Deep Learning-based Point Cloud Geometry and Color Coding

Designing a Perceptually-Driven Differentiable Training Distortion Metric for Deep Learning-based Point Cloud Joint Geometry and Color Coding

Luís Coelho, André F. R. Guarda, and Fernando Pereira, *Fellow, IEEE*

***Abstract*—Deep learning (DL)-based coding has recently become very popular for multimedia data, notably images and point clouds (PCs). Training a DL coding model using the backpropagation algorithm requires a differentiable loss function. Thus, for PC joint geometry and color coding, both the PC geometry and color distortion metrics must be differentiable. Since the distortion/quality metrics commonly used for the final PC quality assessment do not meet this criterion, new PC distortion metrics have to be designed for DL-based training purposes. Moreover, for PC joint geometry and color coding, it is critical to define the balance between the geometry and color distortions in a meaningful way, ideally driven by the human perception and subjective quality assessment. In this context, this paper proposes a perceptually-driven design for a differentiable PC joint geometry and color distortion metric to be used for training purposes in DL-based coding, notably to define the relative weights for the geometry and color distortions. The obtained perceptually-driven weights achieve a rate reduction of around 3% regarding the default balanced weights at no complexity cost. This is the first proposal in the literature with this purpose and this perceptual approach.**

***Index Terms*— Point cloud, learning-based coding, distortion metric, perceptually-driven**

I. INTRODUCTION

In the current times, with the incredible growth of the internet and of the number of available displays, there has never been a greater need for realistic multimedia applications. Nowadays these services are still mostly two-dimensional-based, but due to lacking the desired realism, these representation models are quickly being replaced by three-dimensional (3D) media, with added degrees of freedom. One of the most prominent 3D representation models are Point Clouds (PCs), which are also the focus of this paper. PCs are composed by a set of unordered points, or voxels for a voxelized PC, in 3D space, located on the object's surface. These points are defined by their 3D coordinates, which compose the PC geometry, and may have additional attributes like color. PCs can be classified regarding their variation in time, with static PCs including a single instant in time and dynamic PCs varying along time. Another important PC characteristic concerns their density/sparsity depending on the average proximity between points since, differently from images and videos, PCs are unstructured and do not completely fill a uniform grid. In dense PCs, the points lie closer to each other and vice-versa for sparse PCs.

Since PCs can be composed of up to millions of 3D points, each with their own attributes, there is a critical need for highly efficient PC Coding/Compression (PCC) solutions, both for geometry and color, to make the transmission and storage of this type of content practically feasible. Recognizing this need, the Moving Pictures Experts Group (MPEG) and the Joint Photographic Experts Group (JPEG) have pressed towards this issue. MPEG has already developed two PCC standards [1], namely the Geometry-based PCC (G-PCC) and the Video-based PCC (V-PCC) standards. More recently, with the large-scale availability of multimedia data and the advances in hardware computational power, deep learning (DL) technologies have emerged to play a central role in PCC; currently, DL-based PC coding are already reaching competitive or better compression performance than the MPEG PCC standards, notably depending on their density. These DL-based solutions can code PC content by using convolutional neural networks (CNNs), which learn a transform to extract the most useful PC features. In this learning process, the loss function plays a central role for the successful creation of an efficient coding model. With this purpose in mind, the loss functions are commonly rate-distortion (RD)-driven, thus including a rate and a distortion component. The PC distortion corresponds to the error between the original and decoded PCs, i.e., the loss in reconstruction quality regarding the original PC. While there are several distortion metrics commonly used in the literature to assess the decoded PC quality that would be desirable to use directly in the training loss function, this is not always possible since the distortion metric to be used as a quality loss in the DL model training needs to be differentiable to fit in the backpropagation process. For example, in DL-based image coding, the same quality/distortion metric is commonly used for final performance assessment and training loss, but this is not always possible in DL-based PC coding since the distortion metrics commonly used for final distortion/quality assessment have differentiability constraints. For PC joint geometry and color coding, the distortion includes two elementary distortion metrics, one for geometry and another for color, which may be weighted differently. Since it is well known that geometry and color do not have the same impact on the final subjective quality, it is critical to select appropriate weights to maximize the final PC quality and not

This work has been financially supported by the Fundação para a Ciência (FCT, Portugal) through the research project PTDC/EEI-COM/1125/2021, entitled "Deep Learning-based Point Cloud Representation".

L. Coelho, and F. Pereira are with Instituto Superior Técnico, Universidade de Lisboa and Instituto de Telecomunicações, Lisbon, Portugal (e-mail: luis.pinto.coelho@tecnico.ulisboa.pt, fp@lx.it.pt).

A. F. R. Guarda is with Instituto de Telecomunicações, Lisbon, Portugal (e-mail: andre.guarda@lx.it.pt).

the geometry or color qualities individually.

In this context, this paper proposes a perceptually-driven differentiable PC joint geometry and color distortion metric to be used in the training process of a DL-based PC joint geometry and color codec. This design aims at maximizing the joint RD performance as measured by a PC joint distortion metric and not the individual geometry and color RD performances. This is the first solution where the weights for the training distortion metric are designed with a clear approach, in this case perceptually-driven, and not just by exhaustive trial and error. Ideally, with this design approach the DL coding model is being trained to minimize the perceptual PC distortion, i.e., maximize the final subjective quality, leading to an increase in the joint RD performance due to the consideration of factors like the masking effect that good color has on poor geometry.

The remainder of the paper is structured as follows: Section II briefly reviews the relevant state-of-the-art PCC solutions. Section III introduces a proposed DL-based PC joint geometry and color codec that will be used to assess the impact of the proposed distortion metric. Section IV describes the design procedure behind the proposed PC joint distortion metric. Section V reports and discusses the RD performance results. Finally, Section VI concludes the paper and presents some future work.

II. Related Work

This section includes a brief review of the main PC geometry and color coding solutions in the literature, conventional and DL-based. The most important conventional PCC solutions in the literature are certainly the two MPEG Point Cloud Compression (PCC) standards [1], notably G-PCC and V-PCC for static and dynamic PCs, respectively. The G-PCC standard is an efficient voxel-based coding solution that structures the PC geometry as a hierarchical octree, which can be more or less pruned to save rate at the cost of quality. For denser PCs, G-PCC includes a so-called Trisoup coding mode, where the surface associated with the octree leaf nodes is further detailed as a set of fitting triangles. For color/attributes coding, G-PCC specifies two coding modes, notably the Region-adaptive Hierarchical Transform (RAHT) mode and the Predicting/Lifting Transform mode, based on Level of Detail (LOD) generation [2]. On the other hand, the V-PCC standard is a projection-based PC codec for dynamic PCs; this means the 3D PCs are transformed into 2D maps using 3D to 2D projections of both the geometry and color/attributes of a given PC onto six different planes, corresponding to each face of the PC bounding box. After the depth (i.e., geometry information) and color projected maps are obtained, the very efficient video coding standards already available, such as the High Efficiency Video Coding (HEVC) [3] and Versatile Video Coding (VVC) standards [4], may be used, granting excellent RD performance.

Recently, DL-based technologies started to play a role in multimedia coding in general and in PCC in particular. Among the several DL-based PCC solutions in the literature, it is worthwhile to refer the most relevant. The first is the Adaptive Deep Learning-based PCC (ADL-PCC) geometry codec [5], developed by Guarda *et al.*, which has served as starting point for the PC joint geometry and color codec presented in Section III. The ADL-PCC coding solution grants an efficient PC geometry coding performance for static PCs using a binary, voxel-based PC representation data structure. This coding solution uses an end-to-end DL-based coding model with a main CNN-based autoencoder (AE), which learns a transform that extracts features from the PC geometry, followed by a quantizer and an entropy encoding process. The parameters used for the entropy encoder are learned from a variational autoencoder (VAE), based on the latents statistical characteristics. The ADL-PCC codec trains multiple DL coding models, by adjusting a parameter in the training loss function, and targets higher compression efficiency by selecting the best model for each given PC block, depending on its characteristics, notably density. The ADL-PCC loss function is RD-driven where the geometry distortion is measured by the so-called Focal Loss (FL) [6]. Other DL-based PC geometry coding solutions adopting similar DL model architectures have been proposed in the literature, notably the DL-based PC geometry coding solution, by Quach *et al.* [7] and the multiscale PC geometry compression, by Wang *et al.* [8].

Regarding DL-based PC joint geometry and color, a single solution is proposed in the literature by Alexiou *et al.* [9]. This solution for static PCs jointly codes the geometry and color with a single end-to-end trained DL-based coding model accepting as input four channels, one for the (binary) PC geometry and three others for the (RGB, 8-bit) color components. The architecture for this DL-based model is an extension of the DL-based geometry coding architecture previously proposed by Quach *et al.* [7]. The same model may also be trained to code only the geometry or the color (if the geometry is previously available) by adopting different loss functions, depending on the type of PC data being coded. For joint geometry and color coding, the RD-driven loss function includes an additional parameter to weight the geometry and color distortions. While different weights have been tried, the authors ultimately selected balanced weights, i.e., 0.5 for both geometry and color. Moving from a default balanced definition of these weights is the main target of this paper, towards maximizing the RD performance acknowledging that the geometry and color have different subjective impacts.

III. DL-Based PC Joint Geometry and Color Coding Solution

This section introduces the proposed DL-based PC joint geometry and color coding solution adopted in this paper, labelled as IST-JPCC. An extension of this codec using an additional super-resolution post-processing step, labelled as IT-DL-PCC-GC [10], has been proposed to the JPEG Pleno PCC Call for Proposals [11] and selected as the best performing, in July 2022.

A. Architecture and Walkthrough

The IST-JPCC global architecture is presented in Fig. 1 and the DL-based coding model itself is detailed in Fig. 2.

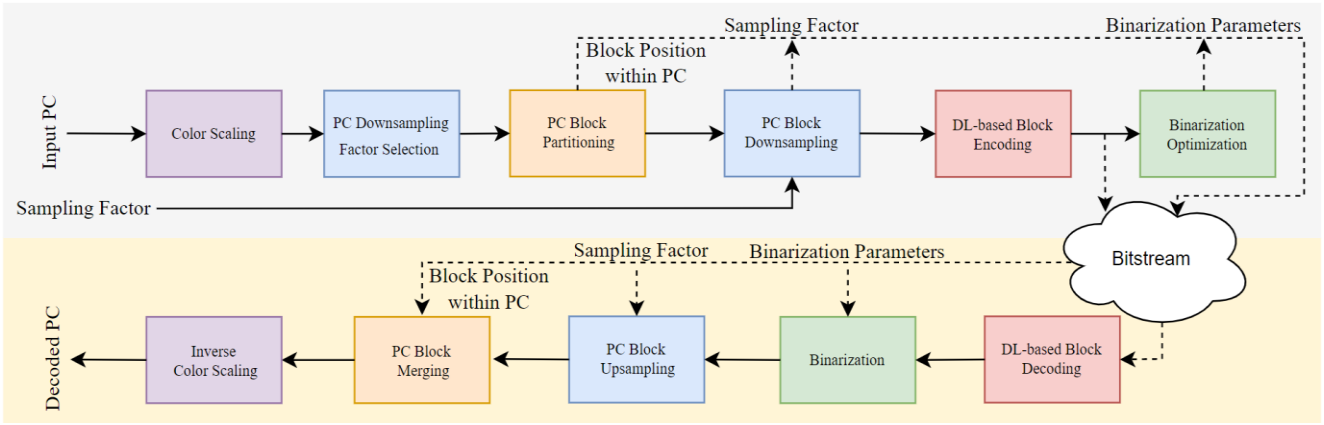

Fig. 1. IST-JPCC codec global architecture.

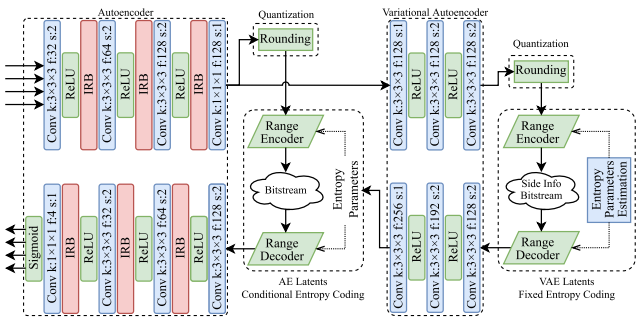

Fig. 2. End-to-end DL-based joint coding model architecture for geometry and color coding. The model input contains four channels: one for geometry and three for color, one for each component.

This coding solution is able to jointly code the geometry and color of a PC using a single end-to-end DL-based coding model, receiving data with four channels as input: one containing the geometry information, and the remaining three containing the color information, one for each color component in the RGB color space. The input PC is structured as 3D blocks of voxels, where the geometry for each voxel is binary-valued, i.e. either '0' (empty) or '1' (filled). On the other hand, the values for each color component have 8-bit precision and thus are integers ranging from 0 to 255. As shown in Fig. 1, the encoding/decoding process proceeds as follows:

- **Color Scaling**: The input PC color component values are scaled from their original range, i.e. [0, 255], down to a [0, 1] range. At this stage, both the geometry and color components are in the same range, which is useful for model training.
- **PC Downsampling Factor Selection**: For the codec to reach lower rates and be efficient for sparser PCs, the PC has to be downsampled to provide denser bocks for the DL-based coding model to code. Here the sampling factor (SF) is automatically determined by first computing the median value of the average distances of all points to their five closest neighbors, then selecting the closest previous power of two, i.e., 1, 2, 4, ... This SF is also included in the bitstream to allow reverting each block back to its original resolution/precision at the decoder side using block upsampling.
- **PC Block Partitioning**: The full input PC is divided into equal-sized 3D blocks of geometry and color, with only non-empty blocks being coded. The position of each block within the full PC is included in the bitstream to allow the reconstruction of the full PC at the decoder.
- **PC Block Downsampling**: The PC blocks (geometry and color) are downsampled using the SF value defined before.
- **DL-based Block Encoding**: Each block is encoded with an end-to-end trained DL-based coding model, using the architecture presented in Fig. 2. The encoder consists of a convolutional AE containing Inception-Residual blocks (IRB) [12], generating a set of coefficients or latents. These latents are then quantized using a quantization step (QS), rounded and then entropy encoded. The entropy coding model is learned from the latents' statistical distribution, using a VAE. All entropy encoded latents are included in the bitstream.
- **Binarization Optimization**: Since the geometry output generated by the DL-based coding model at the decoder are probability filling values for each voxel, there is a need to binarize these voxel geometry values to 1 ('filled') or 0 ('empty'), at the decoder, using some binarization strategy. In this proposed solution, a Top-*k* binarization approach is adopted, in which the *k* voxels with highest filling probability are selected as decoded points, i.e., filled voxels. This *k* value depends on the current block to code and is optimized at the encoder in this module by maximizing the performance for a selected geometry quality metric and for a selected color quality metric, in this case the PSNR D1 and PSNR YUV metrics. Before this optimization, the input block is divided into eight octants, and their occupancy is assessed so that the decoder avoids reconstructing points in octants that should be empty. A code signaling the block occupancy as well as the *k* value are included in the bitstream to be used at decoder side.
- **DL-based Block Decoding**: Every block is decoded by the decoder side of the DL-based model to transform the latents into voxel-filling probabilities.
- **Binarization**: Since the decoded blocks have their geometry values in the form of probabilities, every voxel is classified as filled ('1') or empty ('0'), using the

binarization parameters obtained at the encoder Binarization Optimization module for the Top-k approach and made available to the decoder.

- **PC Block Upsampling**: All blocks are upsampled back to their original resolution/precision, using the same SF used at the encoder for downsampling. While a more sophisticated upsampling solution may be used [13], here the upsampling process simply divides each voxel into a larger number of (smaller) voxels, depending on the used sampling factor and without increasing the number of points.
- **PC Block Merging**: All decoded blocks are put together at the right 3D position to recover the full decoded PC.
- **Inverse Color Scaling**: Since all color attributes obtained from the DL-based coding model are valued between 0 and 1, an additional step is taken to revert them back to their original range. After this scaling, the non-integer color values are rounded to their closest integer since every color component value corresponds to an 8-bit integer.

B. Training

As this paper will demonstrate again, the training process and the associated loss function are some of the most critical issues in DL-based visual data processing. To train the proposed DL-based coding model, presented in Fig. 2, the following RD-driven loss function was adopted:

$$Loss\ Function = Distortion + \lambda \cdot Coding\ Rate\,. \quad (1)$$

This loss function establishes a trade-off between the coding rate and the decoded PC distortion through the λ parameter. If λ is larger, RD points for lower rate and quality are obtained, and vice-versa. The coding rate is estimated as the entropy of the latent representation according to the VAE entropy coding model. Since this is a PC joint codec, the distortion component in the loss function is defined as a trade-off between color and geometry distortion metrics:

$$Distortion = (1-\omega)\cdot D_{Geometry} + \omega \cdot D_{Color}\,, \quad (2)$$

where ω is the trade-off weight. While it would be ideal to use for the training process the same distortion metrics used for the final PC quality assessment, notably the D1 or D2 distortion metrics for geometry, the MSE RGB or MSE YUV distortion metrics for color, or the PCQM [14] joint distortion, this is not possible since these metrics are not differentiable, preventing the use of backpropagation in the training process. This stems from the fact that these metrics compute distances and neighborhoods considering the point coordinates directly, which are not available during training without the binarization process, which is non-differentiable.

For this reason, differentiable distortion metrics had to be selected for the loss function, notably:

Geometry: The Focal Loss [6], which is a binary classification error metric, aiming to quantify whether each voxel is correctly predicted as filled or empty.

Color: The so-called 'same-voxel' (SV-)MSE distortion metrics. The suffix SV comes from the fact that the MSE is computed between the color value of a given filled voxel in the original PC and the color value *at the collocated voxel* in the decoded PC, no matter its probability of being-filled after decoding; this is required to obtain a differentiable MSE color metric. In this context, three SV-MSE color distortion metrics may be used for training, notably SV-MSE RGB, Y and YUV. The SV-MSE distortion metric for a single component, i.e., Y, can be obtained as follows:

$$SV\text{-}MSE_Y = \frac{\sum_{i=0}^{N}(Y_i - Y_j)^2}{N}\,, \quad (3)$$

where N is the total number of points in the original PC, Y_i is the luminance value of a filled voxel in the original PC and Y_j is the luminance value in the same voxel in the decoded PC, whatever the geometry filling probability. Along the same lines, SV-MSE RGB is computed as:

$$SV\text{-}MSE_{RGB} = SV\text{-}MSE_R^{\frac{1}{3}} \cdot SV\text{-}MSE_G^{\frac{1}{3}} \cdot SV\text{-}MSE_B^{\frac{1}{3}}\,, \quad (4)$$

where *SV-MSE R*, *SV-MSE G* and *SV-MSE B* correspond to the same-voxel mean squared error for each color component. Finally, *SV-MSE YUV* can be computed as:

$$SV\text{-}MSE_{YUV} = SV\text{-}MSE_Y^{\frac{6}{8}} \cdot SV\text{-}MSE_U^{\frac{1}{8}} \cdot SV\text{-}MSE_V^{\frac{1}{8}}\,, \quad (5)$$

where *SV-MSE Y* is the same-voxel MSE for the luminance and *SV-MSE U* and *SV-MSE V* are the same-voxel MSE for the chrominances. The weight 6 (out of 8) for the luminance is the usual way to express the higher human visual system sensitivity to this component.

The trade-off between the geometry and color distortions is defined through the ω parameter. By default, this parameter is commonly set to 0.5, thus equally balancing the reduction of color and geometry distortion during training. However, ideally, it should be possible to improve the overall RD performance of the DL-based coding model by considering a different weight distribution, thus acknowledging that geometry and color do not necessarily have the same impact in human perception, notably due to masking; this is, in fact, the goal of this paper.

The DL-based coding model was trained using a selection of static PCs listed in the JPEG Pleno PCC Common Training and Testing Conditions (CTTC) [15]. All selected PCs were downsampled and divided into blocks of size 64×64×64. All blocks with less than 500 'filled' voxels were discarded from training. The selected PCs were divided into training and validation sets, the latter of which was used in an early stopping procedure, with a patience of 5 epochs, to avoid overfitting. In total, 35861 blocks were used for training and 3822 blocks were used for validation.

IV. PERCEPTUALLY-DRIVEN DIFFERENTIABLE JOINT DISTORTION METRIC DESIGN

This section describes the process behind the design of a perceptually-driven differentiable PC joint distortion metric, taking into consideration the human perceptual subjective quality when defining the trade-off weight between geometry and color distortion for training.

A. Design Approach

The key objective of this paper is the design of a RD-based loss function for the training of a DL-based PCC model,

including a PC distortion metric that jointly considers the geometry and color with appropriate weights, in order to better consider the way users visually perceive a PC, notably both the geometry and color jointly. This type of metric should allow for a more optimized allocation of the rate between geometry and color and, ideally, a better final RD performance when a joint quality metric is considered. A key constraint for this training joint distortion metric is that it needs to be differentiable. For this reason, there is not much freedom in the selection of the differentiable PC geometry and color distortion metrics (as described in the previous section) but there is freedom on the selection of the weights for each of the elementary (differentiable) distortion metrics in the joint distortion metric to maximize the final subjective quality impact.

To design appropriate unbalanced weights, the key idea is to maximize the correlation between the objective joint distortion metric scores and a set of subjective distortion scores, experimentally obtained from the human assessment of a number of PCs coded with several PC codecs, at various bitrates. This design and validation process is inspired on the process used to develop some of the state-of-the-art objective PC joint quality metrics, like Point Cloud Quality Metric (PCQM) [14] adopted by JPEG and Point Cloud Structural Similarity Metric (PointSSIM) [16], which are defined as a combination of elementary PC geometry and color features. They were designed targeting maximum correlation with subjective quality scores available in some PC quality assessment datasets, such as the MPEG Point Cloud Compression Dataset (M-PCCD) [17] and the IST Rendering Point Cloud Quality Assessment Dataset (IRPC-D) [18]. For these objective PC joint quality metrics, the weights associated with each combined elementary metric are obtained by means of subjective scores fitting using a regression process between the objective and subjective quality/distortion scores. In this way, a clear relationship between the objective quality metric scores and the experimental subjective quality scores may be established as desired.

While the process above is typically used for the design of objective joint quality metrics, it will be followed in this paper for the design of a differentiable PC joint geometry and color distortion metric, since this is what is needed in a training loss function to measure the error/loss between the reference and decoded PCs.

B. Subjective Scores Database Description

The first step taken towards developing the PC joint distortion metric was to choose a dataset containing appropriate subjective quality scores for decoded PCs. While there were several options, it was decided to adopt the IRPC-D [18] which is composed of subjective data and PCs with different rendering procedures: two rendering procedures without color, one point-based and another mesh-based, and a third one with color, notably a point-based procedure. Since the latter rendering procedure contains a good variety of colored PCs with coding distortion artifacts, this portion of the dataset was chosen to obtain the target weights through regression. Regarding the colored PCs, it is important to note that, in the human assessed PCs, only the geometry is coded, while the color was not coded, i.e. the original PC color is used. However, since the reconstructed geometry differs from the original, due to lossy geometry coding, a recoloring procedure was performed, thus implying also some color distortion regarding the original/reference color since some geometry points may have changed position.

The purpose of this set of decoded PCs is to measure the subjective impact of the masking effect that good color has on geometry, hence the decision to choose this dataset to develop the PC joint distortion metric to be used in the DL-based codec training loss function. This dataset contains subjective Mean Opinion Scores (MOS) for 54 different stimuli, i.e. PCs coded at various rates with different codecs, which are computed by averaging the scores of the set of users for each stimulus. In particular, the IRPC-D dataset is characterized by:

- **PCs**: *Façade9*, *Frog*, *House without Roof*, *Longdress*, *Loot*, and *Egyptian Mask* (this last one was not considered due to outlier behaviour).
- **PC codecs**: Three different geometry coding solutions, notably: V-PCC standard using a patch-based projection approach, reference software version 2 [19]; G-PCC Trisoup, reference software version 1.1 [20]; Octree-based PCL codec, version v1.8 [21].
- **Coding rates**: Each PC is coded with three different, non-specified rates for each codec: one high rate, one middle, and one low rate, aiming to obtain three different perceptual qualities for each given PC coded with a particular codec.

The subjective assessment of the stimuli was performed by 20 subjects using a double stimulus quality assessment protocol, meaning that, for each stimulus, each subject had to score the impairments in the distorted PC using as reference the original PC by attributing a score between 1 and 5, with the following correspondence: 1: Very annoying, 2: Annoying, 3: Slightly annoying, 4: Perceptible, but not annoying, and 5: Imperceptible.

After each subject had assessed a given decoded PC, the mean of all subjective opinion scores was computed, i.e. the MOS, to attenuate the effect of outlier scores for a given decoded PC. This MOS score is the subjective quality score used to represent the subjective quality of a given stimulus and, thus, will be used in the regression process to design the target PC joint distortion metric.

It is important to note that the goal is to design a distortion metric for the training loss function and not a quality metric, thus a subjective distortion score has to be obtained from the subjective quality scores, i.e. MOS. Taking into account that the minimum and maximum quality scores were 1 and 5, respectively, subjective distortion scores were obtained from the subjective quality scores by considering the MOS complement to 5, i.e. Distortion=(5-MOS). As such, the minimum distortion score will correspond to 0 (when the quality score is maximum, i.e., 5) and the maximum distortion score to 4 (when the quality score is minimum, i.e., 1). This mapping will be useful for the regression since the loss function includes distortion and not quality scores.

C. Distortion Metrics: Proxy Metrics and Weight Transfer

As mentioned in Section III, while designing a differentiable distortion metric for the training loss function, the geometry distortion metrics associated to the most popular quality metrics, like PSNR D1 and PSNR D2, could not be used since the D1 and D2 distances are not differentiable. The same happens with MSE YUV and MSE RGB color metrics and with state-of-the-art joint quality metrics such as PCQM [14] and PointSSIM [16], which cannot be used for the same reason.

1) Training Differentiable Distortion Metrics

As described in Section III.B, the list of potential differentiable geometry and color metrics considered for the loss function is the FL, for geometry, and the SV-MSE RGB, Y and YUV, for color.

2) Proxy (Non-Differentiable) Geometry Distortion Metrics

To perform the regression between the to be designed PC joint distortion metric and the subjective scores for the IRPC-D stimuli, it is necessary to assess the PC objective quality with some selected distortion metric, both for geometry and color. These distortion metrics will be fitted to the 5-MOS subjective distortion scores to derive weights for the corresponding metrics (geometry and color) in the training loss function.

Since the IST-JPCC codec measures the training geometry distortion with the FL function, ideally this function would be used in the regression procedure as well. However, unfortunately, the FL function cannot be directly used as a geometry distortion metric between the original and decoded PCs because it is, in fact, not a distortion metric between two PC geometries where the voxels are either filled or empty but rather a binary classification metric that computes the logarithm of the probabilities of a voxel-being filled. In fact, since all voxels of the decoded PC in the IRPC-D dataset are already binarized into 0s and 1s, the FL function cannot even be applied to assess the geometry quality.

In this context, the solution is to select some (non-differentiable) geometry distortion metrics which may act as proxies for the (differentiable) FL function and transfer the weights obtained by regression with the first (non-differentiable) metrics to the second (differentiable) metric. The natural candidates to become proxy metrics for FL and perform the regression are the largely used MSE D1 and MSE D2 geometry distortion metrics.

3) Proxy (Non-Differentiable) Color Distortion Metrics

Since the same dilemma exists for color, the MSE RGB, YUV and Y have been selected as the proxy color distortion metrics for the SV-MSE color distortion metrics, since they can be directly applied to binarized PCs. Keep in mind that, for example, MSE Y is different from SV-MSE Y since MSE Y computes color differences between one point in a PC and the closest point in another PC, while SV-MSE Y computes color differences between collocated points in the two PCs (during training) to avoid differentiability problems.

4) Proxy (Non-Differentiable) Joint Geometry and Color Distortion Metrics

To define the proxy joint distortion metrics, one of the selected proxy geometry distortion metrics has to be combined, using a specific weight, with one of the proxy color distortion metrics. While the case of combining two color distortion metrics at the same time, e.g. MSE RGB and MSE YUV, with a geometry distortion metric has been considered, their final correlation was not better, thus the selected proxy joint distortion metrics will only consider one proxy geometry and one proxy color distortion metric at a time. In summary, six proxy joint distortion metrics were considered, by combining MSE D1 and D2 with MSE RGB, YUV and Y. For each proxy joint distortion metric, the appropriate weights will be determined by regression using the ground truth subjective scores in the IRPC-D dataset [18].

D. Regression Model and Procedure

To obtain the geometry and color weights for the proxy joint distortion metrics, a logistic regression model was adopted [22]. The MOS values were first normalized as in [23]. The objective metric scores for each individual metric in the joint combination were also normalized, using their mean value. In total, six different sets of weights were obtained with this fitting procedure corresponding to each of the six proxy joint distortion metrics listed in Section IV.C.

E. Correlation Performance Assessment and Weights Selection

This section will report the correlation performance for the six fitted models obtained with the procedure described above, as well as the selection process for the final weights to be used in the training loss function.

1) Error and Correlation Performance Assessment Metrics

To assess the correlation performance of the obtained proxy joint distortion metrics with the IRPC-D subjective scores, three performance metrics will be used, notably:

i) Mean squared error between the distortion scores predicted by the proxy objective joint distortion metric and the ground truth subjective distortion scores.

ii) Pearson correlation coefficient assessing the linearity between the distortion scores predicted by the proxy objective joint distortion metric and the ground truth subjective distortion scores, ideally close to 1.

iii) Spearman correlation coefficient assessing the rank order variation between the distortion scores predicted by the proxy objective joint distortion metric and the ground truth subjective distortion scores, ideally close to 1.

Ideally, a good proxy joint distortion metric should assure as much as possible a small error, while granting a linear relationship and an unchanged rank order between the predicted and ground truth subjective scores.

2) Distortion Metrics Performance and Weight Selection

The performance for the various individual distortion metrics and proxy joint distortion metrics is included in Table I. For the proxy joint distortion metrics, the corresponding geometry-color weight pairs for the fitted models are also included. The original weights obtained through regression were processed to be complementary to 1 for geometry and color; this not only expresses their relative weights in a more intuitive way, but it also allows them to be directly used in the distortion component of the training loss function. The results in Table I allow concluding:

- For the individual distortion metrics, the geometry metrics display an error and correlation performance similar to the color distortion metrics.
- Individually, MSE D2 performs better than MSE D1 and MSE RGB performs better than MSE YUV. MSE Y is the color metric that performs the worse, overall.
- For the joint metrics, the metrics with MSE D2 perform better than the metrics with MSE D1, as shown by the error/correlation scores. Moreover, the weights associated with MSE D2 are larger than with MSE D1 and even larger than the corresponding color weights, something that does not happen for the MSE D1 weights.
- For the joint metrics, MSE YUV slightly edged out MSE RGB and Y in terms of performance, when combined with a geometry distortion metric.

Taking into account that all the considered non-differentiable objective distortion metrics used in the fitting process act solely as proxies for the differentiable distortion metrics used for training in the loss function, an extra step was performed in order to obtain a more robust set of weights. Since the largest weight variations are related to the geometry distortion metrics used in the fitting, notably MSE D1 and MSE D2, and the variations corresponding to the color metrics are not that large, it was decided to define two weight pairs: one driven by MSE D1 and another by MSE D2. For each of these cases, the geometry and color weights are computed as the mean of the weights associated to the three color metrics, MSE Y, RGB and YUV. This step reduces the probability of overfitting the obtained weights to a certain proxy distortion metric combination. The weights obtained after this procedure for the differentiable PC joint geometry and color distortion metrics to be used in the IST-JPCC codec loss function are (see Table II):

- **MSE D1 & MSE Color pair**: 0.563 for the SV-MSE RGB or SV-MSE YUV color distortion metric and 0.437 weight for the FL geometry distortion metric.
- **MSE D2 & MSE Color pair**: 0.432 for the SV-MSE RGB or SV-MSE YUV color distortion metric and 0.568 weight for the FL geometry distortion metric.

Interestingly, while the first pair gives more weight to the color, the second pair gives more weight to the geometry, in accordance with the fact that MSE D2 is better correlated with the subjective scores than MSE D1.

V. RD PERFORMANCE ASSESSMENT

This section reports the RD performance gains for the IST-JPCC codec when using the proposed perceptually-driven differentiable PC joint geometry and color distortion.

A. Datasets and Coding Conditions

For the performed experiments, the following PCs from the JPEG CTTC dataset [15] have been used: *Statue Klimt*, *House without Roof*, *Bumbameuboi*, *Longdress*, *Guanyin*, *Phil*, *Rhetorician*, *Ricardo*, and *Romanoillamp*. These PCs have very different characteristics, what is essential to have a meaningful RD performance assessment.

TABLE I
ERROR AND CORRELATION PERFORMANCE OF INDIVIDUAL AND PROXY JOINT DISTORTION METRIC COMBINATION

Metrics (MSE)	(5 – MOS) / 4				
	Error	Correlation		Weight	
	MSE	Pearson	Spearman	Geometry	Color
D1	0.032	0.698	0.654		
D2	0.030	0.724	0.658		
Y	0.031	0.708	0.677		
YUV	0.030	0.719	0.674		
RGB	0.030	0.724	0.701		
D1 & Y	0.027	0.753	0.711	0.466	0.534
D1 & YUV	0.025	0.777	0.721	0.433	0.567
D1 & RGB	0.026	0.762	0.728	0.412	0.588
D2 & Y	0.026	0.764	0.719	0.596	0.404
D2 & YUV	**0.024**	**0.783**	**0.728**	0.552	0.448
D2 & RGB	0.026	0.770	0.718	0.557	0.443

TABLE II
MEAN OF ERROR AND CORRELATION PERFORMANCE FOR EACH PROXY JOINT DISTORTION METRIC

Metrics (MSE)	(5 – MOS) / 4				
	Error	Correlation		Weight	
	MSE	Pearson	Spearman	Geometry	Color
D1 & Color	0.026	0.764	0.720	**0.437**	**0.563**
D2 & Color	0.025	0.772	0.722	**0.568**	**0.432**

B. Quality Metrics

The following quality metrics were selected, following the JPEG CTTC recommendations [15]:

- **Geometry**: PSNR D1 and PSNR D2 are used to assess the geometry quality of the decoded PCs, namely the proximity of the decoded points to the original ones.
- **Color**: PSNR Y and PSNR YUV are used to assess the color quality of the decoded PC, both in terms of luminance and chrominance.
- **Joint Geometry and Color**: The 1-PCQM quality metric is used to jointly assess the quality of the color and geometry of the decoded PC. This is the key quality metric for this paper for which gains are targeted since it is the one which best correlates to the users' perceptual quality, independently of the individual geometry and color qualities. Thus, the goal is to find the best geometry-color balance to maximize the final joint quality and not each quality component individually.

To assess the RD performance gains/losses between codecs, the Bjontegaard Delta (BD)-Rate and BD-Metric metrics will be used. For BD-Rate, negative values are welcome as they correspond to rate reductions for the same quality while the opposite happens for BD-Metric since positive values correspond to quality metric gains for the same rate.

C. Codecs Under Comparisons

The IT-DL-PCC-GC codec, an extension of the IST-JPCC coding solution using an additional post-processing super-resolution procedure, has been submitted to the recent JPEG Pleno PCC Call for Proposals [11] and selected as the best performing, in July 2022, with an average rate reduction over G-PCC Octree and Predlift (geometry and color coding) of

TABLE III
BD-RATE FOR VARIOUS QUALITY METRICS CONSIDERING THE CANDIDATE WEIGHTS AND USING AS REFERENCE THE BALANCED WEIGHTS.

Color Weight	Point Cloud	SV-MSE RGB					SV-MSE YUV				
		PSNR D1	PSNR D2	PSNR Y	PSNR YUV	1-PCQM	PSNR D1	PSNR D2	PSNR Y	PSNR YUV	1-PCQM
0.432	*StatueKlimt*	-11.711%	-11.084%	-8.782%	-9.450%	-11.942%	-5.996%	-7.479%	7.801%	6.690%	0.773%
	HouseWoRoof	-2.605%	-3.889%	11.439%	13.598%	2.772%	-7.332%	-8.578%	4.176%	8.357%	1.283%
	Bumbameuboi	12.153%	-0.020%	8.470%	3.762%	7.546%	-1.762%	-4.460%	-18.666%	-13.022%	-16.727%
	Guanyin	-14.278%	-14.895%	4.412%	5.159%	10.223%	-8.886%	-8.800%	14.928%	12.853%	13.058%
	Longdress	-11.921%	-11.983%	10.324%	9.568%	14.629%	-10.829%	-10.748%	16.860%	16.161%	11.948%
	Phil	-10.277%	-10.537%	4.660%	5.007%	2.443%	-6.849%	-8.074%	15.116%	15.114%	8.376%
	Rhetorician	-14.245%	-15.157%	4.602%	4.874%	0.912%	-6.673%	-6.655%	30.611%	23.655%	22.857%
	Ricardo	-5.784%	-5.622%	9.030%	4.450%	-0.467%	-6.783%	-8.460%	13.168%	9.026%	5.678%
	Romanoillamp	-16.390%	-16.300%	-4.813%	-2.723%	-7.895%	-10.544%	-4.964%	14.735%	12.639%	8.116%
	Average	-8.340%	-9.943%	4.371%	3.805%	2.025%	-7.295%	-7.580%	10.970%	10.164%	6.151%
0.563	*StatueKlimt*	8.313%	9.071%	-15.021%	-17.001%	-20.310%	5.443%	8.770%	-12.318%	-20.703%	-0.685%
	HouseWoRoof	-0.871%	-0.600%	-7.316%	-8.745%	-4.387%	1.118%	1.965%	-4.657%	-14.385%	-1.830%
	Bumbameuboi	-4.194%	20.740%	-6.915%	-8.962%	-8.335%	7.162%	20.944%	-11.679%	-12.937%	-17.331%
	Guanyin	18.753%	20.112%	-0.823%	1.967%	10.942%	9.437%	9.420%	-9.146%	-10.836%	-1.495%
	Longdress	11.138%	11.322%	-1.688%	0.797%	4.028%	12.628%	12.719%	-7.049%	-11.811%	0.788%
	Phil	6.229%	7.389%	1.333%	0.991%	4.475%	10.159%	9.696%	-6.677%	-9.510%	1.310%
	Rhetorician	8.220%	8.496%	-2.938%	-2.440%	0.043%	4.836%	4.966%	-9.454%	-13.022%	-3.574%
	Ricardo	4.688%	4.844%	-17.765%	-17.116%	-5.131%	16.659%	7.008%	-9.775%	-5.791%	-0.482%
	Romanoillamp	8.289%	9.114%	-8.069%	-9.712%	-7.034%	5.222%	6.464%	-9.846%	-11.757%	-3.235%
	Average	**6.729%**	**10.054%**	**-6.578%**	**-6.691%**	**-2.857%**	**8.074%**	**9.106%**	**-8.956%**	**-12.306%**	**-2.948%**

around 84% (measured as BD-Rate) for the JPEG provided test dataset [10]. This IT-DL-PCC-GC codec was trained using a balanced weighted loss function. Considering this, the target now is to assess the gains/losses of using the proposed perceptually-driven weights regarding the balanced weighted loss function. To do so, the performance of the presented IST-JPCC coding solution trained using the perceptually-driven weights will be compared with the performance of the IST-JPCC coding solution trained using the balanced weighted loss function. In this context, the codecs under comparison are:

- IST-JPCC using either SV-MSE RGB or SV-MSE YUV in the loss function and a 0.5 weight for both the geometry and color distortions; this is the reference configuration to compute the BD-Rate and BD-Metric.
- IST-JPCC using either SV-MSE RGB or SV-MSE YUV in the loss function and 0.432 weight for the color distortion (thus 0.568 for the geometry distortion).
- IST-JPCC using either SV-MSE RGB or SV-MSE YUV in the loss function and 0.563 weight for the color distortion (thus 0.437 for the geometry distortion).

IST-JPCC using SV-MSE Y in the loss function ended up not being considered since it produced decoded PCs with very poor subjective quality, i.e. very washed-out, almost gray-scaled colors, due to ignoring the chrominance components during training.

D. RD Performance: Balanced versus Unbalanced Weights

Table III shows the RD performance of the designed perceptually-driven differentiable PC joint distortion metrics, taking as reference the IST-JPCC codec with balanced 0.5 weights. The main conclusions from the results are:

- IST-JPCC shows, on average, geometry rate losses and color and (1-PCQM) gains, for both the SV-MSE RGB and SV-MSE YUV training distortion metrics, when using the 0.563 color weight for the training function. This was expected since the color is now given more importance compared to the 0.5 balanced weights. The rate reductions for the same quality amount to around 3% considering the joint quality metric.
- IST-JPCC shows, on average, geometry rate gains and color and (1-PCQM) losses, for both the SV-MSE RGB and SV-MSE YUV training distortion metrics, when using the 0.432 color weight for the training function. This was expected since the color is now given less weight compared to the geometry regarding the 0.5 weights.
- The trends above seem to confirm that color quality plays a more important role than geometry quality in the overall perceptual quality of the decoded PC. When observing a decoded PC, compression artifacts in the decoded geometry may be masked by decoded color with good quality. Thus more weight should be given to the color distortion metric in the training process, thus stimulating its distortion reduction. This effect happens both for the SV-MSE RGB and SV-MSE YUV training distortion metrics.

In summary, the proposed perceptually-driven weights design in the PC joint distortion metric has shown its power since a rate reduction of 3% regarding the balanced weights could be achieved for the same joint quality as measured by the (1-PCQM) objective quality metric when using a 0.563 color weight, thus taking benefit from the color masking effect. While this gain may not look that large, it is well above the threshold that usually determines the adoption of a new tool in a codec in a standardization context. This happens at no added complexity cost compared to the 0.5 weight.

VI. FINAL REMARKS

This paper proposes an original, perceptually-driven design for the differentiable PC joint geometry and color

distortion metric to be used for training a DL-based joint geometry and color codec. By appropriately adjusting the geometry versus color distortion weights at training time, a rate reduction of around 3% may be obtained at no additional complexity cost. Future work will consider differentiable distortion metrics which are not single-voxel based and can measure the PC quality for training purposes considering the context of each voxel, since it is well known that the final subjective quality impact of geometry and color artifacts depends on the neighborhood where these artifacts happen.

REFERENCES

[1] D. Graziosi *et al.*, "An Overview of Ongoing Point Cloud Compression Standardization Activities: Video-based (V-PCC) and Geometry-based (G-PCC)", *APSIPA Trans. Signal and Inf. Process.*, vol. 9, Apr. 2020.

[2] H. Liu *et al.*, "A Comprehensive Study and Comparison of Core Technologies for MPEG 3-D Point Cloud Compression," *IEEE Trans. Broadcast.*, vol. 66, no. 3, pp. 701-717, Sep. 2020.

[3] G. J. Sullivan *et al.*, "Overview of the High Efficiency Video Coding (HEVC) Standard," *IEEE Trans. Circuits Sys. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] B. Bross *et al.*, "Overview of the Versatile Video Coding (VVC) Standard and its Applications," *IEEE Trans. Circuits Sys. Video Technol.*, vol. 31, no. 10, pp. 3736-3764, Oct. 2021.

[5] A. F. R. Guarda, N. M. M. Rodrigues and F. Pereira, "Adaptive Deep Learning-Based Point Cloud Geometry Coding," *IEEE J. Sel. Topics Signal Process.*, vol. 15, no. 2, pp. 415–430, Feb. 2021.

[6] T. Lin *et al.*, "Focal Loss for Dense Object Detection," *IEEE Int. Conf. Comput. Vision*, Venice, Italy, Oct. 2017.

[7] M. Quach, G. Valenzise and F. Dufaux, "Learning Convolutional Transforms for Lossy Point Cloud Geometry Compression," *IEEE Int. Conf. on Image Process.*, Taipe, Taiwan, Sep. 2019.

[8] J. Wang, D. Ding, Z. Li and Z. Ma, "Multiscale Point Cloud Geometry Compression", *Data Compression Conf.*, Virtual, Mar. 2021.

[9] E. Alexiou, K. Tung and T. Ebrahimi, "Towards Neural Network Approaches for Point Cloud Compression," *SPIE Appl. Of Digit. Image Process. XLIII*, vol. 11510, pp. 18-37, Aug. 2020.

[10] A. F. R. Guarda *et al.*, "IT/IST/IPLeiria Response to the Call for Proposals on JPEG Pleno Point Cloud Coding," *arXiv:2208.02716 [eess.IV]*, Aug. 2022.

[11] ISO/IEC JTC1/SC29/WG1 N100097, "Final Call for Proposals on JPEG Pleno Point Cloud Coding", Online, Jan. 2022.

[12] C. Szegedy *et al.*, "Inception-v4, Inception-ResNet and the Impact of Residual Connections on Learning," *AAAI Conf. on Artif. Intell.*, San Francisco, CA, USA, Feb. 2017.

[13] M. Ruivo, A. Guarda and F. Pereira, "Double-Deep Learning-Based Point Cloud Geometry Coding with Adaptive Super-Resolution", *Eur. Workshop on Vis. Inf. Process.*, Lisbon, Portugal, Sep. 2022.

[14] G. Meynet, Y. Nehmé, J. Digne and G. Lavoué, "PCQM: A Full-Reference Quality Metric for Colored 3D Point Clouds", *Int. Conf. on Qual. Multimedia Experience*, Athlone, Ireland, May 2020.

[15] ISO/IEC JTC 1/SC29/WG1 N100112, "Common Training and Test Conditions for Point Cloud Compression.", Online Meeting, Jan. 2022.

[16] E. Alexiou and T. Ebrahimi, "Towards a Point Cloud Structural Similarity Metric", *IEEE Int. Conf. on Multimedia & Expo Workshops*, London, United Kingdom, June 2020.

[17] E. Alexiou *et al.*, "A Comprehensive Study of the Rate-Distortion Performance in MPEG Point Cloud Compression", *APSIPA Trans. Signal and Inf. Process.*, vol. 8, no. 1, p. e27, Nov. 2019.

[18] A. Javaheri, C. Brites, F. Pereira and J. Ascenso, "Point Cloud Rendering After Coding: Impacts on Subjective and Objective Quality," *IEEE Trans. Multimedia*, vol. 23, pp. 4049-4064, Nov. 2020.

[19] ISO/IEC JTC1/SC29/WG11, "PCC Test Model Category 2 vo", Doc. N17248, Macau, China, Oct. 2017.

[20] ISO/IEC JTC1/SC29/WG11, "G-PCC Codec Description v2", Doc. N18189, Marrakech, Morocco, Jan. 2019.

[21] R. B. Radu and S. Cousins, "3D is Here: Point Cloud Library (PCL)," *IEEE Int. Conf. Robot. Automat.*, Shanghai, China, May 2011.

[22] A. M. Rohaly *et al.*, "Video Quality Experts Group: Current Results and Future Directions," *SPIE Visual Commun. and Image Process.*, vol. 4067, pp. 742-753, May 2000.

[23] I. Viola and P. Cesar, "A Reduced Reference Metric for Visual Quality Evaluation of Point Cloud Contents", *IEEE Signal Process. Letters*, vol. 27, pp. 1660-1664, Sep. 2020.