GEOMETRIC DISTORTION METRICS FOR POINT CLOUD COMPRESSION

Dong Tian, Hideaki Ochimizu, Chen Feng, Robert Cohen, Anthony Vetro

Mitsubishi Electric Research Laboratories, Cambridge, MA 02139, USA

ABSTRACT

It is challenging to measure the geometry distortion of point cloud introduced by point cloud compression. Conventionally, the errors between point clouds are measured in terms of point-to-point or point-to-surface distances, that either ignores the surface structures or heavily tends to rely on specific surface reconstructions. To overcome these drawbacks, we propose using point-to-plane distances as a measure of geometric distortions on point cloud compression. The intrinsic resolution of the point clouds is proposed as a normalizer to convert the mean square errors to PSNR numbers. In addition, the perceived local planes are investigated at different scales of the point cloud. Finally, the proposed metric is independent of the size of the point cloud and rather reveals the geometric fidelity of the point cloud. From experiments, we demonstrate that our method could better track the perceived quality than the point-to-point approach while requires limited computations.

Index Terms— 3D point cloud, quality measurements, point-to-point distortion, point-to-plane distortion

1. INTRODUCTION

With the growth of 3D sensing technologies, 3D point clouds have become an important and practical representation of 3D objects and surrounding environments in many applications, such as virtual reality, mobile mapping, scanning of historical artifacts, 3D printing and digital elevation models [1]. A key challenge in 3D point cloud processing is the handling of a huge amount of data. To address this, compression of the point cloud data is needed. In this context, measuring the errors introduced from compression becomes necessary. That is, we need to evaluate the quality degradation of a compressed 3D point cloud relative to the incoming 3D point cloud.

To solve this problem, point-to-point distances have been considered [2]. First, for every point in one point cloud, a corresponding point from the other point cloud is identified. Then the average or maximum of the Euclidean distances between such point pairs is used as the basis for a measurement. We refer to this approach as point-to-point or cloud-to-cloud distances (C2C). A notable drawback of this approach is that it fails to account for the fact that points in a point cloud often represent surfaces in the structure.

In order to overcome the drawback of the point-to-point approach, another known approach point-to-surface or cloud-to-mesh (C2M) [3] was studied. It first constructs a mesh or model from the reference point cloud. Then, the distances from each point in the processed point cloud to that mesh are computed and used as basis for C2M metrics. This method has been incorporated into some point cloud processing software, like CloudCompare [4]. Unfortunately, as it heavily depends on the method to construct the mesh or model from a point cloud, that often need special tuning on different target point clouds, it is hard to deploy this approach for point cloud compression.

As the point clouds often need to be rendered before being viewed, another potential way for the evaluation could be based on projections to an image plane. That is, the projected view from the uncompressed point cloud serves as a reference. Then, the compressed point cloud would be projected to the same image plane and a distortion could be computed. However, this may be a good approach to measure the distortions on the associated attributes, e.g. RGB/YUV components, it is not straightforward to measure geometric distortion in the projected views.

In this paper, we consider a point-to-plane, or named as cloud-to-plane (C2P) approach, that falls in between the point-to-point and point-to-surface methods. While integrating local plane properties and tracking visual qualities, the proposed point-to-plane metric is less dependent on a complex surface construction and can be deployed in practical.

2. PROPOSED POINT-TO-PLANE METRICS

2.1. Principles of Point-to-Plane Distances

Let $\mathbf{A}$ and $\mathbf{B}$ denote the original and the compressed point cloud, respectively. We are to evaluate the compression errors in point cloud $\mathbf{B}$ relative to the original point cloud $\mathbf{A}$. This would be achieved by a two-pass computation. In each pass, we select one point cloud as reference, e.g. when computing $e_{\mathbf{A},\mathbf{B}}$, $\mathbf{A}$ serves as a reference. In the end, we will select a worse metric among $e_{\mathbf{A},\mathbf{B}}$ and $e_{\mathbf{B},\mathbf{A}}$ as the final measurement.

As illustrated in Fig. 1, the steps of computing point-to-point distance $e^{c2c}_{\mathbf{A},\mathbf{B}}$ and the proposed point-to-plane distance $e^{c2p}_{\mathbf{A},\mathbf{B}}$ are elaborated as follows.

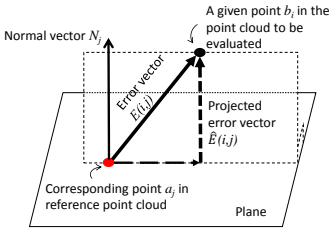


Fig. 1: Point-to-point distance vs. point-to-plane distance.

1) For each point a_j in point cloud $\mathbf{A}$, i.e., the red point in the figure, identify a corresponding point b_i in point cloud $\mathbf{B}$, i.e. the black point in the figure. The nearest neighbor is used to locate the corresponding point.

2) Take the unit normal vector N_j on point a_j in the reference point cloud $\mathbf{A}$, if available. Otherwise, the normal vector would be estimated on the fly using a state-of-the-art method.

3) Compute the error vector $E(i,j)$ by connecting point a_j to point b_i. The length of the error vector would actually lead to the conventional point-to-point error, i.e.,

$$e^{c2c}_{\mathbf{A},\mathbf{B}} = \frac{1}{N_A} \sum_{\forall a_j \in \mathbf{A}} \|E(i,j)\|_2^2. \tag{1}$$

where N_A is the number of points in point cloud $\mathbf{A}$.

4) To get proposed point-to-plane errors, we project the error vector $E(i,j)$ along the normal direction N_j to get a new error vector $\hat{E}(i,j)$. The point-to-plane error is finally computed as,

$$e^{c2p}_{\mathbf{A},\mathbf{B}} = \frac{1}{N_A} \sum_{\forall a_j \in \mathbf{A}} (E(i,j) \cdot N_j)^2. \tag{2}$$

Compared to point-to-point distances, the proposed point-to-plane distances measure projected error vectors along normal directions rather than measuring the original error vectors directly, that would impose larger penalty on the errors that move more away from the local plane surface. For point clouds characterized by surfaces of structures, such point-to-plane distances is better aligned with the perceived quality than point-to-point.

As the point-to-plane distances rely on the normal vector associated with each point, concerns may arise from the normal accuracy near to edges. In this situation, a hybrid measurement could help alleviating the issues, where point-to-point distances would be still used for the points with unreliable normal vectors and point-to-plane distances is used only with points with reliable normals. In practice, however, the overall impact from edge points is negligible because such points only occupy a quite small percentage in a point cloud. Hence, in this work, we won't consider a hybrid metric.

Note that similar error metrics have been studied in the context of surface registration, e.g., Iterative Closest Point (ICP) algorithms [5, 6] for faster convergence. One contribution of this paper is to apply this method for evaluation of the point cloud geometric errors introduced from compression and discover its capability to follow perceptual qualities.

2.2. Peak Signal to Noise Ratio (PSNR)

The mean square error (MSE) proposed so far could be reported as a metric and sometimes it is preferred because it carries the physical units in 3D space. However, people often found it hard to understand the MSE's between multiple point clouds. Hence, converting the MSE's into $PSNR$ numbers would normalize the metrics with respect to a peak value p,

$$PSNR_{\mathbf{A},\mathbf{B}} = 10 \log_{10} \frac{p^2}{e_{\mathbf{A},\mathbf{B}}}, \tag{3}$$

where the value p is to be selected.

Conventionally, the diagonal distance of a bounding box of the point cloud is used to define the peak value p. One obvious disadvantage is that, given the same amount of error for each point, a point cloud from a larger object would produce a higher $PSNR$ number than a point cloud from a smaller object. However, their fidelity should be the same since each point has the same distortions.

Furthermore, since the perceptual qualities are of our interest, we need to fix certain viewing setting-up for all contents to be inspected. That is, a viewing box need to be fixed in all (x, y, z) directions. Let the bounding box of two point clouds denoted as (x_i, y_i, z_i), with $i = 1, 2$. Note that they are not from a single point cloud with different qualities, but two subjects to be studied. To simplify the interpretation, we assume, for the time being, that the distance between neighboring points within one point cloud is a constant $d^{(i)}$, with $i = 1, 2$, and $d^{(1)} \neq d^{(2)}$. Parameter $d^{(i)}$ actually indicates the intrinsic resolution of an input point cloud in the acquisition domain. When we pick variable p based on the bounding box (x_i, y_i, z_i), it means that we are fitting the content from (x_i, y_i, z_i) to (x, y, z), and an overall scaling is likely to be necessary. Even if the ratios between (x_i, y_i, z_i) are kept unchanged during the scaling to avoid geometric deformations, we are certainly to check the geometric details, represented in the original data, in an unfair way for the two point clouds, because the neighboring points rendered in the viewing box (x, y, z) would be apart from each other at different distances for the two subjects.

In order to address the above issues, we propose using the intrinsic resolution $d^{(i)}$ of the input point cloud instead of its bounding box to define the variable p. In that way, the neighboring points from different point clouds would be put at the same distance in the viewing box. In practice, the intrinsic resolution $d^{(i)}$ would be derived from the nearest neighbor distances for all points a_k in the input point cloud, denoted as d_k. Statistical parameters from d_k is proposed to determine the intrinsic resolution. In an advanced choice, one may use the most probable value as the intrinsic resolution via a histogram analysis. In this work, we simply choose

the maximum value due to its simplicity. Hence, we have $p = d^{(i)} = \max_{\forall a_k \in \mathbf{A}}(d_k)$.

The above proposed way to choose p value based on the intrinsic resolution seems to make the $PSNR$ be dependent on the input data. However, since we are to evaluate the qualities in a rendering space rather than the acquisition space, we should focus on the viewing process. By setting the peak value p as per the intrinsic resolution, it actually intends to normalize the MSE's according to the native resolution during the rendering, i.e., the viewing conditions. Different choice of statistic parameters in d_k would lead to some minor adjustment for viewing. With maximum value being selected, the neighboring points with a smaller distance would be mixed to some extent. With minimum value, the neighboring points with a bigger distance would allow some gaps between them or their point sizes are adjusted to fill up the gaps.

In the end, $\sqrt{e_{\mathbf{A},\mathbf{B}}}/p$ in $PSNR$ computation represents a normalized error with respect to a fixed rendering resolution, that is decoupled from the acquisition.

3. PERCEIVED NORMALS AT DIFFERENT SCALES

With a point cloud compression system, we often need to deal with a decoded point cloud having fewer points than in the original point cloud. This may be a consequence from a scalable coding scenario, where the decoded number of points progressively increases with more layers being decoded. Hence, the decoded point cloud may be only a subset of points in the original point cloud. The approach proposed in Section 2, however, does not account for cases when many points may be missing intentionally.

The normal vectors associated with each point in an original point cloud is regarded as a representation of the local surface when the viewer sits from a close distance to the point cloud. When the point cloud becomes sparser, each point would represent a larger space in 3D world. This mimics a procedure that the viewer is moving away from the point cloud. As a result, the perceived surface from the same point is becoming different. This can be illustrated by the viewing experience on a rock in a hill slope in Fig. 2. When we are close to the rock, we could differentiate the points on the rock, and hence there are different normal directions for points on the rock representing the rock's shape. When we are far away from the hill, the perceived normal from the rock becomes a single direction representing the hill slope instead of the rock shape, even the sampled point physically belongs to the rock.

Now we come up with a proposed refinement when computing $e_{\mathbf{A},\mathbf{B}}$ as in Section 2. After an error vector is generated for each point from $\mathbf{A}$, the error vector will be projected along a normal vector that represents the local plane in $\mathbf{B}$. We propose a way to derive the normal vector representing the perceived local plane in $\mathbf{B}$ that follows the principles described previously.We want to avoid estimating normal vectors solely based on points in $\mathbf{B}$ because such estimation is likely to be

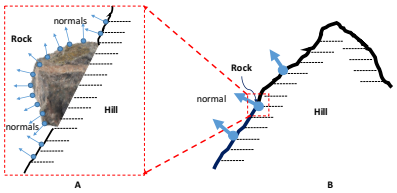


Fig. 2: Rock on a hill slope. The different perceived normals at different scales /viewing distances. Left: Closer view. Right: Further view.

biased due to geometric distortions introduced in $\mathbf{B}$. We also want to avoid directly using the normal vector carried from the corresponding point in $\mathbf{A}$ because it does not represent the "perceived" local plane in $\mathbf{B}$. Instead, we propose to cluster the points in $\mathbf{A}$ using the points in $\mathbf{B}$ as centroids, then the normal vectors from the points within the same cluster in $\mathbf{A}$ will be averaged as the "perceived" normal at the current viewing distance. The clustering can be accomplished via determining nearest neighbors in $\mathbf{B}$. All points in $\mathbf{A}$ sharing the same nearest neighbor in $\mathbf{B}$ are clustered together.

4. EXPERIMENTS

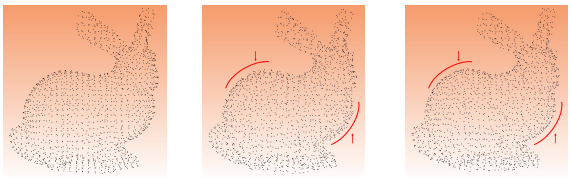
In order to validate the point-to-plane metric in context of point cloud compression systems, we conducted two sets of experiments in this section. The first experiment intends to simulate different types coding errors, while the second experiment evaluates subsampling schemes that may be utilized with a scalable compression system. Point-to-point is used as a benchmark metric. k-NN approach is used to estimate normals, with $k = 12$. $PSNR$ values will be computed using Eq. 3.

4.1. Evaluations on Random Errors vs. Surface Errors

We first simulated different types of displacement errors on the point locations. Such errors may be introduced by different quantization methods.

Two types of errors were investigated. Type 1 errors move the points in an arbitrary 3D direction in a random way, and is called *Random Errors*. With type 2 errors, the points are shifted approximately within a local plane, named as *Surface Errors*. Both displacement errors are bounded with the same magnitude and are applied on all points in a point cloud.

Due to lack of space, we only show the snapshots from Bunny point cloud [7] in Fig. 3. CloudCompare is used for the rendering [4]. Basically, we perceive more noises with respect to the contours from the point cloud with Type 1 errors than the point cloud with Type 2 errors, for example, the contour in the front and the back of the Bunny as highlighted. Perceptually speaking, the latter with Type 2 errors shows less noises during viewing than the former errors.



Original (18.6dB, 24.6dB) (18.6dB, 43.6dB)

Fig. 3: Bunny with errors. From left to right: Original; Random error; Surface error. (point-to-point, point-to-plane) dB.

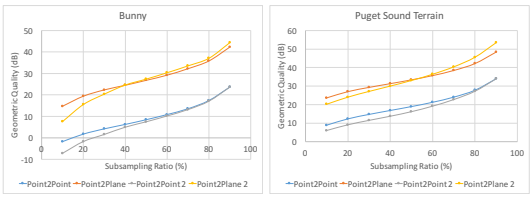


Fig. 4: Evaluation on sampling schemes. "Point2point" and "Point2plane": uniform sampling. "Point2point 2" and "Point2plane 2": contour-based sampling.

On the other hand, because of the same magnitude in both errors, their point-to-point metric reports the same number, 18.6dB. That is, point-to-point metric fails to differentiate the visual qualities. However, point-to-plane metric reports 24.6dB for Type 1 errors and 43.6dB for Type 2 errors, that clearly demonstrates consistent observation as their perceptual qualities.

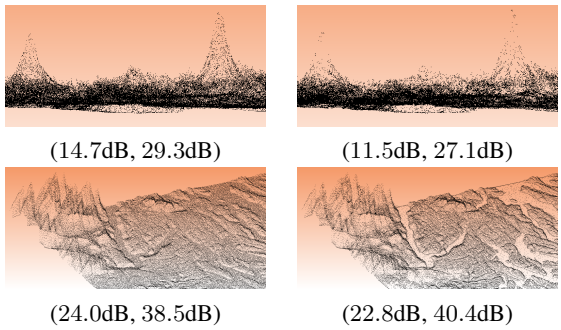
4.2. Evaluations on Different Subsampling Schemes

Subsampling scenarios play important roles in a scalable compression system. In this subsection, two subsampling schemes are to be evaluated. For scheme 1, the sampled points are selected in a uniform way, called *Uniform Sampling*. As comparison, in scheme 2, the points from edge or corner area are forced to have higher probabilities to be chosen [8], and named as *Contour-based Sampling*.

Fig. 4 shows the point-to-point and point-to-plane measures for the two subsampling schemes on two point clouds. The *Terrain* is provided by Lindstrom et.al [9]. The subsampling ratio ranges from 10% to 90%.

When looking at the point-to-point curves, uniform sampling is reported to be better than contour-based sampling across all subsampling ratios. On the contrary, the point-to-plane curves cross each other. In particular, according to point-to-plane curves, for low sampling ratios, uniform subsampling produces higher quality; while at high sampling ratios, contour-based subsampling instead is preferred.

Fig. 5 shows snapshots of Terrain point cloud at subsampling ratio of 30% and 70% with the two subsampling schemes. The $PSNR$ numbers using point-to-point and point-to-plane metrics are also listed. At ratio 30%, both point-to-point and point-to-plane indicates the uniform subsampling is better than contour subsampling. From the snapshots, it could be observed that uniform sampling looks better as it could better maintain the overall geometric shape of the objects, e.g. the hilltop. At ratio 70%, however, the observations from two metrics are different. Point-to-point prefers uniform sampling and point-to-plane prefers contour-based sampling. From the corresponding snapshots, it could be observed that both sampling could describe the overall geometric shape while the contour-based sampling is able to highlight the structure details. Hence, the contour-based sampling is preferred in terms of visual qualities, that is consistent to the observation from the proposed point-to-plane metric.



(14.7dB, 29.3dB) (11.5dB, 27.1dB)

(24.0dB, 38.5dB) (22.8dB, 40.4dB)

Fig. 5: Terrain. Top and Bottom, subsampling ratio are 30%, 70%. Left, Uniform sampling. Right, Contour-based sampling. (point-to-point, point-to-plane) dB.

Starting from very limited bit budget, uniform sampling strategy is preferred as the overall geometric information should be the first priority at the beginning. When more and more bit budget becomes available, contour-based sampling strategy should start taking effect to highlight the point cloud details. Using the proposed point-to-plane metric, it could lead to an adaptive sampling scheme for an overall best viewing experiences. A fully design for such a framework is subject to future work.

5. CONCLUSION

In this paper, we propose a point-to-plane measurement as an objective metric for geometric distortions from point cloud compression. The intrinsic geometry resolution is further proposed to be used to compute $PSNR$ numbers so as to normalize the errors. In addition, the perceived local planes and their normals were investigated that should rely on the scaling level of the point cloud. Comparing to the point-to-point metric, experiments demonstrate that the point-to-plane metric could better track visual qualities of a point cloud through a compression system. On the other hand, comparing with other complex metrics that account for surface structures, point-to-plane metric is able to capture the surface features while requiring low-complexity computations.

6. REFERENCES

[1] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, CN, May 2011.

[2] Daniel Girardeau-Montaut, Michel Roux, Raphaël Marc, and Guillaume Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, vol. 36, no. part 3, pp. W19, 2005.

[3] Paolo Cignoni, Claudio Rocchini, and Roberto Scopigno, "Metro: measuring error on simplified surfaces," *Computer Graphics Forum*, vol. 17, no. 2, pp. 167–174, 1998.

[4] Daniel Girardeau-Montaut, "CloudCompare-open source project," *OpenSource Project*, 2011.

[5] Yang Chen and Gérard Medioni, "Object modelling by registration of multiple range images," *Image and vision computing*, vol. 10, no. 3, pp. 145–155, 1992.

[6] Kok-Lim Low, "Linear least-squares optimization for point-to-plane icp surface registration," *Chapel Hill, University of North Carolina*, vol. 4, 2004.

[7] Brian Curless and Marc Levoy, "A volumetric method for building complex models from range images," in *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*. ACM, 1996, pp. 303–312.

[8] Siheng Chen, Dong Tian, Chen Feng, Anthony Vetro, and Jelena Kovacevic, "Contour-enhanced resampling of 3d point clouds via graphs," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. Accepted, IEEE, 2017.

[9] Peter Lindstrom and Valerio Pascucci, "Visualization of large terrains made easy," in *Proceedings of the Conference on Visualization'01*. IEEE Computer Society, 2001, pp. 363–371.