R-PCC: A Baseline for Range Image-based Point Cloud Compression

Sukai Wang, Jianhao Jiao, Peide Cai, and Lujia Wang

***Abstract*— In autonomous vehicles or robots, point clouds from LiDAR can provide accurate depth information of objects compared with 2D images, but they also suffer a large volume of data, which is inconvenient for data storage or transmission. In this paper, we propose a Range image-based Point Cloud Compression method, R-PCC, which can reconstruct the point cloud with uniform or non-uniform accuracy loss. We segment the original large-scale point cloud into small and compact regions for spatial redundancy and salient region classification. Our range image-based method can keep and align all points from the original point cloud in the reconstructed point cloud, and the setting of the quantization module restricts the maximum reconstruction error. In the experiments, we prove that our easier FPS-based segmentation method can achieve better performance than instance-based segmentation methods such as DBSCAN, and our non-uniform compression framework shows a great improvement on the downstream tasks compared with the state-of-the-art large-scale point cloud compression methods. Our real-time method can achieve 40× compression ratio without affecting downstream tasks, to act as a baseline for range image-based point cloud compression. The code is available on https://github.com/StevenWang30/R-PCC.git.**

I. Introduction

Point clouds from scanning LiDAR can not only provide high-accuracy depth information of objects in a large range, but are also suitable for diverse environments like various lighting conditions. However, the large-volume data stream obtained by the LiDAR will lead to problems in practical use, such as storage or transmission. Take Velodyne-64E for example. It will collect 30GB+ of point cloud data per hour. Therefore, large-scale point cloud compression has become necessary for autonomous driving systems.

Point clouds used in 3D scanning or modeling are very compact and dense. In recent decades, many 3D point cloud compression methods have proposed to compress this type of point cloud, by voxelization [1], [2] or 3D mesh compression [3]. The point cloud compression ratio will increase when the point cloud becomes denser. This is because from the feature aspect, denser points provide more accurate normal information and context features. And from the entropy aspect, the entropy of the point cloud is smaller when the probability of the position of each point is higher. However, compared with the 3D models, point clouds from scanning LiDAR in outdoor scenarios are commonly scattered through regions up to 100 meters away. Thus the compact models can not perform well in the large-scale point cloud compression in the autonomous driving systems.

Tree-based [4]–[6] and range image-based algorithms [7]–[9] are two kinds of effective compression types. Among tree-based methods, in Draco [4], a mesh and KD-tree are fed into the algorithm for compression, while in [10], Huang *et al*. firstly reorganized the point cloud into an octree, and used a machine learning model to predict the probability of each voxel, which can be fed into the arithmetic coding algorithm for end-to-end compression. Meanwhile, range images are the 2D depth images of 3D point clouds from the frontal view, and an unsynchronized and unrectified point cloud will be directly backprojected by the range image stream collected by LiDAR. Image compression methods, such as JPEG [11] and JPEG2000 [12], can be applied to range image compression, such as JPEG or JPEG2000. Compared with the octree-based or KD-tree-based methods, range image-based compression framework can guarantee the number of points in the reconstructed point cloud is exactly the same as that in the original point cloud. Because range image has geometry clusters and larger range of values compared with ordinary image from camera, Sun *et al*. [8] proposed an instance-based segmentation method that uses Euclidean clustering to reduce the spatial redundancy, and achieved 20× compression ratio. However, the instance-based or semantic-based segmentation has high time and space complexity for real-time implementation. In this paper, we propose a region-based method using farthest point sampling (FPS). In Sec. IV-C, we compare the compression ratio and reconstruction quality of instance-based and region-based segmentation methods, and the results illustrate that semantic and accurate segmentation cannot improve the overall compression performance, while our uniform compression framework can achieve 30× compression ratio with 2cm chamfer distance error.

Another reason to segment a large-scale point cloud into small regions is that we can decrease the compressed bitstream size without affecting downstream tasks by retaining high compression accuracy in important regions and reducing the compression accuracy in unimportant areas. Our non-uniform compression framework uses a key point extractor to classify the clusters into four salience levels. The experimental results for 3D object detection and simultaneous localization and mapping (SLAM) on the KITTI dataset show that our method can achieve a higher compression ratio with less decrease in downstream task performance than other baseline methods.

This work was supported in part by the Foshan-HKUST Project FSUST20-SHCIRI06C, Zhongshan Municipal Science and Technology Bureau Fund Project ZSST21EG06, and Guangdong Province Basic and Applied Basic Research Fund Project Project 2021B1515120032. (*Corresponding author: Lujia Wang.*)

Sukai Wang is with the Department of Computer Science and Engineering, Jianhao Jiao, Peide Cai, and Lujia Wang are with the Department of Electronic and Computer Engineering, Hong Kong University of Science and Technology, Hong Kong SAR 999077, China.(e-mail: {swangcy, jjiao, peide.cai}@connect.ust.hk; eewanglj@ust.hk).

The contributions of this work are listed as follows:

- We propose the more efficient and effective FPS segmentation and point-plane mixing modeling method, which can help to achieve better compression ratio by reordering the residual ranges and distributions.
- We propose a uniform and non-uniform compression framework for different requirements. Salient regions with more key points can keep high reconstruction quality for downstream tasks.
- We compare our compression framework with other state-of-the-art algorithms, and achieve superior performance in both reconstruction quality and downstream task performance. Our real-time framework can become a novel range image-based point cloud compression baseline as the front-end in autonomous driving system.

II. Related Works

A. Single-frame Point Cloud Compression

For static unstructured point clouds, octree representation is commonly utilized as a point-cloud geometry compression method. Elseberg *et al*. [13] proposed an efficient octree data structure to store and compress 3D data without loss of precision. Experimental results demonstrated that their method was useful for the exchange file format, fast point cloud visualization, fast 3D scan matching, and shape detection. De Oliveira Renteet *et al*. [14] introduced a graph-based lossy coding algorithm for the geometry of static point clouds. They used an octree-based technique for the base layer and a graph-based transform technique for the enhancement layer, where a residual was coded, leading to impressive coding performance. Zhang *et al*. [15] introduced a clustering and DCT-based color point cloud compression method, in which they used a mean-shift technique to cluster 3D color point clouds into many homogeneous blocks, and a clustering-based prediction method to remove spatial redundancy of the point cloud data. Meanwhile, Tang *et al*. [16] presented an octree-based scattered point cloud compression algorithm, in which the stop condition of segmentation was improved to ensure appropriate voxel size. Additionally, the spatial redundancy and outliers could be removed by traversal queries and bit manipulation.

For static structured point clouds, researchers have focused on employing existing image coders to encode point cloud data by mapping them into 2D arrangements. Houshiar *et al*. [17] proposed to project 3D points onto three panorama images and used an image coding method to compress them. Similar to their approach, Ahn *et al*. [18] presented an adaptive range image compression algorithm for the geometry information of large-scale 3D point clouds. They explored a prediction method to predict the radial distance of each pixel using previously encoded neighbors and only encoded the resulting prediction residuals. In contrast, Zanuttigh *et al*. [19] focused on efficient compression of RGB-D point cloud data. They developed a segmentation method to identify the edges and main objects of a depth map. After that, an efficient prediction process was performed according to the segmentation result, and the residual data between the predicted and real depth map were calculated. Finally, the few prediction residuals were encoded by conventional image compression methods. Sun *et al*. [7]–[9] also proposed a cluster-based method for range images to reduce the residual data range and the entropy of the residual data. However, the ablation and comparative results in this paper shows that the accurate segment methods like DBSCAN for semantic or instance-based segmentation did not show improvement in the overall compression ratio. In our method, we utilize the segmentation results for salience map creation, and we apply a non-uniform framework to achieve a higher compression ratio without a negative effect on the downstream tasks.

B. Key Point Extraction

For most downstream tasks in autonomous driving systems, the feature extraction of the point cloud is crucial. Feature-based methods focus on the sharp features for edges or flat features for surface matching. LOAM [20] is the standard framework of many current SLAM-related methods like LeGO-LOAM [21] or M-LOAM [22]. LOAM splits each scan in the range image into several segments, and finds the top K sharpest and flattest points in the segments as the key points. The more violent the depth change in the adjacent pixels in the same scan, the sharper the edge features. Serafin *et al*. [23] proposed a fast and robust 3D feature extraction method in sparse point clouds, which can extract the planes and edges. In our key point extraction module, we utilize the feature extractor in LOAM [20] and evaluate A-LOAM's [24] performance (an open-source implementation of LOAM) the SLAM downstream tasks in the comparative experiments.

III. Methodology

A. System Overview

Our proposed uniform or non-uniform point cloud compression framework R-PCC is shown in Fig. 1. The decompression part of our framework uses the same basic compressor as in the compression framework, to decompress the segmentation and modeling information data (*info. data*) and quantized residual data. The *info. data* can predict the coarse point clouds as in the compression framework, and the residual is recovered by the inverse quantization module. In the non-uniform framework, the accuracy for each cluster corresponds to the quantization module in compression.

Our proposed compression framework is based on the range image. Note that if the range image is collected from the LiDAR, then the number of points is also the same as in the point cloud; and if the range image is projected by the 3D point cloud, the number of points in the reconstructed point cloud depends on the shape of the range image.

The accuracy loss thus consists of two parts:

1) The projection from the point cloud to the range image.
2) The uniform or non-uniform quantization accuracy.

B. Range Image

Currently, single-frame point clouds from most LiDARs can be projected from 3D to 2D. A LiDAR has different

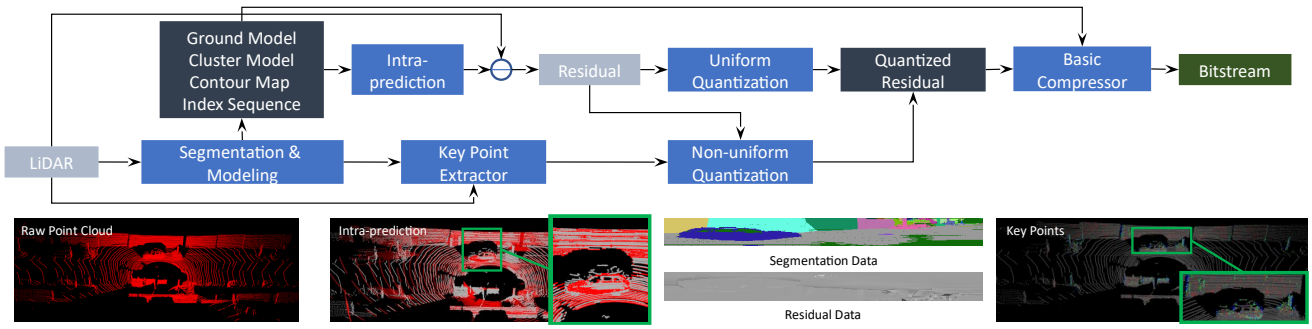

Fig. 1: Block diagram of our uniform and non-uniform compression framework, R-PCC. In the non-uniform quantization module, the residual is quantized by different quantization accuracies in each cluster based on the key point distribution. The green block is the bitstream for storage or transmission, two black blocks are the information data and the quantized residual data, which will be encoded and decoded losslessly by the basic compressor.

Fig. 2: Example of the point modeling and plane modeling method. The blue points are intra-predicted result using point model or plane model. The residual of point A is OA - OA'.

laser beams (e.g. Velodyne HDL-64E has 64 lasers, and 32E has 32 lasers), and all the lasers have a full rotation of 360° along the azimuth direction (horizontal field of view).

Here we take the Velodyne HDL-64E as an example. In the vertical field of view, the range image consists of 64 rows whose angles are distributed between the lowest angle φ_{min} and highest angle φ_{max}. Each scan in the range image represents a fixed angle. If the LiDAR has H lasers, and the horizontal angular resolution is ρ, the shape of the range image collected by the LiDAR should be $[H,W] = [H,\lfloor 360/\rho \rceil]$, where $\lfloor\rceil$ represents the rounding operation.

We can project a 3D point $\mathbf{P} = (x,y,z)$ onto the corresponding 2D pixel $\mathbf{I} = (w,h,r)$ of a range image, where w and h are the vertical index and horizontal index, and r is the Euclidean distance from the point to the LiDAR origin. Values of (w,h,r) are calculated accordingly. $r = \sqrt{x^2+y^2+z^2}$, $h = \lfloor \theta/\rho \rceil$, and $w = \lfloor (\varphi - \varphi_{min})/\sigma \rceil$, where $\theta = \arctan(y/x)$ and $\varphi = \arctan(z/r)$ are the horizontal angle and vertical angle respectively, φ_{min} is the smallest vertical angle, and $\sigma = W/(\varphi_{max} - \varphi_{min})$.

C. Compression Framework

Ground Extraction Module. The ground points have strong regularity because the ground points can be fitted into a large plane. We estimate the ground model using the RANSAC planer fitting method like [8].

Segmentation Module. This module segments the point cloud into several denser point cloud subsets. Compared with the instance-based segmentation methods in [7] and [8], we choose FPS method to find the center of each cluster as region-based segmentation method. The number of clusters is equal to the number of sampled points setting in FPS. In the Sec. IV-C, we compared the DBSCAN as baseline with ours segmentation method, and the results shows that ours method performs better in compression and efficiency.

Modeling Module. After obtaining the small point cloud clusters, we use two methods, point and plane, to model the points in each cluster. Different modeling methods vary the different residual distribution, and more regular distribution can obtain higher compression ratio. When a cluster is compact, we tend to choose point modeling, and when a cluster has plane surface, the plane modeling is better. The point modeling method uses the mean of the points' depth, and the plane modeling method uses a plane, which is estimated by RANSAC, to represent points in each cluster. When the number of points in the cluster is smaller than 30, or the maximum angle between the plane norm and LiDAR scans in this cluster is larger than 75°, we will choose the point modeling method.

For a point set $\{\mathbf{P}_i = (x_i,y_i,z_i)\}_{i=1}^{k}$ in the cluster, the point model: $r = \frac{1}{k}\sum||\mathbf{P}_i||_2$, and the plane model $ax+by+cy+d = 0$, where $\{r\}$ and $\{a,b,c,d\}$ are the model parameters.

Intra-prediction Module. An example of the intra-prediction result of the plane modeling and the point modeling method for a car on the KITTI dataset is shown in Fig. 2. We can predict the whole point cloud with the segmentation *info. data* and the model parameters. For a point in the predicted range image $\hat{\mathbf{I}} = (w,h,\hat{r})$ and its model parameters, we want to predict $\hat{r}$. Its 3D location $\hat{\mathbf{P}} = (\cos\varphi\cdot\cos\theta\cdot\hat{r}, \cos\varphi\cdot\sin\theta\cdot\hat{r}, \sin\varphi\cdot\hat{r})$, where the calculation of φ and θ is in Sec. III-B. When the point is modeled by a point model $\{r\}$, then $\hat{r} = r$; if the point is modeled by a plane model $\{a,b,c,d\}$, and from the plane model equation, we can calculate the predicted $\hat{r}$:

$$\hat{r} = -d/(a\cdot\cos\varphi\cdot\cos\theta + b\cdot\cos\varphi\cdot\sin\theta + c\cdot\sin\varphi). \quad (1)$$

Key Point Extractor and Salience Map. This module is proposed for the non-uniform quantization. Most downstream tasks only need the salient foreground region of the point cloud to be accurate enough, not the whole point cloud. Thus, we propose a non-uniform compression and decompression framework to improve the compression efficiency to maintain the high performance of the downstream tasks.

We select feature points that are on sharp edges and planar surface patches from LiDARs' raw measurements. We follow

[20] to evaluate the curvature of a point in a local region to classify it as an edge (high curvature) and planar point (low curvature) according to

$$c(\mathbf{P}_i) = \frac{||\sum_{\mathbf{P_j} \in \mathscr{S}, \mathbf{P}_i \neq \mathbf{P}_j} (\mathbf{P}_i - \mathbf{P}_j)||}{|\mathscr{S}| \cdot ||\mathbf{P}_i||}, \quad (2)$$

where $\mathscr{S}$ is the set of consecutive points of $\mathbf{P}_i$ in the same scan. And then we set a series of salience score thresholds, to classify each cluster into different salience levels. For the clusters which have fewer key points, we will reduce the quantization accuracy of the residual data.

Residual Quantization Module. The residual data are calculated by the range image minus the intra-predicted range image. For example in Fig. 2, the residual of point A: $res_A = r_A - \hat{r}_A = |OA| - |OA'|$. For the uniform compression framework, the residual data will be quantized into integers using uniform accuracy, and for the non-uniform framework, the residual data in the unimportant clusters (such as leaves on trees, or meadows) will be quantized into integers with lower quantization accuracy. The maximum reconstruction error of each point is half the quantization accuracy.

D. Decompression

In the decompression module, the segmentation *info. data* and the quantized residual data will be decoded from the same basic compressoras in the compression. And the intra-prediction results can be obtained by the intra-prediction module. Then the dequantization module recovers the integer residual data into float numbers as the inverse process of the quantization module. Then we can reconstruct the whole range image by adding the residual data into the intra-predicted range image, which can be transformed into a reconstructed point cloud.

IV. Experiments

A. Evaluation Metrics

In the compression stage, the original point cloud can be encoded into a bit stream for storage or transmission. The bit-per-point (BPP) and compression ratio (CR) are two naive metrics to evaluate the size of the compressed bitstream. In this paper, we only consider the geometry compression of the point cloud, so, the BPP of the original point cloud is 96 for the three float values x, y, and z.

The reconstruction quality is evaluated by three evaluation metrics: F_1 score, point-to-point chamfer distance (CD) [25], [26], and point-to-plane PSNR (D2 PSNR) [27], [28]. For the original point cloud P and reconstructed point cloud $\hat{P}$:

$$F_1 = 2TP/(2TP + FP + FN), \quad (3)$$

$$CD = CD_{sym}(P, \hat{P}) = \left[CD(P, \hat{P}) + CD(\hat{P}, P)\right]/2, \quad (4)$$

$$PSNR = 10\log_{10}\left[3r^2/\max\{MSE(P, \hat{P}), MSE(\hat{P}, P)\}\right], \quad (5)$$

where the distance threshold in F_1 score $\tau_{geo} = 2$cm and the peak constant value of the point-to-plane PSNR $r = 59.70$m, and the other detailed definitions are the same as in [6]. For downstream tasks, the bounding box average precision (AP) [29], [30] of the classes car, pedestrian, and cyclist are evaluated for 3D object detection, and in SLAM, the absolute trajectory error (ATE) and the relative pose error (RPE) [31] for translation and rotation are evaluated.

B. Dataset

To evaluate the compression ability and quality in the point clouds with different densities, we choose three datasets: HKUSTCampus, Oxford [32], and KITTI [33], which were collected by a Velodyne VLP-16, Velodyne HDL-32E, and Velodyne HDL-64E, respectively. The HKUSTCampus dataset was collected with a handheld Velodyne at HKUST. Because the point clouds are achieved directly from the range images in HKUSTCampus, the points in the point clouds and pixels in the range images can be completely assigned; but the point clouds in Oxford and KITTI were pre-processed before publication. In the comparative experiments on the downstream tasks, the results of the original point cloud from the dataset (Raw Data), and point cloud backprojected from a range image (Original Data) are compared both.

The point cloud distribution in different scenes can be very different. We choose city, residential, campus, and road in the KITTI raw data, like [7], to evaluate the compression ratio in various scenes. The point clouds in the first three scenes are denser and better-regulated than in the road scene.

C. Ablation Study

Segmentation and Modeling. The segmentation methods control the cluster size and residual range, and the modeling methods vary the surf residual distribution. We implement *open3D* DBSCAN as the instance-based segmentation baseline as in [7]. The comparative results of *FPS-* and *DBSCAN-* in Fig. 3 shows that our FPS-based segmentation method is better than instance-based segmentation. Object-based segmentation is not necessary in the compression tasks when using BZip2 as basic compressor. Fig. 3 also shows that the residual and residual BPP decrease when the cluster is smaller and denser. The overall BPP is best when the number in the cluster is from 50 to 100. Tab. I shows that the plane modeling method in our framwork can largely decrease the mean of the residual data, especially in the KITTI city dataset, because there are more walls and planes in the city scene. In the KITTI road dataset, the point modeling method is better than the plane modeling method.

Basic Compressor. In this section, we show the compression ratios of our compression framework with different basic compressors, and compare it with the baseline compression. In the baseline compression, the single basic compressor is used to encode the quantized point cloud with the same quantization accuracy as the residual quantization in our method. In our method with different basic compressor, the number of clusters is set to 100 and the segmentation and modeling method is *FPS+plane*. Fig. 4 shows that the denser of the point cloud, the higher of the compression ratio. From Fig. 4 and Tab. II we can find that the compression performance of BZip2 is the best in our experiments, but the speed is also the slowest.

TABLE I: COMPARATIVE RESULTS OF INFORMATION BPP, RESIDUAL BPP, AND MEAN OF RESIDUAL IN FOUR DIFFERENT SETTINGS ON FOUR KITTI DATASETS IN SCENE CITY, RESIDENTIAL, CAMPUS, AND ROAD.

	FPS+point			FPS+plane			DBSCAN+point			DBSCAN+plane		
	IBPP	RBPP	Res	IBPP	RBPP	Res	IBPP	RBPP	Res	IBPP	RBPP	Res
KITTI_city	0.6	1.8	0.7	0.65	1.74	0.56	0.56	1.78	1.68	0.59	1.76	1.5
KITTI_residential	0.51	2.57	0.57	0.55	2.55	0.52	0.43	2.64	2.35	0.44	2.64	2.22
KITTI_campus	0.93	2.53	0.93	0.99	2.53	0.9	1.01	2.54	1.69	1.08	2.52	1.67
KITTI_road	1.08	4.05	0.8	1.11	4.09	0.81	1.03	4.24	2.39	1.03	4.26	2.38

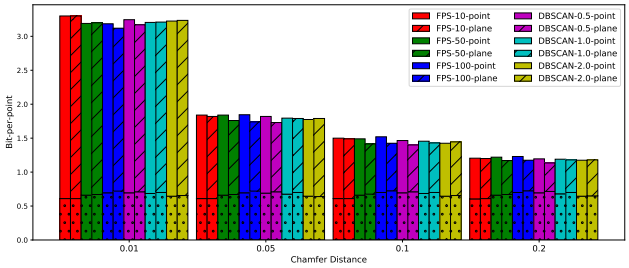

Fig. 3: The chart diagram of BPP with different methods. For example, *FPS-10-point* means the segmentation method is FPS, cluster number is 10, and model method is *point* only; *DBSCAN-2.0-plane* means DBSCAN with neighbor distance (eps) 2.0m and the model method is *plane/point*.

TABLE II: THE ENCODING AND DECODING TIME COST OF DIFFERENT COMPRESSORS.

	Encoding time (ms)				Decoding time (ms)			
	LZ4	BZip2	Deflate	AC	LZ4	BZip2	Deflate	AC
16E	0.03	11.4	3.7	40	0.03	5.2	1.3	19.1
32E	0.04	26.5	9.7	92	0.03	11.7	3.72	43.4
64E	0.1	47.9	17.5	161.7	0.1	20.9	6.8	75.5

D. Comparative Results

In this section, we compare our proposed uniform and non-uniform compression frameworks with the baseline point cloud compression methods: the KDTree-based algorithm from Google: Draco [4]; geometry-based compression method: G-PCC [34], [35]; and range image-based compression using image compression method JPEG2000 (JPEG Range) [11], [12]. Our compression frameworks use *FPS+plane* as segmentation and modeling method, the number of clusters is 100, the basic compressor is BZip2, and the dataset is KITTI city. More specifically, our non-uniform compression method classifies the clusters into four saliency levels, and the rules for the classification and non-uniform quantization accuracy are shown in Tab. III. For example, if the basic accuracy is 0.02m, and one cluster has 7 key points, then the saliency level of this cluster is 2 and the quantization accuracy of the residual data in this cluster is $0.02+0.04=0.06$m. We compare the quality of the reconstructed point cloud and the performance of different downstream tasks in this section.

Reconstruction Quality. Fig. 5 shows the quantitative

TABLE III: CLUSTERS' SALIENCY LEVEL WITH NUMBER OF KEY POINTS AND QUANTIZATION ACCURACY.

	level 3	level 2	level 1	level 0
Key Point Num	[0, 3)	[3, 10)	[10, 30)	[30, inf)
Quantization Acc	+0.06	+0.04	+0.02	Base Acc

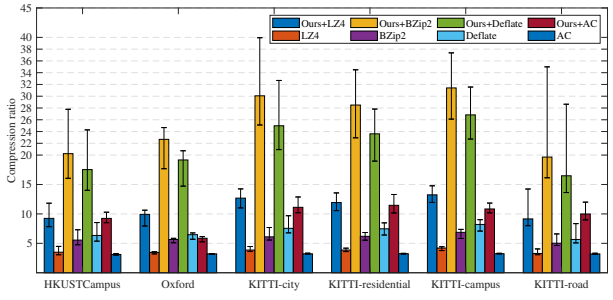

Fig. 4: The chart diagram of the compression ratio of our method with different basic compressors. Three datasets, HKUSTCampus, Oxford [32], and different scenes in KITTI [33] are tested. Single basic compressors are directly applied to compress the quantized point cloud as the baseline method.

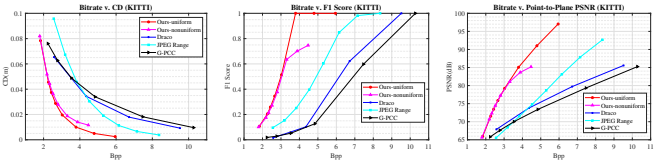

Fig. 5: Quantitative results on KITTI city dataset. Bit-per-point *vs*. symmetric chamfer distance (↓), F1 score (with $\tau_{geo}=0.02$m) (↑), and point-to-plane PSNR (with $r=59.70$) (↑) are shown from left to right.

results of our method with baseline methods on the KITTI city dataset. It shows that the compression effectiveness and reconstruction quality of our framework is much better than that of the other three baseline methods, because we can control the maximum error of the reconstructed point cloud and the high-level basic compressor can compress the range image better. The reconstruction quality of non-uniform compression is a little less than the uniform compression because the unimportant regions have large error.

Downstream Tasks. The first row of Fig. 6 shows the evaluation results of the 3D object detection with a reconstructed point cloud and raw data. Compared with other methods, our proposed compression framework can reach "lossless" compression and decompression at a lower compressed size (BPP). And though the non-uniform framework is not as good as the uniform framework in terms of reconstruction quality, its performance in downstream tasks is better than that of the uniform framework, because the important objects and features are kept after compression and decompression. The performance improvement of the non-uniform framework is more obvious in small object detection

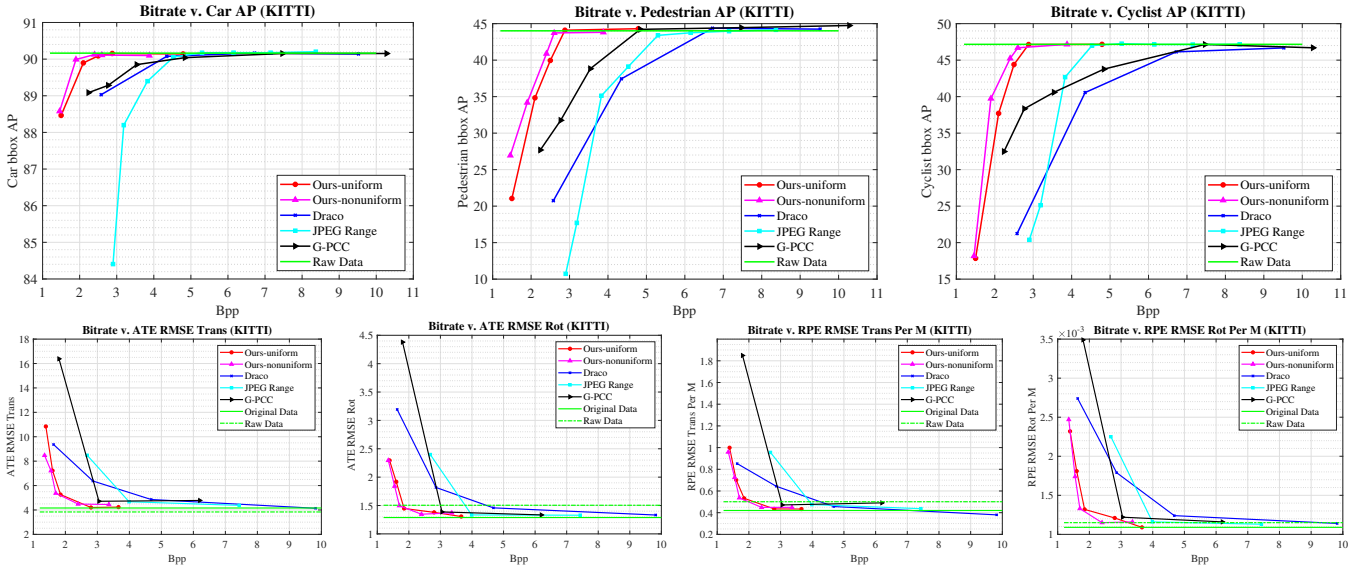

Fig. 6: Quantitative results of the 3D object detection (PointPillar) on the KITTI detection dataset and SLAM (A-LOAM) on the KITTI odometry dataset (seq 00). Bit-per-point *vs*. Car BBox AP@0.7, 0.7, 0.7 (↑), Pedestrian BBox AP@0.5, 0.5, 0.5 (↑), and Cyclist BBox AP@0.5, 0.5, 0.5 (↑), ATE RMSE Trans (↓), ATE RMSE Rot (↓), RPE RMSE Trans Per M (↓), and RPE RMSE Rot Per M (↓) are shown from top to bottom, left to right.

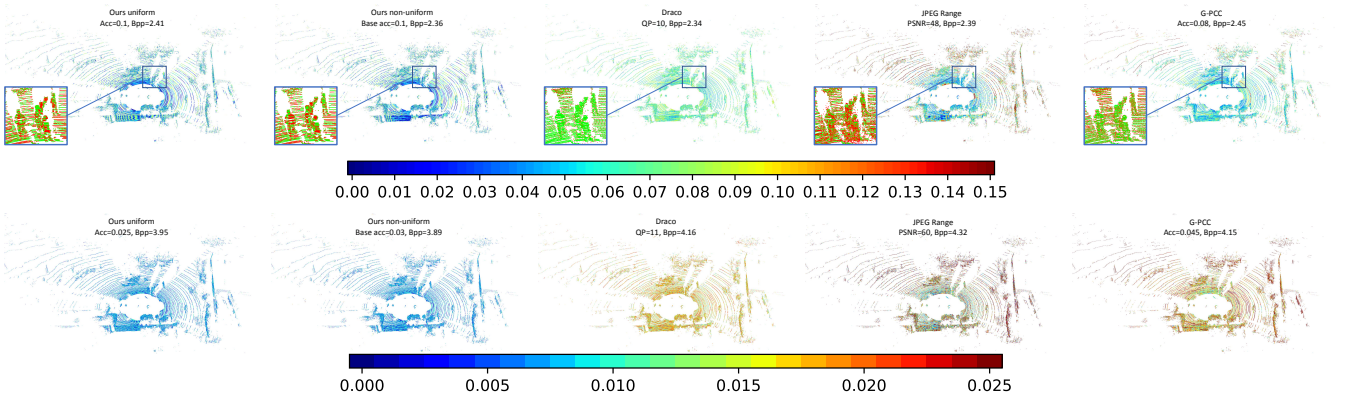

Fig. 7: The qualitative results of our uniform or non-uniform reconstructed point cloud compared with other baseline methods on the KITTI dataset. The color in the point cloud and colorbar are based on the mean symmetric chamfer distance (Eq. (4)) between the reconstructed and original point cloud.

like pedestrians or cyclists. The point-lossless feature of our framwork is another advantage in hard object detection.

The second row of Fig. 6 presents the comparative results of A-LOAM on the KITTI sequence 00 along with different BPPs. The results show that the SLAM performance is less related to the reconstruction quality, which means we can obtain lossless SLAM results with a higher compression ratio and a less-compressed bitrate. The experimental results show that our framework can achieve about 40× compression ratio without affecting downstream tasks (The intersection of the our method and the green line in Fig. 6).

E. Qualitative Results

In this section, we show the reconstructed point cloud with the original point cloud in Fig. 7. The color bar shows the error between the reconstructed and original point cloud. The first parameter in the caption is the setting of each method, and the BPP is the bitrate. We can find that when BPP is near 2, the points in reconstructed point cloud of the pedestrians are almost the same as in the original point cloud.

V. Conclusion

Our proposed uniform and non-uniform range image-based compression method can be seen as a baseline for large-scale lossless point cloud compression. The segmentation result in our framework will only influence the compression ratio but not the reconstruction quality. In downstream tasks such as 3D object detection and SLAM, our method can obtain a compressed bitstream of smaller size than the other baseline methods when the point cloud is lossless. Also, our non-uniform point cloud compression framework can be merged with machine learning for better salience map creation. By assigning a larger accuracy loss to the unimportant areas, we could obtain a higher compression ratio in future work.

REFERENCES

[1] J. Wang, D. Ding, Z. Li, and Z. Ma, "Multiscale point cloud geometry compression," in *2021 Data Compression Conference (DCC)*. IEEE, 2021, pp. 73–82.

[2] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression: A survey," in *The 24th International Conference on 3D Web Technology*, 2019, pp. 1–9.

[3] J. Peng, C.-S. Kim, and C.-C. J. Kuo, "Technologies for 3d mesh compression: A survey," *Journal of visual communication and image representation*, vol. 16, no. 6, pp. 688–733, 2005.

[4] Google, "Draco: 3D Data Compression," *https://github.com/google/draco*, 2018.

[5] Z. Que, G. Lu, and D. Xu, "Voxelcontext-net: An octree based framework for point cloud compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 6042–6051.

[6] S. Biswas, J. Liu, K. Wong, S. Wang, and R. Urtasun, "Muscle: Multi sweep compression of lidar using deep entropy models," *Advances in Neural Information Processing Systems*, vol. 33, pp. 22 170–22 181, 2020.

[7] X. Sun, H. Ma, Y. Sun, and M. Liu, "A novel point cloud compression algorithm based on clustering," *IEEE Robotics and Automation Letters*, vol. 4, no. 2, pp. 2132–2139, 2019.

[8] X. Sun, S. Wang, M. Wang, Z. Wang, and M. Liu, "A novel coding architecture for lidar point cloud sequence," *IEEE Robotics and Automation Letters*, vol. 5, no. 4, pp. 5637–5644, 2020.

[9] X. Sun, S. Wang, and M. Liu, "A novel coding architecture for multi-line lidar point clouds based on clustering and convolutional lstm network," *IEEE Transactions on Intelligent Transportation Systems*, 2020.

[10] L. Huang, S. Wang, K. Wong, J. Liu, and R. Urtasun, "Octsqueeze: Octree-structured entropy model for lidar compression," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 1313–1323.

[11] G. K. Wallace, "The jpeg still picture compression standard," *IEEE transactions on consumer electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[12] M. Rabbani, "Jpeg2000: Image compression fundamentals, standards and practice," *Journal of Electronic Imaging*, vol. 11, no. 2, p. 286, 2002.

[13] J. Elseberg, D. Borrmann, and A. Nüchter, "One billion points in the cloud–an octree for efficient processing of 3d laser scans," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 76, pp. 76–88, 2013.

[14] P. de Oliveira Rente, C. Brites, J. Ascenso, and F. Pereira, "Graph-based static 3d point clouds geometry coding," *IEEE Transactions on Multimedia*, vol. 21, no. 2, pp. 284–299, 2018.

[15] X. Zhang, W. Wan, and X. An, "Clustering and dct based color point cloud compression," *Journal of Signal Processing Systems*, vol. 86, no. 1, pp. 41–49, 2017.

[16] L. Tang, F.-p. Da, and Y. Huang, "Compression algorithm of scattered point cloud based on octree coding," in *2016 2nd IEEE International Conference on Computer and Communications (ICCC)*. IEEE, 2016, pp. 85–89.

[17] H. Houshiar and A. Nuchter, "3d point cloud compression using conventional image compression for efficient data transmission," in *XXV International Conference on Information*, 2015.

[18] J.-K. Ahn, K.-Y. Lee, J.-Y. Sim, and C.-S. Kim, "Large-scale 3d point cloud compression using adaptive radial distance prediction in hybrid coordinate domains," *IEEE Journal of Selected Topics in Signal Processing*, vol. 9, no. 3, pp. 422–434, 2014.

[19] P. Zanuttigh and G. M. Cortelazzo, "Compression of depth information for 3d rendering," in *2009 3DTV Conference: The True Vision-Capture, Transmission and Display of 3D Video*. IEEE, 2009, pp. 1–4.

[20] J. Zhang and S. Singh, "Loam: Lidar odometry and mapping in real-time." in *Robotics: Science and Systems*, vol. 2, no. 9, 2014.

[21] T. Shan and B. Englot, "Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 4758–4765.

[22] J. Jiao, H. Ye, Y. Zhu, and M. Liu, "Robust odometry and mapping for multi-lidar systems with online extrinsic calibration," *IEEE Transactions on Robotics*, 2021.

[23] J. Serafin, E. Olson, and G. Grisetti, "Fast and robust 3d feature extraction from sparse point clouds," in *2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2016, pp. 4105–4112.

[24] T. Qin and S. Cao, "Advanced implementation of loam," *https://github.com/HKUST-Aerial-Robotics/A-LOAM*, 2019.

[25] H. Fan, H. Su, and L. J. Guibas, "A point set generation network for 3d object reconstruction from a single image," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 605–613.

[26] T. Huang and Y. Liu, "3d point cloud geometry compression on deep learning," in *Proceedings of the 27th ACM International Conference on Multimedia*, 2019, pp. 890–898.

[27] R. Mekuria, S. Laserre, and C. Tulvan, "Performance assessment of point cloud compression," in *2017 IEEE Visual Communications and Image Processing (VCIP)*. IEEE, 2017, pp. 1–4.

[28] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *2017 IEEE International Conference on Image Processing (ICIP)*. IEEE, 2017, pp. 3460–3464.

[29] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.

[30] P. Sun, H. Kretzschmar, X. Dotiwalla, A. Chouard, V. Patnaik, P. Tsui, J. Guo, Y. Zhou, Y. Chai, B. Caine *et al.*, "Scalability in perception for autonomous driving: Waymo open dataset," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 2446–2454.

[31] Z. Zhang and D. Scaramuzza, "A tutorial on quantitative trajectory evaluation for visual (-inertial) odometry," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2018, pp. 7244–7251.

[32] W. Maddern, G. Pascoe, C. Linegar, and P. Newman, "1 Year, 1000km: The Oxford RobotCar Dataset," *The International Journal of Robotics Research (IJRR)*, vol. 36, no. 1, pp. 3–15, 2017.

[33] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The kitti dataset," *The International Journal of Robotics Research*, vol. 32, no. 11, pp. 1231–1237, 2013.

[34] 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.

[35] M. Group, "MPEG G-PCC TMC13," *https://github.com/MPEGGroup/mpeg-pcc-tmc13*, 2020.