

RESEARCH

Open Access



# PointPCA: point cloud objective quality assessment using PCA-based descriptors

Evangelos Alexiou<sup>1\*</sup> , Xuemei Zhou<sup>1,2</sup>, Irene Viola<sup>1</sup> and Pablo Cesar<sup>1,2</sup>

\*Correspondence:  
alexiou@xiaomi.com

<sup>1</sup> Distributed and Interactive Systems, Centrum Wiskunde en Informatica, Science Park 123, Amsterdam 1098XG, The Netherlands

<sup>2</sup> Faculteit Elektrotechniek, Wiskunde en Informatica, Delft Institute of Technology, Mekelweg 5, Delft 2628CD, The Netherlands

## Abstract

Point clouds denote a prominent solution for the representation of 3D photo-realistic content in immersive applications. Similarly to other imaging modalities, quality predictions for point cloud contents are vital for a wide range of applications, enabling trade-off optimizations between data quality and data size in every processing step from acquisition to rendering. In this work, we focus on use cases that consider human end-users consuming point cloud contents and, hence, we concentrate on visual quality metrics. In particular, we propose a set of perceptually relevant descriptors based on principal component analysis (PCA) decomposition, which is applied to both geometry and texture data for full-reference point cloud quality assessment. Statistical features are derived from these descriptors to characterize local shape and appearance properties for both a reference and a distorted point cloud. The extracted statistical features are subsequently compared to provide corresponding predictions of visual quality for the distorted point cloud. As part of our method, a learning-based approach is proposed to fuse these individual predictors to a unified perceptual score. We validate the accuracy of the individual predictors, as well as the unified quality scores obtained after regression against subjectively annotated datasets, showing that our metric outperforms state-of-the-art solutions. Insights regarding design decisions are provided through exploratory studies, evaluating the performance of our metric under different parameter configurations, attribute domains, color spaces, and regression models. A software implementation of the proposed metric is made available at the following link: <https://github.com/cwi-dis/pointpca>.

**Keywords:** Point cloud, Quality metric, Objective quality assessment, Principal component analysis, Descriptors

## 1 Introduction

With the increasing popularity of extended reality technology and the adoption of depth-enhanced visual data in modern telecommunication and imaging systems, point clouds have emerged as a promising 3D content representation. However, a faithful rendition of 3D visual information using point clouds requires vast amounts of data, several orders of magnitude higher than what current transmission infrastructure can handle. Thus, reliable point cloud compression schemes are essential and have been a main focus of the Motion Picture Expert Group (MPEG) [1] and Joint Picture Expert Group (JPEG) [2] standardization bodies in the last few years. As a result of these efforts, MPEG has

crafted two standards, namely Video-based Point Cloud Compression (V-PCC) [3], and Geometry-based Point Cloud Compression (G-PCC) [4], while the JPEG Pleno [5] Learning-based Point Cloud Coding standard [6] is under development. These milestones are crucial to establish interoperability and facilitate the integration of point cloud technology in daily use cases.

Compression schemes often offer size reduction at the cost of added visual distortions. Moreover, point cloud contents might undergo signal deformations during processing, transmission, and/or rendering, which may have an additional negative effect on their perceptual quality. Therefore, there is a need for mechanisms to quantify the induced visual impairments, enabling perceptually based optimizations and ensuring the best Quality of Experience (QoE) for the end-users. Ground-truth ratings for the amount of visual impairments in a stimulus are obtained through subjective quality assessments. However, these procedures are time-consuming, costly, and essentially impractical for real-life applications. Thus, objective quality methods that can automatically predict the visual quality of distorted stimuli are required.

Two main types of characterization are commonly used to distinguish approaches for objective quality metrics for point cloud contents. One characterization comes from image and video objective quality metrics, and distinguishes between full-reference, reduced-reference, and no-reference metrics, based on their requirement for the reference content, some reference data, and no reference information at execution time, respectively. An orthogonal characterization for point cloud quality metrics is based on the domain in which the metric is computed, differentiating them as projection-based and point-based [7]. The former refers to 2D solutions, capturing geometric and textural distortions as reflected upon rendering on planar arrangements. These methods commonly adopt or extend techniques that were devised for images in the past, and they are view- and rendering-dependent [7]. Conversely, point-based counterparts operate in the 3D point cloud domain and are rendering-agnostic. Both projection- and point-based schemes could rely on either conventional or learning-based approaches [8]. However, the latter are often treated as a separate category.

Full-reference metrics are widely used in scenarios such as rate-distortion optimization for efficient compression, in which there is a need for comparing and quantifying distortions added to a pristine reference to determine the best rate allocation. Point-based solutions are rendering-agnostic and offer better generalization for cases in which the final rendering parameters are not known. Thus, in this work, we focus on a full-reference, point-based solution.

Initial attempts of full-reference point-based metrics built on simple distances between individual points, whereas more recent algorithms utilize richer features that capture local patterns of geometric and textural information. The majority of modern point-based methods make use of small sets of geometric features, often focusing on specific surface properties, with normal vectors (e.g., [9–11]) and curvatures (e.g., [11–13]) being more widely used. Textural features typically rely on statistics of luminance or lightness (e.g., [11, 13]) and occasionally chromatic components (e.g., [13]), computed over spatial neighborhoods. Geometric and textural features are often linearly combined [13], while more recently, paradigms of more advanced regression models, such as Random Forest [14, 15] and Support Vector Regression [16] are gaining ground. Employing

learning-based frameworks to combine hand-crafted features offers the advantage of interpretability, while still leveraging machine learning to effectively map predictions from the extracted features to a single quality score. Such methods have been successfully used in the field of image and video quality assessment, with VMAF being among the most renowned examples [17].

In this paper, we introduce PointPCA, an objective quality metric that makes use of hand-crafted, interpretable descriptors of geometric and textural properties, based on principal component analysis (PCA), in a learning-based framework for visual quality assessment of point clouds. Subsets of the proposed geometric descriptors have already been used for urban classification [18], semantic interpretation [19], semantic segmentation [20], contour detection [21], and, more recently, no-reference objective quality assessment [16] of point cloud data. We complement the existing literature by proposing an enriched set of PCA-based geometric and a novel set of PCA-based textural descriptors, with corresponding predictors fused through Random Forest regression to a single perceptual quality score in a full-reference design. Our results show that PointPCA achieves high performance under all tested datasets, with substantial improvements over state-of-the-art metrics. Exploratory studies are performed under different parameter configurations, color spaces, attribute-specific descriptors, and regression models to showcase the effectiveness and performance stability of our metric. Our contributions can be summarized as follows:

- We propose the use of statistical features computed from PCA-based descriptors to quantify point cloud geometric and textural distortions. The descriptors are obtained per point after applying PCA over spatial neighborhoods, and capture local geometric and textural properties, while the statistical features estimate average and dispersion trends, promoting interpretability.
- We choose the Random Forest algorithm to produce a unique perceptual quality score by fusing individual predictors obtained from the proposed statistical features in a non-linear manner. We demonstrate the effect of the selected learning-based framework through comparison to other commonly used regression models. Our results show high robustness under any non-linear method.
- We compare the performance of PointPCA to state-of-the-art metrics on a variety of datasets, showing gains in all datasets under consideration.

## 2 Related work

A brief description of point cloud objective quality assessment methods is provided below, after clustering them based on their operating principle. The interested reader may refer to [8] for a more detailed overview.

### 2.1 Point-based objective quality metrics

The point-to-point and point-to-plane [9] denote the earliest attempts for the establishment of point-based objective quality metrics. The former measures the Euclidean distance between point coordinates, while the latter relies on the projected error of distorted points across reference normal vectors. In both metrics, the mean square error (MSE) or the Hausdorff distance is applied over the individual, per-point error

values, to deliver a global degradation score. In [22], the generalized Hausdorff distance is proposed to mitigate the sensitivity of the Hausdorff distance in outlying points, by excluding a percentage of the largest individual errors. The geometric peak-signal-to-noise ratio (PSNR), defined in [23] for both metrics to account for differently scaled contents, was revised in [24] to consider the content's intrinsic or rendering resolution. The plane-to-plane metric is described in [10] and estimates the angular similarity of tangent planes, as expressed through unoriented normals. The point-to-distribution metric, introduced in [25], computes the Mahalanobis distance between a distorted point and a reference neighborhood. The PC-MSDM [12] evaluates the similarity of local curvature statistics, extracted after quadratic fitting in support regions.

Previous metrics examine only geometric distortions. A few more recent attempts employ textural-only information, albeit, the majority of metrics incorporate both geometric and textural information. Specifically, the first texture-only metric follows the point-to-point logic and measures the MSE or PSNR [26], analogously to the well-known 2D image counterpart. More sophisticated texture-only paradigms are proposed in [27], which compute histograms or correlograms of luminance and chrominance components, to characterize color distributions.

Regarding metrics that consider both geometry and texture, the point-to-distribution metric was extended to capture color degradations in [28] by additionally applying the same formula on the luminance component. The PC-MSDM was extended to PCQM [13] by incorporating local statistical measurements from luminance, chrominance, and hue in order to evaluate textural impairments. The PointSSIM [11] relies on statistical dispersion of location, normal, curvature, and luminance data. An optional pre-processing step of voxelization is proposed to enable different scaling effects and reduce intrinsic geometric resolution differences across contents. The VQA-CPC [14] computes statistics upon Euclidean distances between every sample and the arithmetic mean of the point cloud, using geometric coordinates and color values. An extension is presented in [15], namely CPC-GSCT, which involves a point cloud partition stage, before extraction of features per region.

A graph signal processing-based approach, namely GraphSIM, is described in [29] and evaluates statistical moments of color gradients on keypoints, after high-pass filtering on the pristine content's topology. A multi-scale version, namely MS-GraphSIM, is presented in [30]. In [31], local binary patterns are applied to the luminance component of neighboring points. This work is extended in [32] considering the point-to-plane distance between point clouds, and the point-to-point distance between feature maps. A variant descriptor called local luminance pattern is proposed in [33], introducing a voxelization stage. A textural descriptor to compare neighboring color values using the CIEDE2000 distance is reported in [34]. The color differences are coded as bit-based labels, which denote frequency values of pre-defined intervals. An extension is presented in [35], namely, BitDance, which incorporates bit-based labels from a geometric descriptor that relies on the comparison of neighboring normal vectors. The EPES presented in [36], relies on potential energy; that is, the energy needed to move points of a local neighborhood from an origin to their current geometric and color status. The MPED [37] also utilizes the point potential energy, which quantifies

the spatial distribution and color under certain metric space to measure isometrical distortion. The potential energy discrepancy is further extended to a multi-scale form.

The aforementioned are full-reference metrics. Fewer attempts have been reported for reduced-reference and no-reference metrics. In particular, the first reduced-reference objective quality metric, PCM<sub>RR</sub>, is described in [38] and relies on global features that are extracted from location, color, and normal data. More recently, a reduced-reference metric for point clouds encoded with V-PCC is presented in [39]. It is based on a linear model of geometry and color quantization parameters, with the model's parameters determined by a local and a global color fluctuation feature. A no-reference method, namely BQE-CVP, is proposed in [40] that combines point-based geometric features, point-based and projection-based texture degradations, and a joint geometric-color feature. In [16], the logic of using natural scene statistics for no-reference quality assessment of 2D images, is extended to 3D contents. Specifically, the authors propose statistical properties of geometric features and LAB color value distributions, to evaluate the visual quality of both point clouds and meshes.

## 2.2 Projection-based objective quality metrics

The prediction accuracy of 2D quality metrics over images obtained after projecting point clouds on the six faces of a surrounding cube, was initially examined in [41]. The influence of the number of viewpoints in denser camera arrangements and the exclusion of background pixels is explored in [42], which also proposes a weighting scheme based on user interactivity. In [43], a weighted combination of global and local features extracted from texture and depth images, is defined. The Jensen–Shannon divergence on the luminance component serves as the global feature, whereas a depth-edge map, a texture similarity map, and an estimated content complexity factor account for the local features. In [44], color and curvature values are projected on planar surfaces. Color impairments are evaluated using probabilities of local intensity differences, together with statistics of their residual intensities, and similarity values between chromatic components. Geometric distortions are assessed based on statistics of curvature residuals. A hybrid approach using both projection- and point-based algorithms is proposed in [45], namely, LP-PCQM. The point clouds are divided into non-overlapping partitions called layers, with a planarization process taking place at each layer, before applying the IW-SSIM [46] to assess geometric distortions. Color impairments are evaluated using RGB-based variants of similarity measurements defined in [13]. In [47], an image-based metric is proposed that tackles misalignment between the original and the distorted geometry. This is achieved by mapping the color of the distorted point cloud to the original geometry. The resulting and the original point clouds are then projected to the six faces of a surrounding cube, followed by cropping and padding to eliminate background pixels, before the execution of any 2D quality metric. The same process is repeated after mapping the original color to the distorted geometry, and a total quality score is obtained as a weighted average.

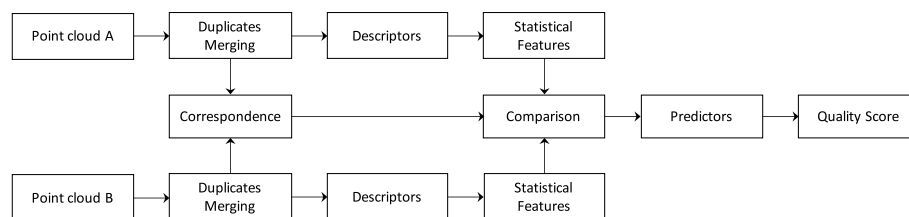
## 2.3 Learning-based objective quality metrics

In [48], Convolutional Neural Network (CNN) pre-trained for classification is evaluated in the task of no-reference point cloud quality assessment, after necessary adjustments.

Geometric distances, mean curvatures, and luminance values are packed into patches, with patch quality indexes computed using a CNN, and a global score obtained after pooling. An extension of this metric for full-reference quality assessment is presented in [49]. In [50], the use of perceptual loss is extended to point clouds, represented as voxel grids or truncated signed distances. The perceptual loss is applied to the latent space, after a simple auto-encoding architecture of convolution layers. In [51] a neural network architecture for no-reference quality assessment based on projected views is proposed, namely PQA-Net. Features are extracted after a series of CNN blocks and are shared between a distortion identifier and a quality prediction unit to obtain a final quality score. In [52], the PM-BVQA is proposed, which relies on a CNN-based joint color-geometric feature extractor that is fed with corresponding projections maps, followed by a two-stage multi-scale feature fusion step, and a spatial pooling module. In [53], point clouds are split into sub-models for geometry representation and 2D image projections for texture representation, with both modalities encoded using PointNet++ and ResNet50, respectively. Symmetric cross-modal attention is employed to fuse multi-modality quality-aware information. In [54], a graph convolution kernel (GPACConv) is introduced to capture the perturbation of structure and texture. Subsequently, the network employs a multi-task framework, with quality regression as the main task, and auxiliary tasks for predicting distortion type and degree. A coordinate normalization module is employed to enhance the stability of GPACConv results when confronted with shifts, scales, and rotations.

### 3 Description of PointPCA

The architecture of the proposed metric can be decomposed into seven stages, namely, (a) Duplicates Merging, (b) Correspondence, (c) Descriptors, (d) Statistical Features, (e) Comparison, (f) Predictors, and (g) Quality Score. A corresponding system diagram is presented in Fig. 1. The metric requires a reference during execution in order to provide a quality prediction for a point cloud under evaluation. Specifically, a correspondence between the two point clouds is obtained after merging points with identical coordinates that belong to the same point cloud. Then, 23 geometric and textural descriptors are computed per point, for both point clouds. For every descriptor, we capture local relations by applying statistical functions, leading to corresponding statistical features. Given the correspondence, 46 statistical features extracted from the reference



**Fig. 1** PointPCA architecture: both the reference (i.e., Point cloud **A**) and the point cloud under evaluation (i.e., Point cloud **B**) are passing from the Duplicates Merging, computation of Descriptors, and computation of Statistical Features stages. After Duplicates Merging, the Correspondence between the two point clouds is computed and used for the Comparison of Statistical Features. A Predictor of visual quality is obtained per Statistical Feature, and all Predictors are finally fused to a total Quality Score through learning-based regression

and the point cloud under evaluation per point, are compared. The derived error samples are pooled together, resulting in a predictor of visual quality per statistical feature. The obtained 46 predictors are finally fused by means of a regression algorithm to obtain a total quality score for the point cloud under evaluation. Below, every stage is detailed separately.

### 3.1 Duplicates merging

Within a single point cloud, points that have identical coordinates are identified and merged; that is, only one point per coordinate set is kept [9, 11]. The color of the merged point is obtained by averaging the color of respective points with the same coordinates. This offers the advantage that points with unique locations form neighborhoods to compute descriptors and statistical features, eliminating bias due to duplicated values. Moreover, redundant correspondences between a reference and a point cloud under evaluation are avoided.

### 3.2 Correspondence

Identifying matches between two sets of points is an ill-posed problem. To favor lower complexity, we use the nearest neighbor algorithm for the identification of correspondences between two point clouds, similar to the majority of existing metrics (e.g., [9–11]). For this purpose, one point cloud is set as the reference and the other as the point cloud under evaluation. Then, for every point  $\mathbf{b}_i$  that belongs to the point cloud under evaluation  $\mathcal{B}$  (i.e.,  $\mathbf{b}_i \in \mathcal{B}$ ), a matching point  $\mathbf{a}_i \in \mathcal{A}$  is identified as its nearest neighbor in terms of Euclidean distance, and is registered as its correspondence. Formally, for the point cloud under evaluation, the correspondence function is defined as  $c^{\mathcal{B},\mathcal{A}} : \mathcal{B} \rightarrow \mathcal{A}$  with  $c^{\mathcal{B},\mathcal{A}}(\mathbf{b}_i) = \mathbf{a}_i$ .

Note that, different sets of matching points are obtained when iterating over the points of  $\mathcal{B}$  to identify nearest neighbors in  $\mathcal{A}$ , with respect to starting from  $\mathcal{A}$  to find matches in  $\mathcal{B}$ ; that is when setting  $\mathcal{A}$ , or  $\mathcal{B}$  as reference, respectively. In our case, we set both the pristine and the impaired point clouds as reference, as further described in Sect. 3.6, and we use a max operation [9–11] to obtain a final prediction that is independent of the reference selection. This is commonly referred to in the literature as symmetric error [8].

### 3.3 Descriptors

A set of 15 geometric and 8 textural descriptors is defined per point, to reflect local properties of point cloud topology and appearance, respectively. The majority of those descriptors are extracted after applying PCA on spatial neighborhoods of geometric coordinates and textural values, correspondingly. Specifically, provided a query point  $\mathbf{p}_i$ , we identify a surrounding support region that belongs to the same point cloud, forming a set  $\mathbf{P}_i$  that consists of points  $\mathbf{p}_n \in \mathbf{P}_i$ . The covariance matrix  $\Sigma_i$  of this set is computed, as shown in Eq. (1):

$$\Sigma_i = \frac{1}{|\mathbf{P}_i|} \sum_{n=1}^{|\mathbf{P}_i|} (\mathbf{p}_n - \bar{\mathbf{p}}_i)(\mathbf{p}_n - \bar{\mathbf{p}}_i)^T, \quad (1)$$

with  $|\mathbf{P}_i|$  indicating the cardinality, and  $\bar{\mathbf{p}}_i$  the centroid of  $\mathbf{P}_i$ , which is given in Eq. (2):

$$\bar{\mathbf{p}}_i = \frac{1}{|\mathbf{P}_i|} \sum_{n=1}^{|\mathbf{P}_i|} \mathbf{P}_n. \tag{2}$$

Eigen-decomposition is then applied to the covariance matrix, which is symmetric and positive definite and, thus, its eigenvalues exist, are non-negative, and correspond to an orthogonal system of eigenvectors. Eigenvectors indicate directions across which the data are mostly dispersed, while eigenvalues denote the variance of the transformed data across the principal axes.

### 3.3.1 Geometric descriptors

For the computation of geometric descriptors, the coordinates of the points that belong in  $\mathbf{P}_i$  are used; hence, in Eqs. 1 and 2, we set  $\mathbf{p}_i = (x_i, y_i, z_i)^T$ . Let us assume that  $\mathbf{e}_1^g, \mathbf{e}_2^g$ , and  $\mathbf{e}_3^g$  denote the eigenvectors that correspond to the eigenvalues  $\lambda_1^g, \lambda_2^g$  and  $\lambda_3^g$ , with  $\lambda_1^g > \lambda_2^g > \lambda_3^g$  obtained after eigen-decomposition of the covariance matrix. Moreover, let us define  $\mathbf{u}_x = (1, 0, 0)^T, \mathbf{u}_y = (0, 1, 0)^T$  and  $\mathbf{u}_z = (0, 0, 1)^T$  to depict unit vectors across the  $x, y$  and  $z$  axis, respectively. Eigenvalues, eigenvectors, and unit vectors are employed to construct the proposed geometric descriptors,  $\mathbf{d}^g \in \mathbb{R}^{1 \times 15}$ , which are defined in Table 1. As can be seen, each descriptor corresponds to an interpretable shape property. Intuitively,  $d_{1-4}^g$  denote the individual (i.e.,  $d_{1-3}^g$ ) and the aggregated sum (i.e.,  $d_4^g$ ) of eigenvalues that indicate dispersion magnitudes for the points distribution across

**Table 1** Definition of descriptors

	Descriptor	Definition
Geometric	First eigenvalue	$d_1^g = \lambda_1^g$
	Second eigenvalue	$d_2^g = \lambda_2^g$
	Third eigenvalue	$d_3^g = \lambda_3^g$
	Sum of eigenvalues	$d_4^g = \sum_{v=1}^3 \lambda_v^g$
	Linearity	$d_5^g = (\lambda_1^g - \lambda_2^g) / \lambda_1^g$
	Planarity	$d_6^g = (\lambda_2^g - \lambda_3^g) / \lambda_1^g$
	Sphericity	$d_7^g = \lambda_3^g / \lambda_1^g$
	Anisotropy	$d_8^g = (\lambda_1^g - \lambda_3^g) / \lambda_1^g$
	Omnivariance	$d_9^g = \sqrt[3]{\lambda_1^g \cdot \lambda_2^g \cdot \lambda_3^g}$
	Eigenentropy	$d_{10}^g = -\sum_{v=1}^3 \lambda_v^g \cdot \ln(\lambda_v^g)$
	Surface variation	$d_{11}^g = \lambda_3^g / \sum_{v=1}^3 \lambda_v^g$
	Roughness	$d_{12}^g =  (\mathbf{p}_i - \bar{\mathbf{p}}_i) \cdot \mathbf{e}_3^g $
	Parallelity <sub>x</sub>	$d_{13}^g = 1 -  \mathbf{u}_x \cdot \mathbf{e}_3^g $
	Parallelity <sub>y</sub>	$d_{14}^g = 1 -  \mathbf{u}_y \cdot \mathbf{e}_3^g $
	Parallelity <sub>z</sub>	$d_{15}^g = 1 -  \mathbf{u}_z \cdot \mathbf{e}_3^g $
Textural	Red channel	$d_1^t = R$
	Green channel	$d_2^t = G$
	Blue channel	$d_3^t = B$
	First eigenvalue	$d_4^t = \lambda_1^t$
	Second eigenvalue	$d_5^t = \lambda_2^t$
	Third eigenvalue	$d_6^t = \lambda_3^t$
	Sum of eigenvalues	$d_7^t = \sum_{v=1}^3 \lambda_v^t$
	Eigenentropy	$d_8^t = -\sum_{v=1}^3 \lambda_v^t \cdot \ln(\lambda_v^t)$



the principal axes.  $d_{5-7}^g$  reveal behaviors of a neighborhood's points arrangement, capturing the dimensionality of the local surface.  $d_8^g$  focuses on data variation across the 1<sup>st</sup> and the 3<sup>rd</sup> principal directions.  $d_{9-11}^g$  provide an estimate of spread, uncertainty, and variation of the underlying surface, respectively, considering all principal axes.  $d_{12}^g$  quantifies the projected error of a queried point from its neighborhood's centroid, across the estimated normal vector,  $\mathbf{e}_3^g$ . Finally,  $d_{13-15}^g$  measure the projected error of  $\mathbf{e}_3^g$  across unit vectors parallel to the Cartesian coordinate system axes where a point cloud lies. In summary,  $d_{1-11}^g$  capture patterns in data dispersion,  $d_{12}^g$  local roughness, and  $d_{13-15}^g$  the direction of data dispersion.

### 3.3.2 Textural descriptors

The red green blue (RGB) color values serve as the first three descriptors of a point, noted as  $d_{1-3}^t$ . For the computation of PCA-based textural descriptors, the RGB color values of the points that belong in  $\mathbf{P}_i$  are employed; hence, we set  $\mathbf{p}_i = (R_i, G_i, B_i)^T$  in Eq. 1 and 2 and obtain the eigenvalues  $\lambda_1^t$ ,  $\lambda_2^t$  and  $\lambda_3^t$ , with  $\lambda_1^t > \lambda_2^t > \lambda_3^t$ . The individual (i.e.,  $d_{4-6}^t$ ) and the aggregated sum (i.e.,  $d_7^t$ ) of eigenvalues, as well as the eigenentropy (i.e.,  $d_8^t$ ) are computed to estimate dispersion magnitudes and uncertainty of the color distribution across one or all principal axes of a local neighborhood, respectively. The formal definition of the textural descriptors,  $\mathbf{d}^t \in \mathbb{R}^{1 \times 8}$ , is given in Table 1.

### 3.3.3 Support regions

A support region is required around every point sample in order to compute corresponding descriptors. Note that for both geometric and textural PCA-based descriptors (i.e., all excluding  $d_{1-3}^t$ ), the same support region is used and is specified based on spatial vicinity. In general, there are two alternatives widely employed to specify point cloud neighborhoods; that is, the  $k$  nearest neighbor and the range search algorithms, hereafter, noted as  $k$ -nn and  $r$ -search, respectively. The former leads to neighborhoods of arbitrary extent and a fixed population of points ( $k$ ), whereas the latter identifies the same spherical volumes (of radius  $r$ ) that enclose varying numbers of samples.

We choose the  $r$ -search algorithm to estimate descriptors. This is justified by our requirement to represent properties of the same surface areas in both the reference and the distorted stimuli. This behavior is granted by the  $r$ -search variant, as opposed to the  $k$ -nn algorithm, which is susceptible to different point densities. For example, in the presence of down-sampling, there is no difference between the size of regions identified in the pristine and the impaired point clouds using the  $r$ -search. However, when using  $k$ -nn, larger regions are considered in the impaired point cloud; thus, descriptor values represent properties of underlying surfaces of different sizes.

## 3.4 Statistical features

A set of 46 statistical features is computed per point, after applying 2 statistical functions to geometric and textural descriptor values that lie in the same neighborhood to capture inter-point local relations (e.g., [11, 13]). In particular, the mean is computed to provide a smoother estimate of a surface property (i.e., either geometric or textural), accounting for a broader region. The standard deviation is also obtained, to quantify the level of variation of a surface property in the surrounding area. Considering a query point  $\mathbf{p}_i$ , we

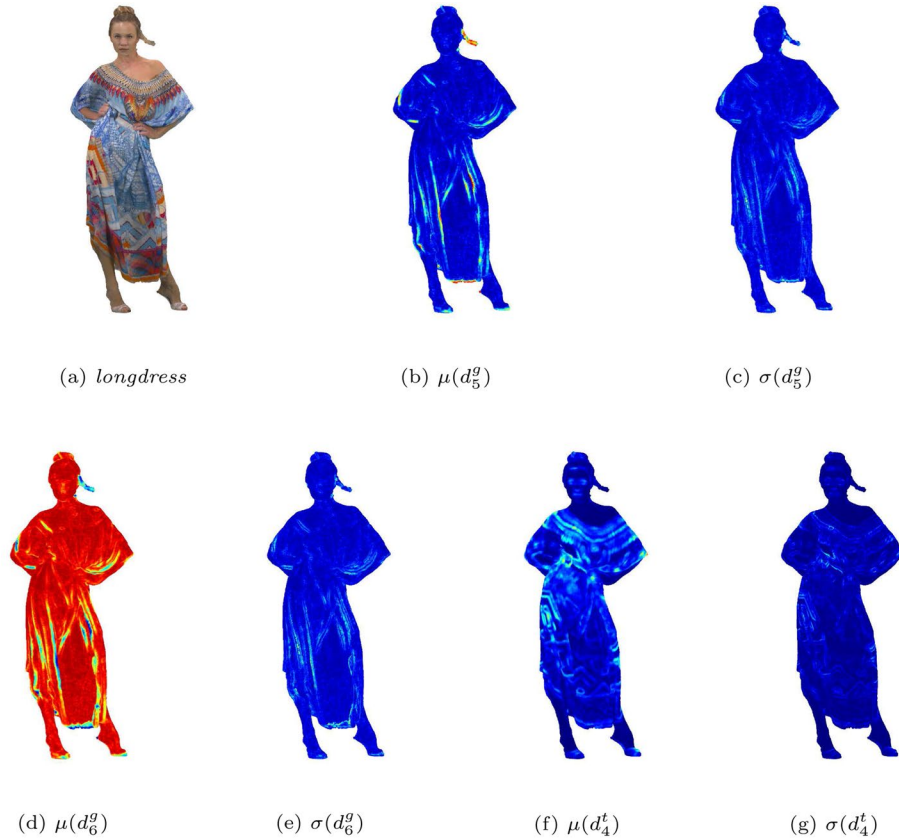
identify a support region defined as a set  $\widehat{\mathbf{P}}_i$  that consists of neighboring points  $\mathbf{p}_{\hat{n}} \in \widehat{\mathbf{P}}_i$ . The first statistical feature of point  $\mathbf{p}_i$  is computed per Eq. 3:

$$\mu_i(d_u^\omega) = \frac{1}{|\widehat{\mathbf{P}}_i|} \sum_{\hat{n}=1}^{|\widehat{\mathbf{P}}_i|} d_u^\omega(\mathbf{p}_{\hat{n}}), \tag{3}$$

where  $d_u^\omega(\mathbf{p}_{\hat{n}})$  denotes a descriptor relative to point  $\mathbf{p}_{\hat{n}}$  from either geometry ( $g$ ) or texture ( $t$ ) domain  $\omega \in \{g, t\}$ , with  $u \in \{1, 2, \dots, 15\}$  if  $\omega = g$ , and  $u \in \{1, 2, \dots, 8\}$  if  $\omega = t$ . The second statistical feature of point  $\mathbf{p}_i$  is then obtained from Eq. 4:

$$\sigma_i(d_u^\omega) = \sqrt{\frac{1}{|\widehat{\mathbf{P}}_i|} \sum_{\hat{n}=1}^{|\widehat{\mathbf{P}}_i|} (d_u^\omega(\mathbf{p}_{\hat{n}}) - \mu_i(d_u^\omega))^2}. \tag{4}$$

For point  $\mathbf{p}_i$ , we denote with  $\boldsymbol{\mu}_i \in \mathbb{R}^{1 \times 23}$  the concatenation of all  $\mu_i(d_u^\omega)$ , for all descriptors from geometry followed by texture domain; analogously, we denote with  $\boldsymbol{\sigma}_i \in \mathbb{R}^{1 \times 23}$  the concatenation of all  $\sigma_i(d_u^\omega)$ . A complete statistical features vector is given as  $\boldsymbol{\phi}_i = [\boldsymbol{\mu}_i, \boldsymbol{\sigma}_i] \in \mathbb{R}^{1 \times 46}$ . In Fig. 2, indicative visual examples of statistical features are presented.



**Fig. 2** The point cloud *longdress* (a) and its statistical features using the mean and standard deviation of linearity (b, c), planarity (d, e), and first eigenvalue on texture (f, g) descriptors. The amplitudes of statistical features are color-mapped, with red indicating higher and blue lower values. It can be noticed that the mean of linearity (b) and planarity (d) capture high- and low-frequency geometric regions, respectively. Moreover, the mean of the first eigenvalue on texture (f) highlights colorfulness. The standard deviation quantifies local dispersion, hence capturing high frequencies for all descriptors

Statistical features are able to better capture dependencies within local neighborhoods, and provide measurements that are more perceptually coherent with respect to single points. Specifically, they are well-aligned with primary characteristics of the human visual system, such as low-pass filtering and sensitivity to high-pass frequencies. Applying the mean in local regions mimics the former, whereas the standard deviation provides an estimate of the latter. Moreover, statistical features are computed per point and contain contributions from its surroundings, thus, alleviating the negative effects of an erroneous correspondence, or outlying descriptor values. That is, considering impaired stimuli that are characterized by point removal or displacement with respect to their pristine positions, errors might be introduced by the matching algorithm, or descriptor values might be poorly estimated. Hence, comparing means instead of descriptor values mitigates the error.

### 3.4.1 Support regions

We choose the  $k$ -nn algorithm to compute statistical features. We argue that, in this case, the operating principle of this approach is beneficial for revealing topological deformations. In particular, by appending neighboring samples until reaching  $k$ , we consider larger areas in a sparser impaired stimulus, and we recruit erroneous points in case of re-positioning. Thus, larger differences will be observed in comparison to corresponding measurements taken from the pristine content. In simpler terms, using  $k$ -nn allows us to penalize point sparsity and displacement.

### 3.5 Comparison

Given the correspondence function  $c^{\mathcal{B},\mathcal{A}}(\mathbf{b}_i) = \mathbf{a}_i$  defined in Sect. 3.2, the  $j^{\text{th}}$  statistical feature of point  $\mathbf{b}_i \in \mathcal{B}$ , namely  $\phi_{i,j}^{\mathcal{B}}$ , is compared to the  $j^{\text{th}}$  statistical feature of point  $\mathbf{a}_i \in \mathcal{A}$ , namely  $\phi_{i,j}^{\mathcal{A}}$  using the relative difference as in [11], per Eq. 5:

$$r_{i,j}^{\mathcal{B},\mathcal{A}} = \frac{|\phi_{i,j}^{\mathcal{A}} - \phi_{i,j}^{\mathcal{B}}|}{\max\left(|\phi_{i,j}^{\mathcal{A}}|, |\phi_{i,j}^{\mathcal{B}}|\right) + \varepsilon}, \quad (5)$$

where  $r_{i,j}^{\mathcal{B},\mathcal{A}}$  indicates the derived error sample that corresponds to  $\mathbf{b}_i$ , with  $1 \leq i \leq |\mathcal{B}|$  and  $1 \leq j \leq 46$ , while  $\varepsilon$  represents a small constant to avoid undefined operations; in this case, we use the machine rounding error for floating point numbers. This computation is repeated for all  $\mathbf{b}_i$ , and corresponding error samples  $r_{i,j}^{\mathcal{B},\mathcal{A}}$  are obtained.

### 3.6 Predictors

For every statistical feature  $j$ , the error samples of  $\mathcal{B}$  are pooled together, as shown in Eq. 6:

$$s_j^{\mathcal{B},\mathcal{A}} = \frac{1}{|\mathcal{B}|} \sum_{i=1}^{|\mathcal{B}|} r_{i,j}^{\mathcal{B},\mathcal{A}}. \quad (6)$$

The same computations are repeated after setting the point cloud  $\mathcal{B}$  as the reference, provided the correspondence function  $c^{\mathcal{A},\mathcal{B}}(\mathbf{a}_k) = \mathbf{b}_k$ , and a corresponding measurement  $s_j^{\mathcal{A},\mathcal{B}}$  is computed. Finally, for every statistical feature  $j$ , a corresponding predictor

$s_j$ , with  $1 \leq j \leq 46$ , is obtained after applying the symmetric max operation similarly to [9–11], per Eq. 7:

$$s_j = \max\left(s_j^{B,A}, s_j^{A,B}\right). \quad (7)$$

### 3.7 Quality score

Each predictor  $s_j$  provides a quality rating based on the  $j^{\text{th}}$  statistical feature. To combine all 46 predictors into a total quality score,  $q$ , any linear or non-linear regression model can be used. Machine learning-based regression models have been extensively used to tackle this problem in the domain of quality assessment. As part of our metric, we use the Random Forest algorithm. This is an ensemble learning method that can improve the prediction performance with respect to single features while limiting overfitting issues. Note that we evaluate the impact of using different regression models on the performance of our method in Sect. 6.4.

### 3.8 Complexity

The total complexity of the algorithm is dominated by the operations that require the definition of a support region using the  $r$ -search and  $k$ -NN algorithms for the computation of descriptors and statistical features, as described in Sects. 3.3 and 3.4, respectively. For a given point cloud  $\mathcal{P}$ , such operations generally have average complexity  $O(|\mathcal{P}| \log |\mathcal{P}|)$  for well-behaved cases, and  $O(|\mathcal{P}|^2)$  in the worst-case scenario. Thus, an upper bound of the complexity of the algorithm can be defined as  $O(N^2)$ , in which  $N = \max(|\mathcal{A}|, |\mathcal{B}|)$ .

## 4 Benchmarking setup

### 4.1 Selection of datasets

Three subjectively annotated data sets are used to evaluate the performance of the proposed and state-of-the-art quality metrics under consideration, namely, M-PCCD (D1) [7], SJTU (D2) [43] and WPC (D3) [55]. D1 consists of 8 colored static point clouds illustrating both human figures and inanimate objects, whose geometry and color are encoded using V-PCC and four G-PCC variants (i.e., Octree-plus-Lifting, Octree-plus-RAHT, TriSoup-plus-Lifting, and TriSoup-plus-RAHT), resulting in 232 distorted stimuli. D2 comprises 9 colored point clouds depicting both human figures and inanimate objects that are subject to octree-based compression, color noise, geometry Gaussian noise, down-scaling, and a superposition of every combination of two aforementioned degradations excluding compression, for a sum of 378 distorted stimuli. Finally, D3 contains 20 colored point clouds depicting inanimate objects, that are subject to octree-based down-sampling, a superposition of geometric and color Gaussian noise, and a superposition of geometric and color compression distortions using a TriSoup- and an Octree-based G-PCC variant, as well as V-PCC, for a total of 740 distorted stimuli.

### 4.2 Computation of performance indexes

To evaluate the performance of an objective quality metric in predicting perceptual quality, Mean Opinion Score (MOS) from subjects participating in dedicated experiments

are employed as ground truth. The metrics are typically benchmarked after applying a fitting function to map the objective scores to the subjective quality range, while also accounting for biases, non-linearities, and saturations from subjective testing. Let us define a score obtained by the execution of an objective metric as a Predicted Quality Score (PQS). A predicted MOS, denoted as P(MOS), is estimated by applying the fitting function on the [PQS, MOS] data set. In our analysis, the Recommendation ITU-T J.149 [56] is followed, using the logistic function type II. Then, the Pearson Linear Correlation Coefficient (PLCC), the Spearman Rank Order Correlation Coefficient (SROCC), and the Root Mean Square Error (RMSE) are computed between the P(MOS) and MOS to draw conclusions on the linearity, monotonicity, and accuracy of the objective quality metrics, respectively.

### 4.3 Configuration and execution of objective quality metrics

State-of-the-art objective quality metrics are employed in our performance evaluation analysis for comparison purposes. In particular, we use the point-to-point, point-to-plane [9], and color PSNR on luminance component, which are being used in the MPEG standardization activities for point cloud compression. We also use the plane-to-plane [10], the joint point-to-distribution metric [28] with logarithmic values, the BitDance [35], the PointSSIM [11] (on geometry, normal, curvature and luminance), the PCQM [13], and the MPED [37].

To compute the point-to-point and point-to-plane, the software version 0.13.5 [26] is used. For the latter, the normals are computed using a quadratic fitting with  $r$ -search and  $r = 0.01 \times B_R$ , where  $B_R$  indicates the maximum length of the bounding box of the reference point cloud. For plane-to-plane, the normals are computed based on quadratic fitting with  $r$ -search and  $r = 0.02 \times B_R$ , following literature best practices [57]. In the point-to-distribution metric, neighborhoods consisting of  $k = 31$  point samples are considered. For BitDance, we use the recommended configurations, namely,  $k = 6$  for the target voxel edge size, while the neighborhood size is set to 6/12 and the label bits to 16/8 for geometry/color histogram. For PointSSIM, the default parameters are employed, with the variance as the selected estimator of statistical dispersion, and  $k = 12$ ; for the computation of curvatures and normals, quadric fitting with  $r$ -search and  $r = 0.01 \times B_R$  was used. In PCQM, the default configurations are used. For MPED [37], the default settings are employed, with  $L$  defined as a fraction of the total number (i.e., 1/10000), and the square of  $\ell^2$  norm adopted as the distance function. For PointPCA, the PCA-based descriptors (i.e., all except  $d_{1-3}^t$ ) are estimated using the  $r$ -search with  $r = 0.008 \times B_R$ , while for the statistical features, the  $k$ -nn algorithm with  $k = 9$  is used. The Random Forest regression method is implemented using the scikit-learn python framework [58] with the default configuration, namely, MSE as a criterion for split and 100 trees. Note that results from the PSNR versions of point-to-point and point-to-plane are not reported, due to the presence of infinity values, which prevented correlation computations and fair comparison.

### 4.4 Evaluation of objective quality metrics

As part of our analysis, we evaluate the performance of each individual predictor on the datasets D1, D2, and D3; in this case, all contents of each dataset are considered.

Moreover, we evaluate the performance of PointPCA after fusing individual predictors using learning-based regression. However, such a validation requires splitting the datasets into training and testing sets. In our analysis, performance indexes are computed and provided only for the testing counterparts. In particular, PointPCA quality prediction models obtained using either Random Forest (i.e., as part of our architecture with results reported in Sect. 5.2) or other regression models (i.e., as part of our comparative analysis in Sect. 6.4), are validated both within and across datasets using the leave- $p$ -out method. Specifically, each dataset is split into two partitions that contain 80% and 20% of the contents for training and testing, respectively, with all the distorted versions of a specific content placed in one partition. For D1, D2, and D3, we use 6/2, 7/2, and 16/4 contents for training/testing, respectively. Then, a quality prediction model is trained on the training data and tested on the corresponding testing data of the same dataset, for within-dataset validation. Moreover, the same quality prediction model is tested on each of the other two (entire) datasets for cross-dataset validation. This process is repeated for all possible 80%-20% splits of each dataset, leading to 28, 36, and 4845 testing partitions and an equal number of corresponding quality prediction models for D1, D2, and D3, respectively. The average and the standard deviation of the performance indexes across all testing partitions are reported for the within-dataset validation, while only the average is reported for the cross-dataset validation.

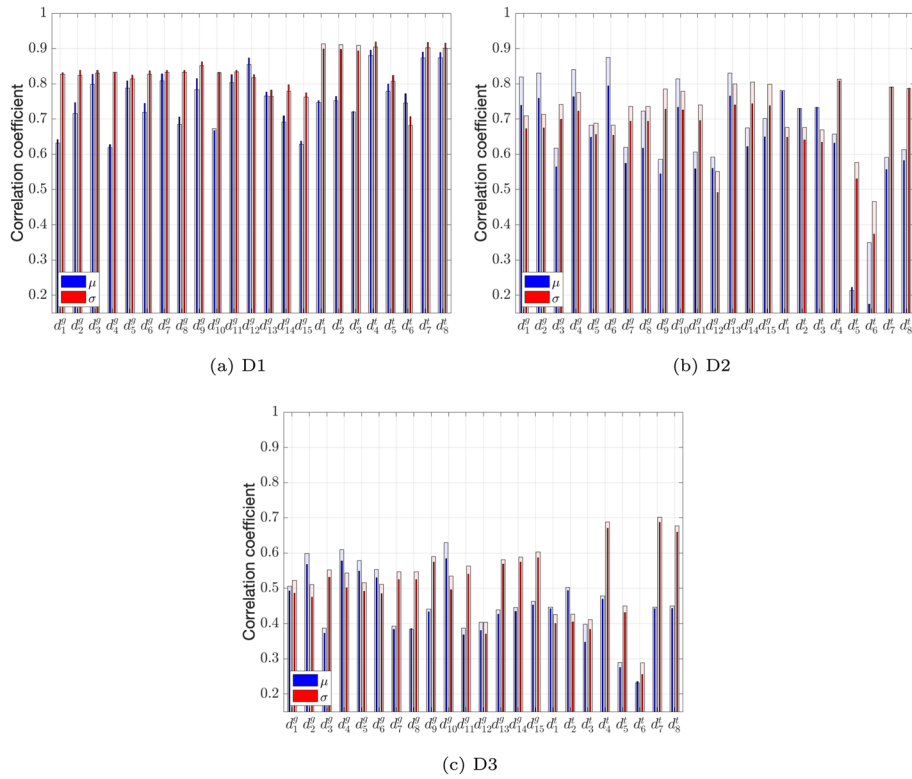
Finally, we compare PointPCA with state-of-the-art metrics. To enable a fair comparison between PointPCA quality models and non-learning-based metrics from the literature, performance indexes for the latter are computed over the same testing partitions. That is, on the same testing data obtained after applying the leave- $p$ -out method with 80%-20% splits on each dataset, separately. Then, the average and the standard deviation of every performance index are computed across all testing partitions of each dataset (i.e., 28, 36, and 4845 testing partitions for D1, D2, and D3, respectively).

## 5 Results

### 5.1 Performance evaluation of predictors

In Fig. 3, the PLCC and SROCC of every predictor are illustrated in the form of bars grouped per descriptor, against subjectively annotated datasets. It can be noticed that the prediction accuracy of the proposed predictors is reaching a different performance plateau per dataset; in particular, we observe high performance for D1 and D2, while substantially lower for D3. This can be explained by the different distortion characteristics of each dataset. Specifically, geometric-only and textural-only predictors cannot accurately capture combinations of different geometric and textural degradation levels (e.g., D3), whereas better trends are expected when the level of degradation in both geometry and texture is amplified simultaneously (e.g., D1 and D2).

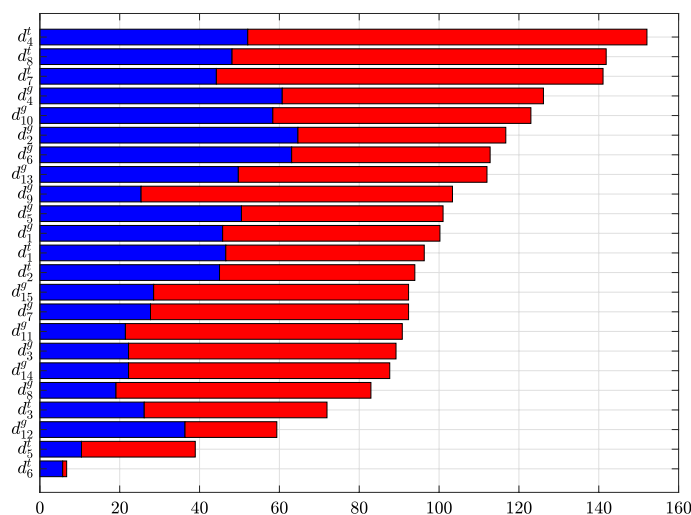
Moreover, the standard deviation is found to perform better than the mean across all datasets, showing a certain level of consistency. Specifically, for  $d_{3,7,8,9,11,14,15}^g$  and  $d_{4,5,7,8}^t$  the standard deviation performs steadily better than the mean, while the mean is superior only for  $d_{12}^g$ . For the remaining descriptors, different behaviors are observed across datasets, although the differences are limited. For instance, for  $d_1^g$ , the standard deviation exhibits higher accuracy in D1 compared to the mean, with the opposite being true for D2, while equivalent performance is observed in D3.



**Fig. 3** Benchmarking of predictors by means of PLCC (thin opaque bars) and SROCC (thick transparent bars), grouped per descriptor  $d_u^\omega$ . Each bar represents a predictor, which relies on either the  $\mu(d_u^\omega)$  or the  $\sigma(d_u^\omega)$  statistical feature, and is indicated with blue and red color, respectively

Finally, it is remarked that predictors using the textural descriptors  $d_{4,7,8}^t$  are ranked among the best places consistently across all datasets. In general, they are found to be superior to every geometric predictor in D1 and D3, while in D2 they show high predictive power, despite the fact that geometric predictors perform overall better in this dataset. The high effectiveness of textural predictors can be justified by considering that they incorporate a spatial dimension through the usage of geometric neighborhoods for the computation of descriptors and statistical features. Therefore, they not only explicitly evaluate textural distortions, but they additionally capture topological deformations in an implicit manner.

The above observations are in alignment with the results presented in Fig. 4, where the importance ranking scores of the proposed predictors are depicted. Specifically, the average ranking order of every predictor is computed across all datasets based on the average PLCC and SROCC. The average ranking order is then scaled to the range [1–100], with 1 indicating the minimum and 100 the maximum importance score, which corresponds to the lowest and highest average ranking order, respectively. Importance ranking scores are grouped and stacked per descriptor (blue corresponds to  $\mu(d_u^\omega)$  and red to  $\sigma(d_u^\omega)$  statistical feature), before being sorted in descending order, based on their aggregated sum. Thus, the final ranking scale would range between [3–199]. The results show that the predictor based on  $\sigma(d_4^t)$  achieves the highest



**Fig. 4** Importance ranking scores of predictors, computed based on their average ranking order across all datasets, stacked per descriptor  $d_U^w$ . The ranking order is determined using both PLCC and SROCC. Blue and red bars represent predictors that rely on  $\mu(d_U^w)$  and  $\sigma(d_U^w)$ , respectively

**Table 2** Performance evaluation of PointPCA over datasets D1, D2, and D3

Train	Test								
	D1			D2			D3		
	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$
D1	0.938	0.942	0.444	0.808	0.803	1.423	0.567	0.569	18.864
D2	0.824	0.836	0.769	0.932	0.907	0.859	0.683	0.678	16.727
D3	0.786	0.817	0.838	0.862	0.842	1.229	0.894	0.890	10.132

Italics indicate within-dataset results

score, with predictors based on  $\sigma(d_7^t)$  and  $\sigma(d_8^t)$  closely following. These results confirm the superiority of textural predictors based on  $d_{4,7,8}^t$ , as already noted in Fig. 3.

**5.2 Performance evaluation of PointPCA**

Table 2 shows the performance of PointPCA over the three selected datasets, for both within- and cross-dataset validation as described in Sect. 4.4. Substantial improvements are remarked when combining predictors with respect to using them singularly, as depicted in Fig. 3. In particular, significant performance boosts are observed for D3, which is the most populated dataset with the most diverse distortion types. Notable gains are also shown for D2, while smaller differences are noticed for D1.

As expected, within-dataset results generally achieve better performance with respect to cross-dataset results. Considering cross-dataset validation results, training on D1 leads to poor generalization capabilities on D2 and D3, compared to training on D3 and D2, respectively. Training on D2 leads to better generalization on D1 with respect to training on D3, while the performance on D3 remains low. These results can be explained by the intrinsic characteristics of the datasets; D1 contains only compression distortions with both human and object models, D2 additionally employs geometric and



color noise, while D3 is the most diverse in terms of distortion types containing only objects (see Sect. 4.1).

### 5.3 Comparison with the state of the art

In Table 3, we show performance results of PointPCA and existing point cloud quality metrics across the selected datasets, for comparison purposes. Specifically, we

**Table 3** Performance evaluation of state-of-the-art quality metrics

Metric	D1			D2		
	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$
Point-to-point_MSE [9]	0.884 $\pm$ 0.047	0.896 $\pm$ 0.042	0.615 $\pm$ 0.142	0.697 $\pm$ 0.138	0.612 $\pm$ 0.157	1.662 $\pm$ 0.396
Point-to-plane_MSE [9]	0.891 $\pm$ 0.033	0.901 $\pm$ 0.025	0.604 $\pm$ 0.116	0.663 $\pm$ 0.128	0.578 $\pm$ 0.155	1.762 $\pm$ 0.334
PSNR Y [26]	0.796 $\pm$ 0.140	0.798 $\pm$ 0.162	0.751 $\pm$ 0.270	0.751 $\pm$ 0.080	0.743 $\pm$ 0.083	1.561 $\pm$ 0.210
Plane-to-plane [10]	0.837 $\pm$ 0.072	0.847 $\pm$ 0.076	0.719 $\pm$ 0.161	0.836 $\pm$ 0.033	0.761 $\pm$ 0.039	1.311 $\pm$ 0.136
LOG point-to-distribution-joint [28]	0.919 $\pm$ 0.027	0.921 $\pm$ 0.024	0.526 $\pm$ 0.106	0.692 $\pm$ 0.113	0.645 $\pm$ 0.120	1.698 $\pm$ 0.301
BitDance [35]	0.850 $\pm$ 0.073	0.859 $\pm$ 0.061	0.688 $\pm$ 0.161	0.767 $\pm$ 0.074	0.748 $\pm$ 0.077	1.521 $\pm$ 0.228
PointSSIM (geometry) [11]	0.848 $\pm$ 0.060	0.841 $\pm$ 0.066	0.700 $\pm$ 0.161	0.754 $\pm$ 0.039	0.712 $\pm$ 0.032	1.573 $\pm$ 0.132
PointSSIM (normal) [11]	0.831 $\pm$ 0.082	0.831 $\pm$ 0.082	0.723 $\pm$ 0.175	0.840 $\pm$ 0.049	0.764 $\pm$ 0.059	1.294 $\pm$ 0.227
PointSSIM (curvature) [11]	0.873 $\pm$ 0.059	0.876 $\pm$ 0.050	0.640 $\pm$ 0.152	0.852 $\pm$ 0.046	0.772 $\pm$ 0.055	1.250 $\pm$ 0.213
PointSSIM (luminance) [11]	<u>0.937</u> $\pm$ 0.026	0.925 $\pm$ 0.024	<u>0.458</u> $\pm$ 0.071	0.735 $\pm$ 0.055	0.708 $\pm$ 0.070	1.621 $\pm$ 0.181
PCQM [13]	0.930 $\pm$ 0.041	<u>0.940</u> $\pm$ 0.032	0.472 $\pm$ 0.136	<u>0.878</u> $\pm$ 0.025	<u>0.862</u> $\pm$ 0.030	<u>1.145</u> $\pm$ 0.121
MPED [37]	0.857 $\pm$ 0.095	0.870 $\pm$ 0.091	0.657 $\pm$ 0.233	0.872 $\pm$ 0.056	0.856 $\pm$ 0.067	1.148 $\pm$ 0.239
PointPCA	<b>0.938</b> $\pm$ 0.039	<b>0.942</b> $\pm$ 0.042	<b>0.444</b> $\pm$ 0.127	<b>0.932</b> $\pm$ 0.025	<b>0.907</b> $\pm$ 0.043	<b>0.859</b> $\pm$ 0.166
Metric	D3					
	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$			
Point-to-point_MSE [9]	0.582 $\pm$ 0.067	0.563 $\pm$ 0.071	18.490 $\pm$ 0.939			
Point-to-plane_MSE [9]	0.465 $\pm$ 0.065	0.452 $\pm$ 0.065	20.170 $\pm$ 0.696			
PSNR Y [26]	0.638 $\pm$ 0.059	0.614 $\pm$ 0.061	17.507 $\pm$ 1.145			
Plane-to-plane [10]	0.474 $\pm$ 0.068	0.454 $\pm$ 0.069	20.051 $\pm$ 0.780			
LOG point-to-distribution-joint [28]	0.475 $\pm$ 0.079	0.435 $\pm$ 0.065	20.005 $\pm$ 0.860			
BitDance [35]	0.488 $\pm$ 0.055	0.451 $\pm$ 0.054	19.912 $\pm$ 0.799			
PointSSIM (geometry) [11]	0.402 $\pm$ 0.039	0.345 $\pm$ 0.054	20.914 $\pm$ 0.483			
PointSSIM (normal) [11]	0.631 $\pm$ 0.051	0.605 $\pm$ 0.059	17.664 $\pm$ 0.848			
PointSSIM (curvature) [11]	0.597 $\pm$ 0.056	0.573 $\pm$ 0.058	18.271 $\pm$ 0.904			
PointSSIM (luminance) [11]	0.485 $\pm$ 0.058	0.465 $\pm$ 0.059	19.945 $\pm$ 0.777			
PCQM [13]	<u>0.754</u> $\pm$ 0.034	<u>0.749</u> $\pm$ 0.036	<u>14.960</u> $\pm$ 0.892			
MPED [37]	0.618 $\pm$ 0.056	0.620 $\pm$ 0.055	17.893 $\pm$ 0.926			
PointPCA	<b>0.894</b> $\pm$ 0.027	<b>0.890</b> $\pm$ 0.029	<b>10.132</b> $\pm$ 1.195			

Bold represents best and underlined second-best performance

report the performance indexes as obtained from the within-dataset validation of PointPCA and the evaluation of the alternative metrics on the same testing partitions, as described in Sect. 4.4. Our results suggest that the PointPCA metric achieves the best performance in all datasets with high scores. Considering D1, the luminance-based PointSSIM variant achieves the second-best performance in terms of PLCC and RMSE followed by PCQM, which attains the second-best performance in terms of SROCC. The PCQM is consistently ranked as the second-best option in D2 and D3, followed by the MPED, and the normal- and curvature-based variants of the PointSSIM. It is evident that in D2 and D3, our proposed metric achieves substantial gains in terms of PLCC, SROCC, and RMSE with respect to alternative metrics.

## 6 Exploratory studies

In this section, we evaluate the impact of several parameters on the performance of the proposed metric to further understand their effect. In particular, we first analyze how the support region sizes influence the performance of individual predictors and total quality scores. Secondly, we explore the usage of different color spaces for the definition of textural descriptors. Thirdly, we study the effect of using predictors coming from only one out of the two attribute domains, namely, geometry or texture, to validate our selection of both. Lastly, we investigate the usage of different regression models to fuse individual predictors to total quality scores.

### 6.1 Support regions

In this first study, we aim to understand the impact of varying the size of the support regions over which we compute descriptors and statistical features on the performance of our metric. Please note that for the former, we use  $r$ -search to define a support region, whereas for the latter we employ  $k$ -nn, as explained in Sect. 3.3 and 3.4, respectively.

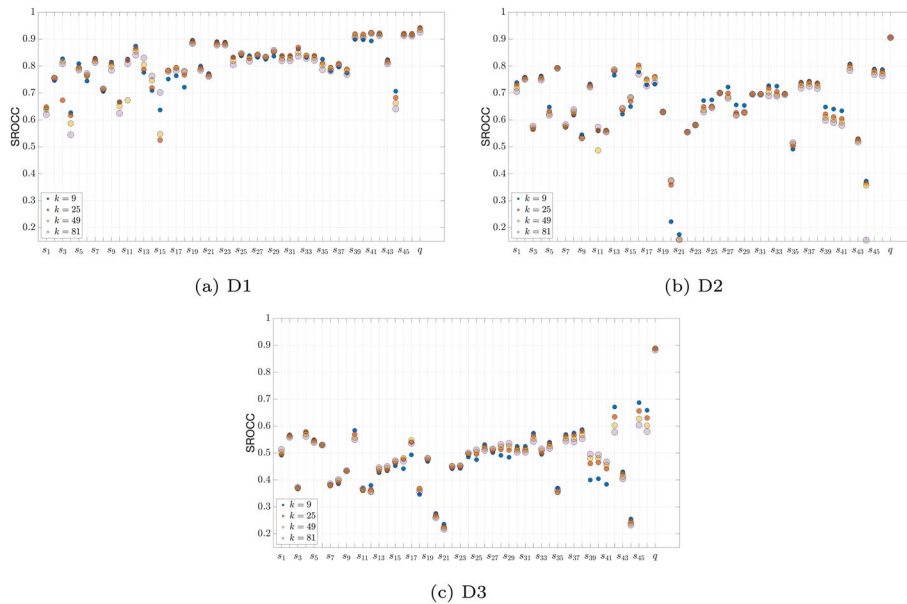
It is worth noting that there is an inter-dependency between support regions for descriptors and for statistical features. For example, decreasing the descriptors' support region leads to descriptor values being more susceptible to noise; thus, neighboring descriptor values will exhibit greater differences, better capturing high-frequency components. On the contrary, increasing the descriptors' support region causes a loss of fine details and is equivalent to smoothening the surface properties or applying a low-pass filter; in this case, neighboring descriptor values will be similar. At the same time, lowering the statistical features' support region implies that the descriptor values under consideration will be similar given that they are adjacent and reflect surface properties from very close vicinities. Conversely, increasing a statistical features' support region decreases the error due to the larger sample size; yet, it increases the dispersion between descriptor values due to the recruitment of remote, spatially irrelevant samples. Thus, there is a need to evaluate the effect of their configuration on the performance of the proposed metric. For this purpose, we initially fix the descriptors' and alter the statistical features' support region size; then, we fix the statistical features' and alter the descriptors' support region size.

**6.1.1 Support regions for statistical features**

In this case, we compute the statistical features using the  $k$ -nn algorithm with  $k = \{9, 25, 49, 81\}$ , and the descriptors using the  $r$ -search with  $r = 0.008 \times B_R$ . Our selection of  $k$  values is based on the fact that the point clouds of the datasets under consideration are voxelized, dense, and represent large models; thus, we may assume that small point neighborhoods represent local regions, which in turn can be approximated by planar surfaces. The selected  $k$  values represent the number of vertices in fully occupied planes of length size equal to 2, 4, 6, and 8 times the distance between two voxels. Figure 5 illustrates the SROCC values achieved by every predictor  $s_j, 1 \leq j \leq 46$  and the average SROCC values across all testing partitions attained by the total quality score  $q$  (i.e., the PointPCA metric), with different colors indicating the performance over different  $k$  values. Recall that predictors  $s_{1-23}$  make use of the mean, while predictors  $s_{24-46}$  employ the standard deviation. Moreover, for every statistic, the first 15 predictors refer to the geometry and the last 8 to the texture domain. Our results show that the selected neighborhood size for the computation of statistical features does not have a large impact on the performance, with the trends indicating that predictors perform better under smaller rather than larger neighborhoods. Moreover, it can be observed that the total quality scores  $q$  always outperform each individual predictor  $s_j$ . Finally, different neighborhood sizes lead to minor differences in the performance of total quality scores, slightly favoring smaller neighborhoods.

**6.1.2 Support regions for descriptors**

In this case, we compute the descriptors using the  $r$ -search algorithm with  $r = \{0.006 \times B_R, 0.008 \times B_R, 0.01 \times B_R\}$ , and the statistical features using the  $k$ -nn with  $k = 9$ . Our selection of  $r$  values is inspired by the current literature (e.g., [11, 13]), where



**Fig. 5** SROCC for every predictor  $s_j$  and average SROCC for the total quality score  $q$  in every dataset, under different neighborhood sizes using the  $k$ -nn algorithm with  $k = \{9, 25, 49, 81\}$  to compute statistical features, and the  $r$ -search with  $r = 0.008 \times B_R$  to compute descriptors

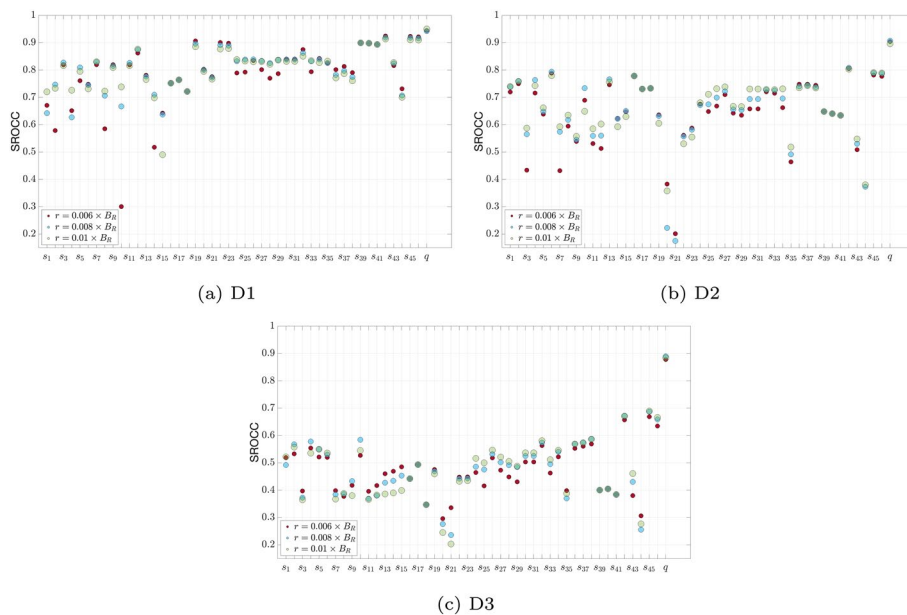
similar volume sizes have been used to compute point cloud features for objective quality assessment. Figure 6 shows the SROCC values achieved by every predictor  $s_j$  with  $1 \leq j \leq 46$  and the average SROCC values of the total quality score  $q$ . Our results indicate no clear pattern in the performance of mean-based predictors across all datasets, with geometric predictors (i.e.,  $s_{1-15}$ ) showing no consistent trends, and textural predictors (i.e.,  $s_{16-23}$ ) performing better in smaller neighborhoods. For the majority of predictors that employ standard deviation (i.e.,  $s_{24-46}$ ), though, larger neighborhood sizes are preferable. Please note that no differences can be observed across different  $r$  values for textural predictors  $s_{16-18}$  and  $s_{39-42}$  since they do not employ a support region for the computation of corresponding descriptors (i.e., these are the non-PCA-based descriptors,  $d_{1-3}^t$ , equal to the RGB color values). After fusing predictors into a total quality score  $q$ , we observe clear benefits with respect to individual predictors  $s_j$ . Finally, considering total quality scores, marginal differences with slight gains for mid over smaller or larger neighborhood sizes are remarked.

### 6.1.3 Final selection

Our results confirm that the total quality scores lead to high prediction accuracy under all tested configurations for the descriptors' and statistical features' support region sizes. In the proposed settings of our metric, we set  $r = 0.008 \times B_R$  and  $k = 9$  for descriptors and statistical features, respectively.

## 6.2 Color spaces

In this study, we examine the performance achieved with the proposed metric by computing the same textural descriptors in alternative color spaces that are popular in the literature. In particular, alongside the RGB color space, we use the YCbCr which has



**Fig. 6** SROCC for every predictor  $s_j$  and average SROCC for the total quality score  $q$  in every dataset, under different neighborhood sizes using the  $r$ -search algorithm with  $r = \{0.006 \times B_R, 0.008 \times B_R, 0.01 \times B_R\}$  to compute descriptors, and the  $k$ -nn with  $k = 9$  to compute statistical features

been widely used for objective quality assessment; in our case, the color space conversion is performed following the ITU-R Recommendation BT.709 [59]. Moreover, we employ the GCM [60] which is reported to correlate well with human perception, and CIELAB [61] which is recommended by the International Commission on Illumination in 1976 and designed for perceptual uniformity. Note that in this analysis, we use all predictors from both geometric and textural domains. Specifically, instead of using textural predictors only, we additionally include geometric predictors to compute total quality scores, which are then compared to subjective ground truth ratings. This way, we do not explicitly assess the performance of the same textural predictors under different color spaces; but, we explore the effect of different color spaces in the performance of the proposed metric and aim to identify the one that leads to the most beneficial interactions between geometric and textural predictors. Similarly to the analysis of Sect. 5.2, we learn optimal weights for all predictors per dataset and test the accuracy of the learned models in both within- and cross-dataset validation.

In Table 4, we present the performance indexes obtained from our metric considering different color spaces. In general, small variations in performance can be observed. In the majority of cases, RGB has either equivalent or marginally better performance with respect to the other color spaces. In particular, RGB leads to better performance for within-dataset validation for D1 (PLCC = 0.938, SROCC = 0.942) and D3 (PLCC = 0.894, SROCC = 0.890), whereas it ranks second behind YCbCr for D2 (PLCC = 0.935, SROCC = 0.911 for YCbCr, against PLCC = 0.932, SROCC = 0.907 for RGB). For cross-dataset validation, YCbCr performs better when training on D1 and D2 and testing on D3 (training on D1, testing on D3: PLCC = 0.571, SROCC = 0.574; training on D2, testing on D3: PLCC = 0.690, SROCC = 0.679). GCM performs better when training on D2 and D3, and testing on D1 (training on D2, testing on D1: PLCC = 0.828, SROCC = 0.837; training on D3, testing on D1: PLCC = 0.802, SROCC = 0.835). On the other hand, RGB performs better when training on

**Table 4** Performance evaluation of different color spaces (CSs)

CS	Train	Test								
		D1			D2			D3		
		PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$	PLCC $\uparrow$	SROCC $\uparrow$	RMSE $\downarrow$
YCbCr	D1	0.932	<u>0.939</u>	0.461	<u>0.795</u>	<u>0.793</u>	<u>1.466</u>	<b>0.571</b>	<b>0.574</b>	<b>18.794</b>
	D2	0.822	0.832	0.773	<b>0.935</b>	<b>0.911</b>	<b>0.838</b>	<b>0.690</b>	<b>0.679</b>	<b>16.557</b>
	D3	0.775	0.795	0.856	0.853	0.822	1.266	<b>0.894</b>	<u>0.885</u>	<u>10.142</u>
GCM	D1	0.931	<u>0.939</u>	0.465	0.791	0.786	1.477	0.565	<u>0.569</u>	18.891
	D2	<b>0.828</b>	<b>0.837</b>	<b>0.760</b>	<u>0.933</u>	0.905	<u>0.856</u>	0.680	0.673	16.789
	D3	<b>0.802</b>	<b>0.835</b>	<b>0.810</b>	<u>0.856</u>	<u>0.841</u>	<u>1.256</u>	<u>0.884</u>	0.878	10.582
CIELAB	D1	<u>0.933</u>	<u>0.939</u>	<u>0.460</u>	0.775	0.775	1.524	0.557	0.558	19.021
	D2	0.821	0.830	0.774	0.931	0.901	0.869	0.643	0.632	17.549
	D3	<u>0.794</u>	<u>0.829</u>	<u>0.825</u>	0.834	0.825	1.340	0.871	0.864	11.151
RGB	D1	<b>0.938</b>	<b>0.942</b>	<b>0.444</b>	<b>0.808</b>	<b>0.803</b>	<b>1.423</b>	<u>0.567</u>	<u>0.569</u>	<u>18.864</u>
	D2	<u>0.824</u>	<u>0.836</u>	<u>0.769</u>	0.932	<u>0.907</u>	0.859	<u>0.683</u>	<u>0.678</u>	<u>16.727</u>
	D3	0.786	0.817	0.838	<b>0.862</b>	<b>0.842</b>	<b>1.229</b>	<b>0.894</b>	<b>0.890</b>	<b>10.132</b>

Italics indicate within-dataset results. The best performance for train/test combination across CSs is shown in bold; the second best is underlined

D1 and D3 and testing on D2 (training on D1, testing on D2: PLCC = 0.808, SROCC = 0.803; training on D3, testing on D2: PLCC = 0.862, SROCC = 0.842). However, as can be seen, the differences are rather small, showing the robustness of our metric with respect to the color space selection.

### 6.3 Geometric and textural predictors

In this study, we evaluate the impact of using predictors from different attribute domains (i.e., geometry or texture) on the proposed metric. To do so, we compute total quality scores considering geometry-only (i.e.,  $[s_{1-15}, s_{24-38}]$ ) and texture-only predictors (i.e.,  $[s_{16-23}, s_{39-46}]$ ), and we compare their performance with respect to using the whole set (i.e.,  $[s_{1-46}]$ ).

Results are shown in Table 5 for all datasets. It can be observed that for within-dataset validation, using both attribute domains leads to steadily better performance with respect to only using one. For D1 and D3, using textural information only leads to better performance with respect to using geometry only (D1: PLCC = 0.930, SROCC = 0.941 for texture only, versus PLCC = 0.903, SROCC = 0.907 for geometry only; D3: PLCC = 0.823, SROCC = 0.812 for texture only, versus PLCC = 0.662, SROCC = 0.625 for geometry only), whereas for D2, the opposite is true (D2: PLCC = 0.911, SROCC = 0.868 for geometry only, versus PLCC = 0.882, SROCC = 0.864 for texture only). This can be explained considering the nature of the datasets, namely, while D1 and D3 contain compression distortions where geometry and texture are simultaneously affected, D2 contains several point clouds with only geometry or only texture distortions.

For cross-dataset validation, we can observe that when testing on D1, using texture-only descriptors leads to better performance with respect to using the whole set, whereas when testing on D2, using the whole set leads to consistently better results. When training on D1 and testing on D3, textural information leads to the best performance; however, when training on D2, using the whole set is preferable. In general, we see that using predictors from both attribute domains leads to higher performance, followed by texture-only predictors, with geometry-only predictors denoting the least optimal solution.

**Table 5** Performance evaluation of different attribute domains (ADs)

AD	Train	Test								
		D1			D2			D3		
		PLCC ↑	SROCC ↑	RMSE ↓	PLCC ↑	SROCC ↑	RMSE ↓	PLCC ↑	SROCC ↑	RMSE ↓
Geometry	D1	0.903	0.907	0.561	<u>0.754</u>	0.695	<u>1.593</u>	0.438	0.459	20.609
	D2	0.636	0.658	1.044	<u>0.911</u>	<u>0.868</u>	<u>0.992</u>	0.533	0.499	19.378
	D3	0.655	0.667	1.025	<u>0.810</u>	0.750	<u>1.422</u>	0.662	0.625	17.059
Texture	D1	<u>0.930</u>	<u>0.941</u>	<u>0.469</u>	0.737	<u>0.732</u>	1.640	<b>0.577</b>	<b>0.584</b>	<b>18.707</b>
	D2	<b>0.845</b>	<b>0.848</b>	<b>0.721</b>	0.882	0.864	1.113	<u>0.630</u>	<u>0.614</u>	<u>17.733</u>
	D3	<b>0.827</b>	<b>0.834</b>	<b>0.764</b>	0.777	<u>0.783</u>	1.526	<u>0.823</u>	<u>0.812</u>	<u>12.929</u>
Both	D1	<b>0.938</b>	<b>0.942</b>	<b>0.444</b>	<b>0.808</b>	<b>0.803</b>	<b>1.423</b>	<u>0.567</u>	<u>0.569</u>	<u>18.864</u>
	D2	<u>0.824</u>	<u>0.836</u>	<u>0.769</u>	<b>0.932</b>	<b>0.907</b>	<b>0.859</b>	<b>0.683</b>	<b>0.678</b>	<b>16.727</b>
	D3	<u>0.786</u>	<u>0.817</u>	<u>0.838</u>	<b>0.862</b>	<b>0.842</b>	<b>1.229</b>	<b>0.894</b>	<b>0.890</b>	<b>10.132</b>

The best performance for train/test combination across ADs is shown in bold; the second best is underlined

### 6.4 Regression models

In this study, we evaluate the performance achieved by the proposed metric when using different regression models to fuse individual predictors to a total quality score. Specifically, the Linear regression (R1), K-Nearest Neighbors (R2), Support Vector Regression (R3), XGBoost (R4), and Multi-Layer Perceptron (R5) are examined as alternatives to the proposed Random Forest (R6), as implemented in the scikit-learn python package [58]. For R1–R4, we use the default parameters. For R5, we use 3 hidden fully connected layers with 128 neurons each; the input nodes are set equal to the number of predictors (i.e., 46) and the output nodes to one; ReLU activation function and MSE as loss function are employed. For R6, we use MSE as a criterion for a split. Moreover, our experimentation on the number of trees indicates stable performance from 50 to 350 trees; hence, we keep the default configuration with 100 trees, as mentioned in Sect. 4.3.

Performance results for every quality prediction model are presented in Table 6. As can be seen, the performance remains high and stable for the majority of regression models when training and testing on the same dataset; drops are observed using R1 with D1 and D2, and also using R3 with D3. R3 is the best-performing model in D1 (PLCC = 0.941, SROCC = 0.942), whereas for D2 and D3, R6 is the best for within-dataset validation (D2: PLCC = 0.932, SROCC = 0.907; D3: PLCC = 0.894, SROCC = 0.890).

Regarding the performance of the tested regression models, R1 seems to be the weakest option, with limited generalization capabilities, independently of the dataset used for training. For the remaining regression models, the trends are similar, although different selections lead to the best generalization results, per training dataset. For instance, when

**Table 6** Performance evaluation of different regression models (RMs)

RM	Train	Test								
		D1			D2			D3		
		PLCC ↑	SROCC ↑	RMSE ↓	PLCC ↑	SROCC ↑	RMSE ↓	PLCC ↑	SROCC ↑	RMSE ↓
R1	D1	0.816	0.796	0.742	0.396	0.363	2.192	0.258	0.113	22.079
	D2	0.730	0.740	0.896	0.866	0.834	1.145	0.410	0.374	20.815
	D3	0.568	0.448	1.116	0.777	0.734	1.523	0.875	0.871	10.969
R2	D1	0.922	<u>0.934</u>	0.493	0.786	0.785	1.499	<b>0.621</b>	<b>0.623</b>	<b>17.963</b>
	D2	<u>0.893</u>	<u>0.905</u>	<u>0.611</u>	0.926	0.902	0.896	0.652	0.643	17.371
	D3	0.672	0.693	1.006	0.804	0.787	1.442	0.852	0.835	11.844
R3	D1	<b>0.941</b>	<b>0.942</b>	<b>0.431</b>	<b>0.813</b>	0.793	<b>1.409</b>	0.492	0.370	19.953
	D2	<b>0.910</b>	<b>0.922</b>	<b>0.564</b>	<u>0.928</u>	0.902	0.884	<b>0.685</b>	<u>0.675</u>	<b>16.699</b>
	D3	<b>0.840</b>	<b>0.854</b>	<b>0.738</b>	<b>0.862</b>	<u>0.830</u>	<u>1.230</u>	0.777	0.760	14.317
R4	D1	0.928	0.932	0.486	0.801	<u>0.795</u>	1.449	<u>0.572</u>	<u>0.577</u>	<u>18.764</u>
	D2	0.794	0.803	0.820	0.924	0.895	0.909	0.618	0.599	17.973
	D3	0.677	0.694	0.998	0.848	0.807	1.283	0.882	0.877	10.703
R5	D1	0.916	0.923	0.520	0.784	0.770	1.496	0.463	0.464	20.272
	D2	0.889	0.903	0.620	<b>0.932</b>	<u>0.906</u>	<u>0.860</u>	0.679	0.667	16.809
	D3	0.753	0.787	0.895	<u>0.854</u>	0.828	1.264	<u>0.892</u>	<u>0.886</u>	<u>10.238</u>
R6	D1	<u>0.938</u>	<b>0.942</b>	<u>0.444</u>	<u>0.808</u>	<b>0.803</b>	<u>1.423</u>	0.567	0.569	18.864
	D2	0.824	0.836	0.769	<b>0.932</b>	<b>0.907</b>	<b>0.859</b>	<u>0.683</u>	<b>0.678</b>	<u>16.727</u>
	D3	<u>0.786</u>	<u>0.817</u>	<u>0.838</u>	<b>0.862</b>	<b>0.842</b>	<b>1.229</b>	<b>0.894</b>	<b>0.890</b>	<b>10.132</b>

The best performance for train/test combination across RMs is shown in bold; the second best is underlined

training on D1, R3 and R6 show higher generalization capabilities on D2 (PLCC = 0.813, SROCC = 0.793 for R3, PLCC = 0.808, SROCC = 0.803 for R6), while R2 is the best for D3 (PLCC = 0.621, SROCC = 0.623). When training on D2, R3 is the best option on D1 (PLCC = 0.910, SROCC = 0.922) and D3 (PLCC = 0.685, SROCC = 0.675), while R2 and R6 achieve second-best performances, respectively. Finally, when training on D3, R3 obtains the best performance on D1 by large margins (PLCC = 0.840, SROCC = 0.854), whereas on D2, R6 outperforms the rest (PLCC = 0.862, SROCC = 0.842) and is closely followed by R3 (PLCC = 0.862, SROCC = 0.830).

To see whether the difference in results between different regressors had statistical significance, we ran a 2-tailed t-test on the performance indexes obtained when training and testing on the same dataset, for all regressor pairs, across all the splits. For D1, R1 had statistically significant differences with respect to all other regressors, according to all performance indexes ( $p < 0.001$  for all comparisons). In terms of PLCC, statistical differences were found between R3 and R5 ( $p = 0.0248$ ), and between R6 and R5 ( $p = 0.0465$ ); analogous results were obtained in terms of RMSE (R3–R5:  $p = 0.0055$ ; R3–R5:  $p = 0.0183$ ), whereas for SROCC, statistical differences were only observed for R3 with respect to R5 ( $p = 0.0169$ ). For D2, R1 was the only regressor exhibiting statistically significant differences with respect to all other regressors, according to all performance indexes (for PLCC and RMSE,  $p < 0.001$  for all comparisons; for SROCC, R1–R3:  $p = 0.0010$ ; R1–R4:  $p = 0.0028$ ,  $p < 0.001$  for all other comparisons). Finally, for D3, we found statistically significant differences between all the regressors under test, according to all performance metrics ( $p < 0.001$  for all comparisons). The latter is to be expected due to the large number of training/testing splits, which results in high degrees of freedom for the t-test. In general, the statistical test confirms our previous observations: with the exception of linear regression, all regressors under testing have similarly high performance, which demonstrates the robustness of the predictors with respect to the choice of regression models.

In conclusion, R3 and R6 lead to quality prediction models with the highest performance and generalization capabilities. In particular, R3 shows slightly better performance when testing on D1, whereas R6 achieves much better results on D3; on D2, R6 is the best with R3 attaining comparable performance. Overall, statistical analysis shows that differences between different regressors are not significant for D1 and D2, except R1, which was always found to be significantly different than the other regressors. It is worth noting that for both R3 and R6, performance indexes from within-dataset validation show improvements over state-of-the-art metrics. Finally, all regression models excluding R1 perform better than alternative metrics in D2 and D3, while they are following closely in D1, if not preceding.

## 7 Conclusion

In this paper, we propose a point cloud objective quality metric that relies on PCA-based shape and appearance predictors to evaluate distortions in the geometry and color domain, respectively. Statistical functions are applied to the descriptor values in order to capture local relationships between point samples, which are compared between a reference and a point cloud under evaluation, producing predictions of visual quality for the latter. The proposed predictors are assessed individually, showing good overall



performance, with some textural variants leading to higher accuracy consistently across all tested datasets. To boost the performance by leveraging the predictive potential of all the proposed predictors and return a single quality score, the Random Forest regression model is employed as part of our architecture. Alternative learning-based models are examined and evaluated, indicating that non-linear variants lead to similarly high performance. Moreover, the selection of parameter configuration, color space, and usage of descriptors from both geometry and texture domains are justified through a series of exploratory studies. Our results show that PointPCA outperforms existing metrics in all tested datasets. Considering that certain predictors are more efficient against particular types of contents and degradations, future work will focus on the identification and adoption of optimal subsets of predictors, per use case. Moreover, ensemble of regressors will be tested to increase the prediction power of our predictors.

#### Acknowledgements

We thank Konstantinos Ntemos and Hermina Petric Maretic for the useful discussion around the complexity of our algorithm.

#### Author contributions

EA provided the main idea for the work, defined the framework and theoretical basis of the metric, conducted the majority of the experiments and the experimental analysis, and drafted the manuscript. XZ aided in the running of the experiments, specifically the regression models, and with the comparison with the state of the art. IV aided in the definition of the theoretical framework and experimental analysis, as well as in the drafting of the manuscript. PC provided feedback on the idea and experimental setup and aided in the drafting of the manuscript.

#### Funding

This work was partially supported through the NWO WISE grant and the European Commission Horizon Europe program, under the grant agreement 101070109, *TRANSMIXR* <https://transmixr.eu/>. *Funded by the European Union.*

#### Availability of data and materials

The data that support the findings of this study are available from the parties that provided the data, as cited in the document [7, 43, 55]. Restrictions may apply to the availability of these data, which were used under license for the current study, and so they might not be publicly available. Data are however available from the authors upon reasonable request and with permission of the third parties. The software developed in this work is available at the following link: [https://github.com/cwi-dis/pointpca\\_suite](https://github.com/cwi-dis/pointpca_suite), under *PointPCA*.

#### Declarations

##### Competing interests

The authors declare that they have no conflict of interest.

Received: 5 February 2024 Accepted: 6 May 2024

Published online: 09 August 2024

#### References

1. S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P.A. Chou, R.A. Cohen, M. Krivokuća, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A.M. Tourapis, V. Zakharchenko, Emerging MPEG standards for point cloud compression. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **9**(1), 133–148 (2019). <https://doi.org/10.1109/JETCAS.2018.2885981>
2. T. Ebrahimi, S. Foessel, F. Pereira, P. Schelkens, JPEG Pleno: toward an efficient representation of visual reality. *IEEE Multimedia* **23**(4), 14–20 (2016). <https://doi.org/10.1109/MMUL.2016.64>
3. ISO/IEC 23090-5: Information technology—Coded representation of immersive media—Part 5: Visual volumetric video-based coding (V3C) and video-based point cloud compression (V-PCC). International Organization for Standardization (2021)
4. ISO/IEC 23090-9: Information technology—Coded representation of immersive media—Part 9: Geometry-based point cloud compression. International Organization for Standardization (2023)
5. P. Astola, L.A. Silva Cruz, E.A. Da Silva, T. Ebrahimi, P.G. Freitas, A. Gilles, K.-J. Oh, C. Pagliari, F. Pereira, C. Perra et al., *Jpeg pleno: Standardizing a coding framework and tools for plenoptic imaging modalities* (ICT Discoveries, ITU Journal, 2020)
6. ISO/IEC AWI 21794-6: Information technology—Plenoptic image coding system (JPEG Pleno)—Part 6: Learning-based Point Cloud Coding. International Organization for Standardization (2024)
7. E. Alexiou, I. Viola, T.M. Borges, T.A. Fonseca, R.L. de Queiroz, T. Ebrahimi, A comprehensive study of the rate-distortion performance in MPEG point cloud compression. *APSIPA Trans. Signal Inf. Process.* **8**, 27 (2019)

8. E. Alexiou, Y. Nehmé, E. Zerman, I. Viola, G. Lavoué, A. Ak, A. Smolic, P. Le Callet, P. Cesar, Subjective and objective quality assessment for volumetric video. In: Valenzise, G., Alain, M., Zerman, E., Ozcinar, C. (eds.) *Immersive Video Technologies*, (Academic Press, Cambridge, Massachusetts, 2023), pp. 501–552. <https://doi.org/10.1016/B978-0-32-391755-1.00024-9>
9. D. Tian, H. Ochimizu, C. Feng, R. Cohen, A. Vetro, Geometric distortion metrics for point cloud compression. In: 2017 IEEE International Conference on Image Processing (ICIP), pp. 3460–3464 (2017). <https://doi.org/10.1109/ICIP.2017.8296925>
10. E. Alexiou, T. Ebrahimi, Point cloud quality assessment metric based on angular similarity. In: 2018 IEEE International Conference on Multimedia and Expo (ICME), pp. 1–6 (2018). <https://doi.org/10.1109/ICME.2018.8486512>
11. E. Alexiou, T. Ebrahimi, Towards a point cloud structural similarity metric. In: 2020 IEEE International Conference on Multimedia Expo Workshops (ICMEW), pp. 1–6 (2020). <https://doi.org/10.1109/ICMEW46912.2020.9106005>
12. G. Meynet, J. Digne, G. Lavoué, PC-MSDM: a quality metric for 3D point clouds. In: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–3 (2019). <https://doi.org/10.1109/QoMEX.2019.8743313>
13. G. Meynet, Y. Nehmé, J. Digne, G. Lavoué, PCQM: a full-reference quality metric for colored 3D point clouds. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2020). <https://doi.org/10.1109/QoMEX48832.2020.9123147>
14. L. Hua, M. Yu, G. Jiang, Z. He, Y. Lin, VQA-CPC: a novel visual quality assessment metric of color point clouds. In: Dai, Q., Shimura, T., Zheng, Z. (eds.) *Optoelectronic Imaging and Multimedia Technology VII*, International Society for Optics and Photonics, vol. 11550, (SPIE, Bellingham, WA, 2020), pp. 244–252.
15. L. Hua, M. Yu, Z. He, R. Tu, G. Jiang, CPC-GSCT: Visual quality assessment for coloured point cloud based on geometric segmentation and colour transformation. *IET Image Processing* (2021)
16. Z. Zhang, W. Sun, X. Min, T. Wang, W. Lu, G. Zhai, No-reference quality assessment for 3D colored point cloud and mesh models. *IEEE Trans. Circuits Syst. Video Technol.* **32**(11), 7618–7631 (2022). <https://doi.org/10.1109/TCSVT.2022.3186894>
17. Z. Li, A. Aaron, I. Katsavounidis, A. Moorthy, M. Manohara, Toward a practical perceptual video quality metric. *Netflix Tech Blog* **6**(2) (2016)
18. N. Chehata, L. Guo, C. Mallet, Airborne lidar feature selection for urban classification using random forests. In: *Laser-scanning* (2009)
19. M. Weinmann, B. Jutzi, S. Hinz, C. Mallet, Semantic point cloud interpretation based on optimal neighborhoods, relevant features and efficient classifiers. *ISPRS J. Photogramm. Remote. Sens.* **105**, 286–304 (2015). <https://doi.org/10.1016/j.isprsjprs.2015.01.016>
20. T. Hackel, J.D. Wegner, K. Schindler, Fast semantic segmentation of 3D point clouds with strongly varying density. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 177–184 (2016)
21. T. Hackel, J.D. Wegner, K. Schindler, Contour detection in unstructured 3D point clouds. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1610–1618 (2016). <https://doi.org/10.1109/CVPR.2016.178>
22. A. Javaheri, C. Brites, F. Pereira, J. Ascenso, A generalized Hausdorff distance based quality metric for point cloud geometry. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2020). <https://doi.org/10.1109/QoMEX48832.2020.9123087>
23. D. Tian, H. Ochimizu, C. Feng, R. Cohen, A. Vetro, Evaluation metrics for point cloud compression. ISO/IEC JTC1/SC29/WG11 Doc. M39966, Geneva, Switzerland (2017)
24. A. Javaheri, C. Brites, F. Pereira, J. Ascenso, Improving psnr-based quality metrics performance for point cloud geometry. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 3438–3442 (2020). <https://doi.org/10.1109/ICIP40778.2020.9191233>
25. A. Javaheri, C. Brites, F. Pereira, J. Ascenso, Mahalanobis based point to distribution metric for point cloud geometry quality evaluation. *IEEE Signal Process. Lett.* **27**, 1350–1354 (2020). <https://doi.org/10.1109/LSP.2020.3010128>
26. D. Tian, H. Ochimizu, C. Feng, R. Cohen, A. Vetro, Updates and Integration of Evaluation Metric Software for PCC. ISO/IEC JTC1/SC29/WG11 Doc. MPEG2017/M40522, Hobart, Australia (2017)
27. I. Viola, S. Subramanyam, P. Cesar, A color-based objective quality metric for point cloud contents. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2020). <https://doi.org/10.1109/QoMEX48832.2020.9123089>
28. A. Javaheri, C. Brites, F. Pereira, J. Ascenso, A point-to-distribution joint geometry and color metric for point cloud quality assessment. In: 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6 (2021). <https://doi.org/10.1109/MMSP53017.2021.9733670>
29. Q. Yang, Z. Ma, Y. Xu, Z. Li, J. Sun, Inferring point cloud quality via graph similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1–1 (2020) <https://doi.org/10.1109/TPAMI.2020.3047083>
30. Y. Zhang, Q. Yang, Y. Xu, MS-GraphSIM: Inferring point cloud quality via multiscale graph similarity. In: *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 1230–1238. Association for Computing Machinery, New York, NY, USA (2021). <https://doi.org/10.1145/3474085.3475294>
31. R. Diniz, P.G. Freitas, M.C.Q. Farias, Towards a point cloud quality assessment model using local binary patterns. In: 2020 Twelfth International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2020). <https://doi.org/10.1109/QoMEX48832.2020.9123076>
32. R. Diniz, P.G. Freitas, M.C.Q. Farias, Multi-distance point cloud quality assessment. In: 2020 IEEE International Conference on Image Processing (ICIP), pp. 3443–3447 (2020). <https://doi.org/10.1109/ICIP40778.2020.9190956>
33. R. Diniz, P.G. Freitas, M.C.Q. Farias, Local luminance patterns for point cloud quality assessment. In: 2020 IEEE 22nd International Workshop on Multimedia Signal Processing (MMSP), pp. 1–6 (2020). <https://doi.org/10.1109/MMSP48831.2020.9287154>
34. R. Diniz, P.G. Freitas, M. Farias, A novel point cloud quality assessment metric based on perceptual color distance patterns. *Electron. Imaging* **2021**(9), 256–125611 (2021). <https://doi.org/10.2352/ISSN.2470-1173.2021.9.IQSP-256>
35. R. Diniz, P.G. Freitas, M.C.Q. Farias, Color and geometry texture descriptors for point-cloud quality assessment. *IEEE Signal Process. Lett.* **28**, 1150–1154 (2021). <https://doi.org/10.1109/LSP.2021.3088059>

36. Y. Xu, Q. Yang, L. Yang, J.-N. Hwang, EPES: point cloud quality modeling using elastic potential energy similarity. *IEEE Trans. Broadcast.* **68**(1), 33–42 (2022)
37. Q. Yang, Y. Zhang, S. Chen, Y. Xu, J. Sun, Z. Ma, MPED: quantifying point cloud distortion based on multiscale potential energy discrepancy. *IEEE Trans. Pattern Anal. Mach. Intell.* **45**(5), 6037–6054 (2023)
38. I. Viola, P. Cesar, A reduced reference metric for visual quality evaluation of point cloud contents. *IEEE Signal Process. Lett.* **27**, 1660–1664 (2020). <https://doi.org/10.1109/LSP.2020.3024065>
39. Q. Liu, H. Yuan, R. Hamzaoui, H. Su, J. Hou, H. Yang, Reduced reference perceptual quality model with application to rate control for video-based point cloud compression. *IEEE Trans. Image Process.* **30**, 6623–6636 (2021). <https://doi.org/10.1109/TIP.2021.3096060>
40. L. Hua, G. Jiang, M. Yu, Z. He, BQE-CVP: Blind quality evaluator for colored point cloud based on visual perception. In: 2021 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB), pp. 1–6 (2021). <https://doi.org/10.1109/BMSB53066.2021.9547070>
41. E.M. Torlig, E. Alexiou, T.A. Fonseca, R.L. de Queiroz, T. Ebrahimi, A novel methodology for quality assessment of voxelized point clouds. In: Tescher, A.G. (ed.) *Applications of Digital Image Processing XLI*, International Society for Optics and Photonics, vol. 10752, (SPIE, Bellingham, WA, 2018), pp. 174–190.
42. E. Alexiou, T. Ebrahimi, Exploiting user interactivity in quality assessment of point cloud imaging. In: 2019 Eleventh International Conference on Quality of Multimedia Experience (QoMEX), pp. 1–6 (2019). <https://doi.org/10.1109/QoMEX.2019.8743277>
43. Q. Yang, H. Chen, Z. Ma, Y. Xu, R. Tang, J. Sun, Predicting the perceptual quality of point cloud: A 3D-to-2D projection-based exploration. *IEEE Transactions on Multimedia*, 1–1 (2020) <https://doi.org/10.1109/TMM.2020.3033117>
44. Z. He, G. Jiang, Z. Jiang, M. Yu, Towards a colored point cloud quality assessment method using colored texture and curvature projection. In: 2021 IEEE International Conference on Image Processing (ICIP), pp. 1444–1448 (2021). <https://doi.org/10.1109/ICIP42928.2021.9506762>
45. T. Chen, C. Long, H. Su, L. Chen, J. Chi, Z. Pan, H. Yang, Y. Liu, Layered projection-based quality assessment of 3D point clouds. *IEEE Access* **9**, 88108–88120 (2021). <https://doi.org/10.1109/ACCESS.2021.3087183>
46. Z. Wang, Q. Li, Information content weighting for perceptual image quality assessment. *IEEE Trans. Image Process.* **20**(5), 1185–1198 (2011). <https://doi.org/10.1109/TIP.2010.2092435>
47. A. Javaheri, C. Brites, F. Pereira, J. Ascenso, Joint geometry and color projection-based point cloud quality metric. *IEEE Access* **10**, 90481–90497 (2022). <https://doi.org/10.1109/ACCESS.2022.3198995>
48. A. Chetouani, M. Quach, G. Valenzise, F. Dufaux, Deep learning-based quality assessment of 3D point clouds without reference. In: 2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW), pp. 1–6 (2021). <https://doi.org/10.1109/ICMEW53276.2021.9455967>
49. A. Chetouani, M. Quach, G. Valenzise, F. Dufaux, Convolutional Neural Network for 3D point cloud quality assessment with reference. In: 2021 IEEE 23rd International Workshop on Multimedia Signal Processing (MMSp), pp. 1–6 (2021). <https://doi.org/10.1109/MMSp53017.2021.9733565>
50. M. Quach, A. Chetouani, G. Valenzise, F. Dufaux, A deep perceptual metric for 3D point clouds. *Electron. Imaging* **2021**(9), 257–12577 (2021). <https://doi.org/10.2352/ISSN.2470-1173.2021.9.IQSP-257>
51. Q. Liu, H. Yuan, H. Su, H. Liu, Y. Wang, H. Yang, J. Hou, PQA-Net: Deep no reference point cloud quality assessment via multi-view projection. *IEEE Transactions on Circuits and Systems for Video Technology*, 1–1 (2021) <https://doi.org/10.1109/TCSVT.2021.3100282>
52. W. Tao, G. Jiang, Z. Jiang, M. Yu, Point cloud projection and multi-scale feature fusion network based blind quality assessment for colored point clouds. In: *Proceedings of the 29th ACM International Conference on Multimedia*, pp. 5266–5272. Association for Computing Machinery, New York, NY, USA (2021)
53. Z. Zhang, W. Sun, X. Min, Q. Zhou, J. He, Q. Wang, G. Zhai, Mm-pcqa: Multi-modal learning for no-reference point cloud quality assessment. *arXiv preprint arXiv:2209.00244* (2022)
54. Z. Shan, Q. Yang, R. Ye, Y. Zhang, Y. Xu, X. Xu, S. Liu, GPA-Net: No-reference point cloud quality assessment with multi-task graph convolutional network. *IEEE Transactions on Visualization and Computer Graphics* (2023)
55. Q. Liu, H. Su, Z. Duanmu, W. Liu, Z. Wang, Perceptual quality assessment of colored 3D point clouds. *IEEE Transactions on Visualization and Computer Graphics*, 1–1 (2022) <https://doi.org/10.1109/TVCG.2022.3167151>
56. ITU-T J.149: Method for specifying accuracy and cross-calibration of Video Quality Metrics (VQM). International Telecommunication Union (2004)
57. E. Alexiou, T. Ebrahimi, Benchmarking of the plane-to-plane metric (2020)
58. F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, E. Duchesnay, Scikit-learn: machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011)
59. ITU-R BT.709-6: Parameter values for the HDTV standards for production and international programme exchange. International Telecommunication Union (2015)
60. J.M. Geusebroek, R. Boomgaard, A.W.M. Smeulders, H. Geerts, Color invariance. *IEEE Trans. Pattern Anal. Mach. Intell.* **23**(12), 1338–1350 (2001)
61. ISO/CIE 11664-4:2019: Colorimetry — Part 4: CIE 1976 L\*a\*b\* colour space. International Organization for Standardization (2019)

## Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.