B2DC: Balanced Point Cloud Data Reduction based on Biplanar 2D Curvature

Jaeseong Park, Woongsoo Kim, Junhyeong Ryu, and Jeongyeup Paek

Department of Computer Science & Engineering, Chung-Ang University, Seoul, Republic of Korea
{ppoo9811, woongsu0614, rjh6883, jpaek}@cau.ac.kr

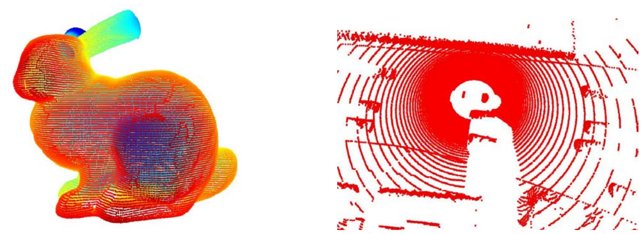
***Abstract*—As the use of 3D point cloud data increases in applications such as autonomous driving, augmented reality, and robotics, there is a growing need for technologies to process this data in real-time while minimizing storage requirements. Since the efficiency of subsequent processing is highly dependent on the size of these datasets, there is growing research on point cloud reduction that effectively reduces points while preserving essential features. However, traditional methods often lack of considering feature loss, while 3D curvature-based and deep learning-based methods, though more accurate, typically involve high computational costs that challenge real-time processing. To address these challenges, we propose *binplanar 2D curvature* (*B2DC*), a novel point cloud reduction algorithm which projects 3D data onto two 2D planes. Unlike prior works, *B2DC* balances feature preservation and computational efficiency by leveraging 2D curvature to retain significant features while effectively reducing complexity. Both qualitative and quantitative results demonstrate that *B2DC* is about 30% faster compared to the 3D curvature-based approach while achieving over 90% feature preservation accuracy.**

***Index Terms*—Point Cloud Reduction, Curvature, Voxel-Grid, 2D Projection, Kdtree**

I. Introduction

3D sensor data is being increasingly utilized across various fields, including autonomous driving, augmented reality, and robotics. One of the most prominent applications is autonomous driving, where LiDAR sensors are commonly used to generate 3D point cloud data. Point cloud data is characterized by its large volume due to the extensive number of points it contains, leading to considerable data size. Numerous studies have explored various tasks using point cloud data, such as compression [1], registration [2], [3], and object detection [4], [5]. However, the performance of these tasks is highly dependent on the size of the dataset, as larger datasets require significantly more processing time and computational resources. As 3D sensor technology advances, point cloud data is becoming more detailed and larger. Consequently, these growing datasets pose significant challenges for real-time processing and storage, especially in resource-constrained environments like embedded systems in autonomous vehicles.

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government (MSIT) (No. 2022R1A4A5034130 & No. RS-2024-00359450), and also by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2024-RS-2022-00156353) supervised by the IITP (Institute for Information & Communications Technology Planning & Evaluation)



(a) Stanford Bunny Data **(b)** OPV2V LiDAR Data

Fig. 1: Representative 3D point cloud data

Given these constraints including memory, bandwidth, and computational capacity, there is a pressing need for efficient point cloud reduction algorithms. Efficient point cloud reduction must not only reduce data size but also preserve essential features to ensure that critical information is retained and subsequent processing tasks remain unaffected. This is crucial for managing large datasets, such as those depicted in Fig. 1 while enabling real-time processing and optimizing the overall performance of point cloud-based applications.

The critical challenges in point cloud data reduction are balancing computational efficiency and preserving essential features to avoid issues in subsequent processing stages. Traditional mesh-based methods [6] are prohibitively impractical due to their excessive computation time. Consequently, point-based methods have become more prevalent in recent years. State-of-the-art methods such as Farthest Point Sampling (FPS) and Random Sampling (RS) often struggle with their capacity to adequately account for point features. While curvature-based methods [7], [8] can effectively preserve features by reflecting the geometric properties of surfaces or curves, they suffer from high computational costs.

In this paper, we propose a biplanar 2D curvature (*B2DC*) to overcome the limitations of previous methods. Our method can effectively reduce point cloud data size by projecting 3D points onto two 2D planes, leveraging 2D curvature. By reducing the dimensionality of the problem, *B2DC* not only lowers computational complexity compared to 3D curvature-based method but also achieves advanced feature preservation to FPS and RS while significantly reducing processing time.

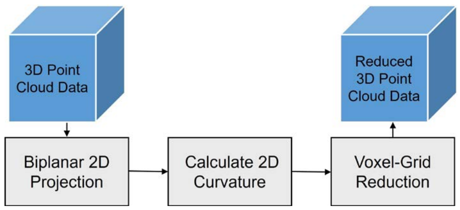


Fig. 2: The flow of *B2DC*

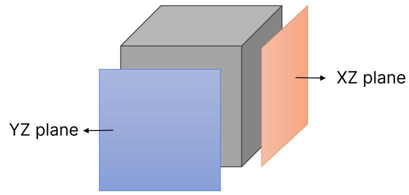


Fig. 3: Biplanar 2D projection

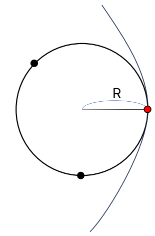


Fig. 4: 2D curvature with three points

II. Related Work

This section discusses the related prior works in the literature on point cloud reduction and point cloud applications.

A. Point Cloud Reduction

Point cloud reduction is a method to reduce the number of points in a point cloud data while retaining essential features and structures such as corners, edges, and high curvature area. There are broadly two categories: mesh-based and point-based methods. Mesh-based methods are outdated and are not commonly used because of intensive computational requirements. As a result, recent research predominantly focuses on point-based methods, which can be further divided into geometric methods and deep learning-based methods. A state-of-the-art method is FPS which iteratively samples the farthest point and updates distances. It can evenly distribute points across the entire space of a point cloud, but it does not sufficiently take into account the features of the data. Additionally, since it requires continuous distance calculations, it can be computationally expensive and time-consuming. RS is also a widely used method. This is easy to implement and computationally efficient because it simply selects points randomly according to a desired ratio. However, RS does not account for the features of the data at all.

To address these issues, several approaches have been proposed. Voxel-grid-based method [9] involves dividing the data into a voxel grid and selecting a representative point for each voxel using algorithms such as a median filter. Clustering-based method [10] applies k-means clustering and then uses existing techniques such as FPS for each cluster. Curvature-based method [8] performs reduction by calculating Gaussian curvatures and mean curvatures from the principal curvature in 3D space, using these curvature measures as the basis for the reduction process. While these methods provide the best feature preservation among the previously mentioned methods, they are limited by their high computational cost. While the above methods strive to preserve features, they either involve high computational costs or still fall short of adequately preserving features.

Recently, approaches utilizing deep learning approaches have been actively proposed [11], [12]. These methods demonstrate high accuracy with less feature loss. However, they are often considered unsuitable for preprocessing due to their computational complexity and other limitations.

B. Point Cloud Application

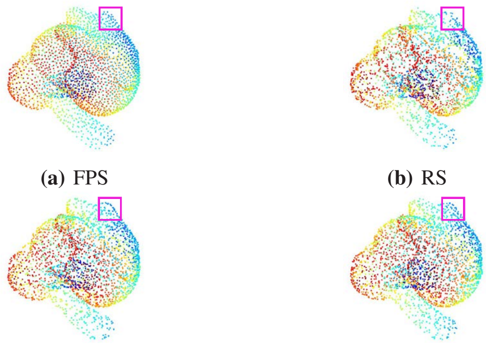
Recently, various applications have been developed utilizing point cloud data. These applications may also apply point cloud reduction as a preprocessing step.

Point cloud compression is a method for encoding point cloud data more efficiently for communication between devices. The MPEG group has even proposed a standard for point cloud data known as G-VCC [1]. Point cloud registration and fusion are techniques used to align and merge different point cloud datasets into a unified representation. F-Cooper [2] performs fusion by first dividing the point cloud into voxels, and then extracting features from each voxel to guide the fusion process. Object detection is a process for identifying and classifying objects in 3D space using point cloud data. VoxelNet [5] revolutionizes point cloud processing by converting raw point clouds into a voxel grid representation. It adopts a point cloud reduction method known as RS. PointPillars [4] introduces a novel pillar-based representation, where the 3D point cloud is divided into vertical columns or pillars each representing a segment of the scene. PointNet++ [13] is a deep learning network that enhances 3D point cloud processing through a hierarchical approach, capturing both local and global features for improved understanding and classification of complex structures. It employs a point cloud reduction method known as FPS.

III. Biplanar 2D Curvature (*B2DC*)

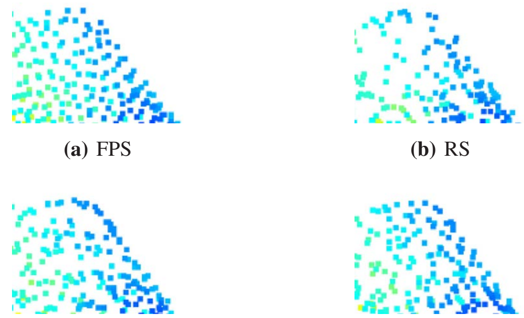
In this section, we introduce the biplanar 2D curvature-based point cloud reduction to achieve an improved balance between feature preservation and computational efficiency.

Existing methods are often hindered by high computational time and resource consumption. FPS and RS frequently overlook significant features, while 3D curvature-based methods, although effective for feature preservation, are limited by their high computational costs. To handle this, we propose leveraging 2D curvature to reduce computational complexity. Calculating curvature in 3D requires extensive computation, whereas performing these calculations in 2D significantly reduces the computational load. Our goal is to achieve efficient reduction while maintaining a balanced preservation of detailed information. Instead of representing the data through sampling of existing points, which does not inherently adjust the ratio of data points, our approach adjusts the voxel grid size to control the sampling ratio without altering the spatial positions of the points.



(a) FPS **(b)** RS **(c)** 3D Curvature **(d)** *B2DC*

Fig. 5: Reduction results for Stanford Bunny dataset



(a) FPS **(b)** RS **(c)** 3D Curvature **(d)** *B2DC*

Fig. 6: Enlarged foot of the Stanford Bunny dataset

B2DC involves a sequence of the following steps; biplanar 2D projection, calculate 2D curvatures, calculate mean curvatures, and voxel-grid point cloud reduction (Fig. 2).

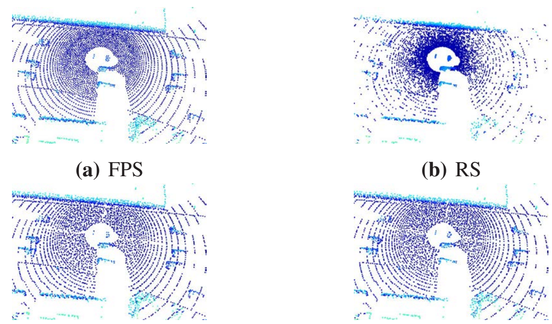
First, we perform a biplanar 2D projection by mapping each point onto the XZ and YZ planes (Fig. 3). In other words, it simply means that only two out of the three XYZ coordinates are used at a time. Using only a single plane would be insufficient for capturing the global curvature information necessary to represent comprehensive features. Conversely, utilizing all three planes would lead to unnecessary complexity and computational overhead. By employing two orthogonal planes, our method strikes a balance between computational efficiency and feature preservation. This biplanar strategy mitigates the computational burden associated with a three-plane projection while ensuring sufficient feature information is retained for precise data representation.

Next, we calculate 2D curvatures for each plane separately. To determine the curvature in 2D, we use the three-point method (Fig. 4). This involves applying a kdtree-based k-nearest neighbors (K-NN) algorithm to find the two nearest neighbors for each point. We then compute the curvature as the reciprocal of the radius of the circle that passes through these three points. Projecting onto 2D planes can simplify the K-NN process, as the kdtree-based K-NN algorithm operates faster in 2D due to the reduced dimensionality. Additionally, the 2D projection eliminates the need for the complex differentiation and matrix operations required for 3D curvature calculations, simplifying the process to computing only the radius of the circle defined by the three points. Following this, we compute the mean curvature by averaging the curvatures obtained from the two planes.

Finally, we apply voxel-grid point cloud reduction by dividing the point cloud into a uniform voxel grid. Within each voxel, we retain the point with the highest mean curvature and discard the others.

IV. Evaluation

In this section, we evaluated the accuracy and processing time of *B2DC* with other three widley adopted point cloud reduction methods. For the comparison, accuracy was assessed based on how closely the preservation of feature points matched that of the 3D curvature-based method, which is used as a baseline due to its established performance in this area. We used an AMD Ryzen 5 7500F with 16GB of RAM, along with the Stanford Bunny and OPV2V [14] datasets. Stanford Bunny is a single 3D model dataset, while OPV2V is vehicle data generated through CARLA simulation (Fig. 1). Both datasets are widely used in the field of 3D point cloud processing. Stanford Bunny data have relatively consistent and higher density, allowing for the use of a smaller voxel size, while OPV2V data tend to be sparse and have highly variable point density, requiring a larger voxel size during processing. In terms of point reduction, the Stanford Bunny dataset is reduced from 35,947 points to 2,825 points, achieving a reduction rate of 92.1%. Similarly, the OPV2V dataset was reduced from 57,069 to 7,950 points, reflecting an 86.1% reduction rate.



(a) FPS **(b)** RS **(c)** 3D Curvature **(d)** *B2DC*

Fig. 7: Reduction results for OPV2V LiDAR dataset

For qualitative evaluation, we compared the result images (Figs. 5 to 7). Fig. 6 shows a enlarged version of the foot of the Stanford Bunny (Fig. 5), while Fig. 7 displays a magnified section of the OPV2V LiDAR dataset focusing on the areas with a high visibility of vehicles. FPS tends to maintain a uniform point density effectively, but it does not consider important features such as flat area and high curvature area compared to curvature-based method. RS, due to its random

	Stanford Bunny	OPV2V LiDAR
FPS	1.28 s	8.86 s
RS	0.001 s	0.002 s
B2DC	0.13 s	0.19 s
3D Curvature	0.18 s	0.27 s

TABLE I: Processing time comparison for Stanford Bunny and OPV2V LiDAR datasets

	Stanford Bunny		OPV2V LiDAR	
	w/ tolerance	w/o tolerance	w/ tolerance	w/o tolerance
FPS	84.34%	8.57%	86.95%	45.79%
RS	83.55%	7.47%	82.16%	14.23%
B2DC	90.13%	16.75%	91.45%	53.25%

TABLE II: Accuracy comparison for Stanford Bunny and OPV2V LiDAR datasets relative to 3D curvature-based method as the 100% baseline.

selection of points, does not account for features at all, a limitation that is particularly evident in Fig. 7b. The 3D curvature-based method preserves high curvature area well, and also there is no substantial difference in visual compared to our proposed *B2DC* method.

For quantitative evaluation, we measured the time required for point cloud reduction and assessed accuracy against the 3D curvature-based method. In Table I, we can observe that FPS requires the longest time for processing because it calculates distances at each step. On the other hand, RS is the fastest method by far as it simply selects points randomly. *B2DC* shows a significant improvement in processing time over 3D methods, reducing processing time by 27.78% on the Stanford Bunny dataset and 29.73% on the OPV2V dataset. Although there is no significant difference in accuracy compared to other methods, *B2DC* achieves a meaningful reduction in processing time. Table II presents the proportion of points that are either within a certain tolerance of each other or are exactly identical when sampled, based on the 3D curvature-based method. The tolerance values are selected based on the density of the dataset and the average inter-point distance. The results indicate that, for both datasets, accuracy is highest for *B2DC*, followed by FPS and RS in that order.

Summarizing the experimental results, our proposed *B2DC* method demonstrates feature preservation accuracy comparable to that of the 3D curvature-based method while achieving approximately 30% faster processing speeds compared to the 3D curvature-based method.

V. Conclusion

We proposed a novel method, *B2DC* for point cloud data reduction using biplanar 2D curvature, effectively addressing the challenge of balancing computational efficiency with the preservation of essential features. By projecting 3D points onto two 2D planes and leveraging curvature in these reduced dimensions, *B2DC* significantly reduces computational complexity compared to traditional 3D curvature-based approaches, with minimal accuracy loss. Experimental results demonstrated that our approach achieves superior performance in reducing point cloud size, unlike RS, while maintaining the integrity of critical features, outperforming existing methods such as FPS and the 3D curvature-based method in terms of computational efficiency. This work contributes to the ongoing efforts to optimize point cloud processing, especially in applications requiring real-time performance, and opens up new avenues for further research in feature-preserving data reduction techniques. Moreover, applying our method has the potential to develop new optimized point cloud registration and object detection algorithms for applications requiring a balance between computational load and feature preservation.

References

[1] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuća, S. Lasserre, Z. Li *et al.*, “Emerging MPEG standards for point cloud compression,” *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 9, no. 1, pp. 133–148, 2018.

[2] Q. Chen, X. Ma, S. Tang, J. Guo, Q. Yang, and S. Fu, “F-cooper: Feature based cooperative perception for autonomous vehicle edge computing system using 3D point clouds,” in *Proceedings of the 4th ACM/IEEE Symposium on Edge Computing*, 2019, pp. 88–100.

[3] J. Ryu and J. Paek, “Poster: Fast Field-of-View Expansion for Collaborative Object Detection,” in *Proceedings of the 22nd Annual International Conference on Mobile Systems, Applications and Services*, 2024, pp. 684–685.

[4] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, “PointPillars: Fast Encoders for Object Detection from Point Clouds,” in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2019, pp. 12 697–12 705.

[5] Y. Zhou and O. Tuzel, “VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection,” in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 4490–4499.

[6] P. Cignoni, C. Montani, and R. Scopigno, “A comparison of mesh simplification algorithms,” *Computers & Graphics*, vol. 22, no. 1, pp. 37–54, 1998.

[7] Y. Zhang, F. Meng, and N. Nan, “Research on Adaptive Grid Method for Point Clouds Reduction,” in *9th Annual International Conference on Network and Information Systems for Computers (ICNISC)*. IEEE, 2023, pp. 364–367.

[8] G. Wang, L. Wu, Y. Hu, and M. Song, “Point cloud simplification algorithm based on the feature of adaptive curvature entropy,” *Measurement Science and Technology*, vol. 32, no. 6, p. 065004, 2021.

[9] W. Lyu, W. Ke, H. Sheng, X. Ma, and H. Zhang, “Dynamic Downsampling Algorithm for 3D Point Cloud Map Based on Voxel Filtering,” *Applied Sciences*, vol. 14, no. 8, p. 3160, 2024.

[10] B.-Q. Shi, J. Liang, and Q. Liu, “Adaptive simplification of point cloud using k-means clustering,” *Computer-Aided Design*, vol. 43, no. 8, pp. 910–922, 2011.

[11] Y. Lin, Y. Huang, S. Zhou, M. Jiang, T. Wang, and Y. Lei, “DA-Net: density-adaptive downsampling network for point cloud classification via end-to-end learning,” in *4th International Conference on Pattern Recognition and Artificial Intelligence (PRAI)*. IEEE, 2021, pp. 13–18.

[12] C. Wu, J. Zheng, J. Pfrommer, and J. Beyerer, “Attention-based point cloud edge sampling,” in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2023, pp. 5333–5343.

[13] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, “Pointnet++: Deep hierarchical feature learning on point sets in a metric space,” *Advances in neural information processing systems*, vol. 30, 2017.

[14] R. Xu, H. Xiang, X. Xia, X. Han, J. Li, and J. Ma, “Opv2v: An open benchmark dataset and fusion pipeline for perception with vehicle-to-vehicle communication,” in *International Conference on Robotics and Automation (ICRA)*. IEEE, 2022, pp. 2583–2589.