSIM-Color | *0.842* | 0.823 | 2.234 | **0.910** | **0.918** | 2.436 | 0.869 | 0.865 | 2.697 | *0.676* | *0.682* | 5.354 | *0.824* | 0.822 | 3.180 |
| PointSSIM-Geometry | 0.804 | 0.820 | 2.102 | 0.784 | 0.834 | 2.321 | 0.849 | 0.905 | 2.534 | 0.527 | 0.560 | 5.323 | 0.741 | 0.780 | 3.070 |
| BitDance (proposed) | **0.876** | *0.896* | **0.572** | 0.819 | 0.839 | *1.068* | 0.936 | 0.932 | 0.544 | **0.730** | **0.714** | **3.663** | **0.840** | *0.845* | **1.462** |

Seven types of distortions were used: Octree-based compression, Color Noise, Downscaling, Downscaling plus Color noise, Downscaling plus Geometry Gaussian noise, Geometry Gaussian noise, and finally Color noise plus Geometry Gaussian noise.

Notice that D2 and D3 contain only MPEG PC compression distortions, while D1 and D4 have a more diverse set of distortions, with D4 being the larger and more complete dataset in terms of different types of distortions.

For performance analysis, we used the following PCQA metrics as benchmark: the set of point-based MPEG metrics [24], PCQM [11], and PointSSIM [12]. Independent Y, Cb, and Cr color error components, used by some MPEG proposed metrics, were combined using the function proposed by Ohm *et al.* [25]. Currently, PCQM and PointSSIM are considered state-of-the-art metrics. While the MPEG metrics reference implementation and PCQM provide single distance values between reference and test PC, PointSSIM provides independent distances according to the selected feature extractor - geometry or color. We compared the predicted scores with the subjective scores provided in the datasets using Spearman's Rank Correlation Coefficient (SROCC), Pearson's Correlation Coefficient (PCC), and Root-Mean-Square Error (RMSE) as performance metrics.

## IV. Numerical Results

Table I shows the PCC, SROCC, and RMSE results obtained for the proposed BitDance metric, the MPEG PCQA metrics [6], PCQM [11], and PointSSIM [12]. The best results are shown in bold, while the second-best results are shown in italics. In the case of dataset D1, which contains two types of distortions, BitDance and PCQM perform similarly, with BitDance having the best PCC and RMSE values and PCQM having the best SROCC value. For datasets D2 and D3, which contain only MPEG PC compression distortions, PointSSIM-Color and po2plane_MSE deliver the best PCC, respectively, while PointSSIM-Color and PCQM provide the best SROCC, respectively. For these two datasets, BitDance presents a competitive performance. In the case of dataset D4, by far the largest dataset with the most diverse types of distortions, BitDance outperforms all the other metrics.

The last column of Table I shows the average results. We can see that BitDance, PointSSIM, and PCQM are the three best-performing metrics. Among the MPEG PCQA metrics, po2plane_MSE is the best-performing one, which is in agreement with a recent study by Perry *et al.* [22]. BitDance provides the best PCC and PointSSIM-Color delivers the second best. PCQM has the best SROCC, with BitDance as a close second best. Finally, BitDance provides the lowest RMSE both on average and in most datasets.

## V. Conclusions

In this paper, we presented a PCQA metric, named BitDance, that uses local and global statistics to assess the PC quality. BitDance uses two new color and geometry texture descriptors. The statistics of the outputs of these descriptors were compared using the Jensen-Shannon metric and, then, the computed geometry and color distances were combined. Finally, a logistic function was fitted to this data to produce a quality estimate. One of the side effects of the way we calculated the statistics is that the geometry and color feature extractors are invariant to the PC local topology and, as a consequence, they are rotation and scale-invariant.

BitDance was compared to other state-of-the-art PCQA metrics using four different PC quality datasets. Results showed that our strategy of using low complexity bit-shift operations provided a good and robust accuracy that outperformed all MPEG PCQA metrics and presented similar results to state-of-the-art PCQA metrics. More specifically, BitDance performed well in both types of datasets: those with only MPEG compression distortions and those with more generic distortions. The low complexity of BitDance is an important and appealing feature because PC contents have lots of data and most multimedia applications cannot handle algorithms with high computational cost. The C++ source code of BitDance is available for download at github.com/rafael2k/bitdance-pc_metric.

## REFERENCES


[1] E. H. Adelson, J. R. Bergen *et al.*, *The plenoptic function and the elements of early vision*. Vision and Modeling Group, Media Laboratory, Massachusetts Institute of ..., 1991.

[2] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "An overview of ongoing point cloud compression standardization activities: Video-based (v-pcc) and geometry-based (g-pcc)," *APSIPA Transactions on Signal and Information Processing*, vol. 9, 2020.

[3] L. Cruz, E. Dumic, E. Alexiou, J. Prazeres, R. Duarte, M. Pereira, A. Pinheiro, and T. Ebrahimi, "Point cloud quality evaluation: Towards a definition for test conditions," in *11th International Conference on Quality of Multimedia Experience (QoMEX 2019)*, 2019, pp. 1–6.

[4] F. Pereira, "Point cloud quality assessment: Reviewing objective metrics and subjective protocols," in *ISO/IEC JTC1/SC29/WG1 M78036. JPEG*. JPEG, 2018, pp. 1–8.

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

[6] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Updates and integration of evaluation metric software for pcc," *ISO/IEC JTC1/SC29/WG11 input document MPEG2017 M*, vol. 40522, 2017.

[7] A. Javaheri, C. Brites, F. M. B. Pereira, and J. M. Ascenso, "Point cloud rendering after coding: Impacts on subjective and objective quality," *IEEE Transactions on Multimedia*, 2020.

[8] A. Javaheri, C. Brites, F. Pereira, and J. Ascenso, "A generalized hausdorff distance based quality metric for point cloud geometry," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[9] ——, "Mahalanobis based point to distribution metric for point cloud geometry quality evaluation," *IEEE Signal Processing Letters*, vol. 27, pp. 1350–1354, 2020.

[10] I. Viola, S. Subramanyam, and P. Cesar, "A color-based objective quality metric for point cloud contents," in *2020 12th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[11] G. Meynet, Y. Nehme, J. Digne, and G. Lavoué, "PCQM: A full-reference quality metric for colored 3d point clouds," in *2020 12th International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[12] E. Alexiou and T. Ebrahimi, "Towards a point cloud structural similarity metric," in *2020 IEEE International Conference on Multimedia and Expo Workshops (ICMEW)*. IEEE, 2020, pp. 1–6.

[13] Q. Yang, Z. Ma, Y. Xu, Z. Li, and J. Sun, "Inferring point cloud quality via graph similarity," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2020.

[14] E. M. Torlig, E. Alexiou, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A novel methodology for quality assessment of voxelized point clouds," in *Applications of Digital Image Processing XLI*, vol. 10752. International Society for Optics and Photonics, 2018, p. 107520I.

[15] Y. Liu, Q. Yang, Y. Xu, and L. Yang, "Point cloud quality assessment: Large-scale dataset construction and learning-based no-reference approach," *arXiv preprint arXiv:2012.11895*, 2020.

[16] R. Diniz, P. G. Freitas, and M. C. Farias, "Multi-distance point cloud quality assessment," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 1–5.

[17] ——, "Local luminance patterns for point cloud quality assessment," in *2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP)*. IEEE, 2020, pp. 1–6.

[18] ——, "Towards a point cloud quality assessment model using local binary patterns," in *2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX)*. IEEE, 2020, pp. 1–6.

[19] M. R. Luo, G. Cui, and B. Rigg, "The development of the cie 2000 colour-difference formula: Ciede2000," *Color Research & Application: Endorsed by Inter-Society Color Council, The Colour Group (Great Britain), Canadian Society for Color, Color Science Association of Japan, Dutch Society for the Study of Color, The Swedish Colour Centre Foundation, Colour Society of Australia, Centre Français de la Couleur*, vol. 26, no. 5, pp. 340–350, 2001.

[20] D.-D. Shi, D. Chen, and G.-J. Pan, "Characterization of network complexity by communicability sequence entropy and associated jensen-shannon divergence," *Physical Review E*, vol. 101, no. 4, p. 042305, 2020.

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

[22] S. Perry, H. P. Cong, L. A. da Silva Cruz, J. Prazeres, M. Pereira, A. Pinheiro, E. Dumic, E. Alexiou, and T. Ebrahimi, "Quality evaluation of static point clouds encoded using mpeg codecs," in *2020 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2020, pp. 3428–3432.

[23] Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang, and J. Sun, "Predicting the perceptual quality of point cloud: A 3d-to-2d projection-based exploration," *IEEE Transactions on Multimedia*, 2020.

[24] D. Flynn, R. Julien, D. Tian, R. Mekuria, C. Jean-Claude, and V. Valentin, "Mpeg's pcc metric version 0.13.5," http://mpegx.int-evry.fr/software/MPEG/PCC/mpeg-pcc-dmetric, Mar. 2020.

[25] J.-R. Ohm, G. J. Sullivan, H. Schwarz, T. K. Tan, and T. Wiegand, "Comparison of the coding efficiency of video coding standards—including high efficiency video coding (hevc)," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1669–1684, 2012.