Presenting a Novel Pipeline for Performance Comparison of V-PCC and G-PCC Point Cloud Compression Methods on Datasets with Varying Properties

Albert Christensen, Daniel Lehotský, Mathias Poulsen and Thomas Moeslund
Visual Analysis and Perception Lab, Aalborg University, Rendsburggade 14, DK-9000, Aalborg, Denmark

Keywords: Point Cloud Compression, 3D Compression, V-PCC, G-PCC, Draco, Performance Comparison Pipeline.

Abstract: The increasing availability of 3D sensors enables an ever increasing amount of applications to utilize 3D captured content in the form of point clouds. Several promising methods for compressing point clouds have been proposed but lacks a unified method for evaluating their performance on a wide array of point cloud datasets with different properties. We propose a pipeline for evaluating the performance of point cloud compression methods on both static and dynamic point clouds. The proposed evaluation pipeline is used to evaluate the performance of MPEG's G-PCC octree RAHT and MPEG's V-PCC compression codecs.

1 INTRODUCTION

With the increasing availability of 3D sensors such as LiDARs, time-of-flight cameras, stereo cameras etc. more objects and scenes are captured as point clouds. Point clouds are being used in a wide array of applications and tasks such as autonomous driving (Li et al., 2021), virtual and augmented reality (Bruder et al., 2014), object scanning (Chen et al., 2016), scene scanning (Ingale and J., 2021) and object detection and segmentation (Bello et al., 2020).

A point cloud is a simple data structure consisting of a list of points containing 3D geometric information and additional attribute information such as colours, normals, and reflectance, with no correlation between the points. Since objects and scenes captured as point clouds can contain millions of points, the storage and bandwidth requirements are often unfeasible. Therefore, there exists a need for effective point cloud compression codecs.

Several methods for compressing point clouds have been proposed. Amongst these, the most notable are Draco by Google[1], and the G-PCC (Mammou et al., 2019) and V-PCC (MPEG, 2020) proposed by the Moving Pictures Expert Group (MPEG), with compression standards from the *Joint Photographic Expert Group* (JPEG) still underway (JPEG, 2020).

In their call for proposals (MPEG, 2017), MPEG differentiate between 3 types of point clouds - static objects and scenes, dynamic objects, and dynamic point cloud acquisition. While this categorization of point clouds are relevant for the use cases of MPEG's compression codecs, different point cloud compression methods might benefit from being evaluated on a wider set of datasets with a different set of properties, reflecting other use cases.

G-PCC was made for static objects and scenes and dynamic point cloud acquisition, while V-PCC was made to compress dynamic objects. G-PCC, V-PCC (Schwarz et al., 2019)(Li et al., 2020)(Kim et al., 2020)(Liu et al., 2020) and compression codecs made since are often evaluated on the 8iVSLF dataset (Krivokuca et al., 2018), while before the standardization efforts of MPEG, compression algorithms would often be evaluated on completely different datasets (Huang et al., 2008)(Fan et al., 2013)(Mekuria et al., 2017a). Comparison between different methods are further complicated by the datasets used for evaluation of some deep learning methods for compression, as they are often different (Quach et al., 2019)(Que et al., 2021).

The work of (Wu et al., 2020) aimed to solve some of these problems, by introducing their *PCC arena* framework which compared different compression methods on multiple datasets. This work is limited to static datasets only and the compression of geometry only. PCC arena is the work closest to ours.

In this paper, we propose a comprehensive pipeline for evaluating the compression performance of codecs in a reproducible manner that allows for easy comparison on a diverse set of publicly available point cloud datasets. Both dynamical and static datasets are used with and without RGB attribute in-

[1] https://github.com/google/draco

Christensen, A., Lehotský, D., Poulsen, M. and Moeslund, T.
Presenting a Novel Pipeline for Performance Comparison of V-PCC and G-PCC Point Cloud Compression Methods on Datasets with Varying Properties.
DOI: 10.5220/0010820200003124
In *Proceedings of the 17th International Joint Conference on Computer Vision, Imaging and Computer Graphics Theory and Applications (VISIGRAPP 2022) - Volume 4: VISAPP*, pages 387-393
ISBN: 978-989-758-555-5; ISSN: 2184-4321

formation. We demonstrate the evaluation pipeline with the V-PCC, G-PCC (RAHT / Octree) and Draco compression codecs. The framework is modular and can be extended with more datasets and more compression codecs. Finally, we also propose an objective metric describing the density of a point cloud as well as a new dataset for dense static point cloud compression named MIA-Heritage.

2 METHOD

When evaluating the performance of point cloud compression codecs, a set of descriptive performance metrics have to be chosen and evaluated on a diverse set of benchmark datasets. We propose an evaluation pipeline capable of comparing the compression performance of various compression codecs on both dynamic and static datasets, with a set of quality metrics that allows for direct comparison.

2.1 Evaluation Metrics

Evaluation metrics are needed to describe the performance of the different compression methods. To this end we use objective quality metrics, which describes the quality of the reconstructed point cloud after compression and decompression.

The comparison pipeline adopts the point-to-point quality metric (D1), proposed by (Mekuria et al., 2017b), as it is a widely used objective quality metric (Schwarz et al., 2019)(Graziosi et al., 2020)(Kim et al., 2020). D1 is the peak signal-to-noise ratio (PSNR) reported in dB for geometry and the 3 colours in the YCrCb colour spectrum respectively. The error is the Euclidean distance in the geometric dimension and the colour dimensions for the nearest neighbours between the original and the degenerated point cloud. Instead of using 3 different colour values, one for each YCrCb channel as MPEG suggest, it has been chosen to use the Euclidean distance in YCrCb colour space to calculate one merged metric for the colour PSNR. Using a single value to represent the colour PSNR allows for easier and more direct comparison of the codec's performance on compressing the colour attributes.

The Bjøntegaard-Delta (BD) metrics (Bjøntegaard, 2001) are commonly used in the video codec community for comparing the performance of two different compression codecs against each other. It is also widely used for comparing lossy point cloud codecs at various bit rates (Gu et al., 2019) (Santos et al., 2021) (Wang et al., 2021) (Xiong et al., 2021). There exist two BD-metrics:

- BD-PSNR: The average PSNR difference in dB for the same bit rate.
- BD-Rate: The average bit rate difference in percent to produce the same PSNR.

The BD-metrics are found by fitting the computed D1 PSNR values for their corresponding bit rate values to a third degree logarithmic which is done for each codec. The integral difference between codecs can then be computed, and the average difference can be found by dividing the integral difference over a distance. We adopt the BD-PSNR metric as the primary evaluation metric.

The bits per input point (bpp) is chosen in favor of the conventionally used bit rate. Bpp is a normalized bit rate, making it easier to compare compression rates of point clouds of various sizes. Bpp describes how many bits on average that is required to represent a single point of the input point cloud. The number of points in the input point cloud is used instead of the number of points in the output point cloud as the number of points might be reduced during the compression process.

2.2 Point Cloud Properties

Point clouds differ in their properties with their intended usage. The compression performance of a compression method might therefore vary between point clouds with different properties. We propose a set of 5 binary point cloud properties for categorizing different point clouds.

- **Type** (Scene / Object)
- **Density** (Sparse / Dense)
- **Temporality** (Static / Dynamic)
- **View** (Single-view / Multi-view)
- **Information** (Geometry / Geometry + Attributes)

One element from each pair can be combined with any other element from the following pairs. By combining these characteristics, it is possible to generalise point cloud types. The first property, **type**, describes what is captured by the point cloud, as point clouds can be divided into scenes or single objects (Wu et al., 2020). A scene can be considerably more complex than a single object. The second property, **density**, describes the number of points. Dense point clouds have a lot of points per unit area, while sparse point clouds are relative thinly distributed. The third property, **temporality**, describes whether it is a single static frame or a dynamic point cloud sequence of frames (Cao et al., 2019). The fourth property, **view**, describes the angle from which the point cloud was captured. Multi viewpoint clouds provide up to a 360°

view of the captured scene or an object. Finally, the property **information** relates to the information encoded within the data. Geometry means that the point cloud only consists of XYZ data, while geometry + other attributes could be a combination of XYZ and RGB data. However, point clouds may also include additional information such as normals, reflectivity, etc.

2.3 Surface Density

To the best of the authors knowledge the surface *density* property is poorly defined in the literature. Thus, we propose *surface density* as the metric that quantifies the *density* property of a point cloud once it has been voxelized. Surface density for a voxelized point cloud is defined as the mean of the distance in voxels to the nearest neighbour for each voxel in the point cloud, see Equation (1).

$$P_{sd} = \frac{1}{n}\sum_{i=1}^{n} ||v_{nn} - v_i|| \tag{1}$$

Where P_{sd} is the surface density of point cloud P, n is the number of voxels in P, and v is a voxel in P, while v_{nn} is the nearest neighbour to v_i measured as euclidian distance expressed in voxels.

Surface density of a dataset is computed by taking the mean surface density of all point cloud sequences in a dataset, see equation (2).

$$D_{sd} = \frac{1}{k}\sum_{i=1}^{k} S_i \tag{2}$$

Where D_{sd} is the surface density of a dataset, and S_i is the mean surface density of a point cloud sequence, with k being the number of point cloud sequences in a dataset.

The surface density of the datasets used in this paper can be seen in Figure 1.

2.4 Evaluation Dataset

The compression codecs were evaluated on the datasets listed along with their properties in Table 1. A total of 5 different datasets were selected in order to have a broad range of dataset properties. The MIA-Heritage dataset has been created for the purpose of this paper and consist of 3 high quality scans from the Minneapolis Institute of Art [2]. The dataset was made to represent dense static objects. Additionally, a downsampled version of MIA-Heritage is included, where the voxelized point cloud is downsampled uniformly by only keeping every 50'th value in the voxel grid. The value of 50 was chosen by iteratively downsampling the point clouds more to obtain a difference in the surface density as seen in Figure 1. The density of the downsampled dataset is considered as sparse. This was done to compare the performance of the compression codecs on dense and sparse datasets where the only difference is the **surface density**, this dataset is referred to as MIA-Heritage-Low. Thus, the addition of the two MIA-heritage datasets creates a more diverse set of point cloud properties for evaluation.

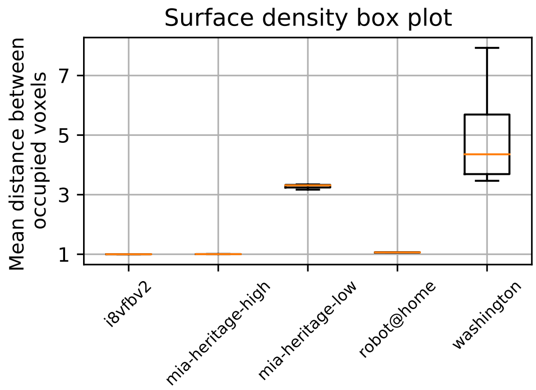

Figure 1: Box plots of the computed surface densities. Notice the small surface density in the dense datasets, *i8vfbv2*, *mia-heritage-high* and *robot@home*, compared to the sparse datasets *mia-heritage-low* and *washington*.

The compression codecs require all datasets to be voxelized. All datasets were voxelized to a bit depth of 10.

Frame examples of point cloud sequences from each dataset can be seen in Table 2.

2.5 Evaluation Pipeline

The open source evaluation pipeline [3], used for evaluating the different codecs with various bitrate configurations on the different datasets, can be seen in Figure 2.

The adopted metrics are calculated for all the individual point cloud sequences and averaged across each dataset and compression codec.

It has been chosen to average the evaluation metrics across a whole dataset since it is not necessary to evaluate the performance on each of the individual point cloud sequences. This decision assumes, that each of the datasets has similar point cloud properties.

[2] https://github.com/HuchieWuchie/mia-heritage

[3] https://github.com/math5581/PCCCP

Table 1: Overview of the test datasets along with their properties. The MIA-Heritage-Low is downscaled by a value of 50.

Dataset	Type	Density	Temporality	View	Information
8iVFB v2(d'Eon et al., 2017)	Object	Dense	Dynamic	Multi-view	Geometry + RGB
Robot@Home(Ruiz-Sarmiento et al., 2017)	Scene	Dense	Static	Multi-view	Geometry + RGB
MIA-Heritage[2]	Object	Dense	Static	Multi-view	Geometry + RGB
MIA-Heritage-Low[2]	Object	Sparse	Static	Multi-view	Geometry + RGB
Washington (Lai et al., 2011)	Object	Sparse	Dynamic	Single-view	Geometry + RGB

Table 2: (a)-(c) 8iVFB-v2, (d)-(f) MIA-Heritage (dense), (g)-(i) Robot@Home, (j)-(l) Robot@Washington RGB-D, (m)-(o) MIA-Heritage downsampled by factor 50.

3 RESULTS

In order to evaluate the performance of several codecs against each other using the BD metrics, an anchor is used. The anchor serves as a baseline for comparing the various codecs. For this report, Google's Draco was chosen as the anchor to compare MPEG's G-PCC and V-PCC against each other. Each of the tested codecs was made to compress a specific point cloud type. G-PCC was developed for compression of static point cloud objects and scenes. V-PCC was developed for compression of dynamic point cloud sequences, such as point cloud videos. Lastly, Draco was originally developed for mesh compression. Despite this, the comparison is made across all datasets. This is done to evaluate their performance on the properties defined in Section 2.2.

3.1 Draco Configurations

Draco [4] was evaluated with the configuration parameters given in Table 3.

Table 3: The Draco parameters that varied between the various bit rates. RX: various bitrate configurations. CL: compression level, QL: quantization level.

	R1	R2	R3	R4	R5	R6	R7	R8
CL	10	10	10	10	10	10	10	10
QL	6	7	8	9	10	11	12	13

3.2 G-PCC Configurations

G-PCC was evaluated with the octree transform for geometry compression and the region adaptive Haahr transform (RAHT) for attribute compression. Dynamic point clouds are compressed by compressing the frames individually. G-PCC was evaluated with the configuration parameters given in Table 4. The rest of the parameters were left at default. The latest reference implementation version 14.0 was used [5].

Table 4: The G-PCC parameters that varied between the various bit rates. RX: various bitrate configurations. PQ: Position quantization, LQ: Luma Quantization.

	R1	R2	R3	R4	R5	R6
PQ	0.125	0.25	0.5	0.75	0.875	0.9375
LQ	51	46	40	34	28	22

3.3 V-PCC Configurations

V-PCC was evaluated with the configuration parameters given in Table 5. Furthermore, the following configuration files where used in the written order *common/ctc-common.cfg* and *condition/ctc-all-intra.cfg* which can be found at the V-PCC reference

[4]https://github.com/google/draco
[5]https://github.com/MPEGGroup/mpeg-pcc-tmc13

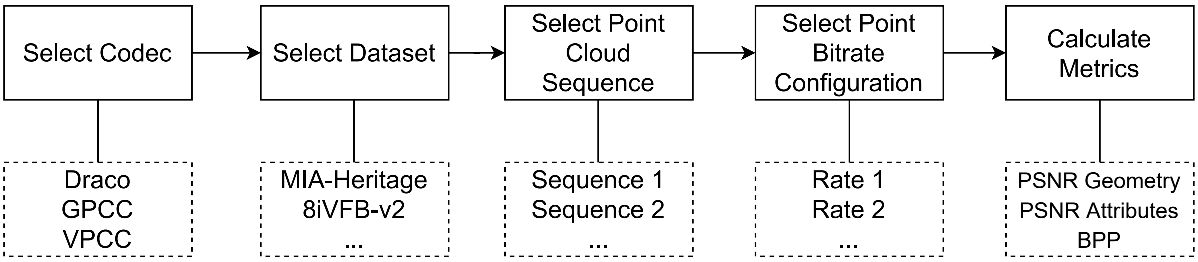

Figure 2: Illustration of the evaluation pipeline. The metrics are calculated for each combination of {*Codec, Dataset/Point Cloud Sequence* and *Bitrate*}.

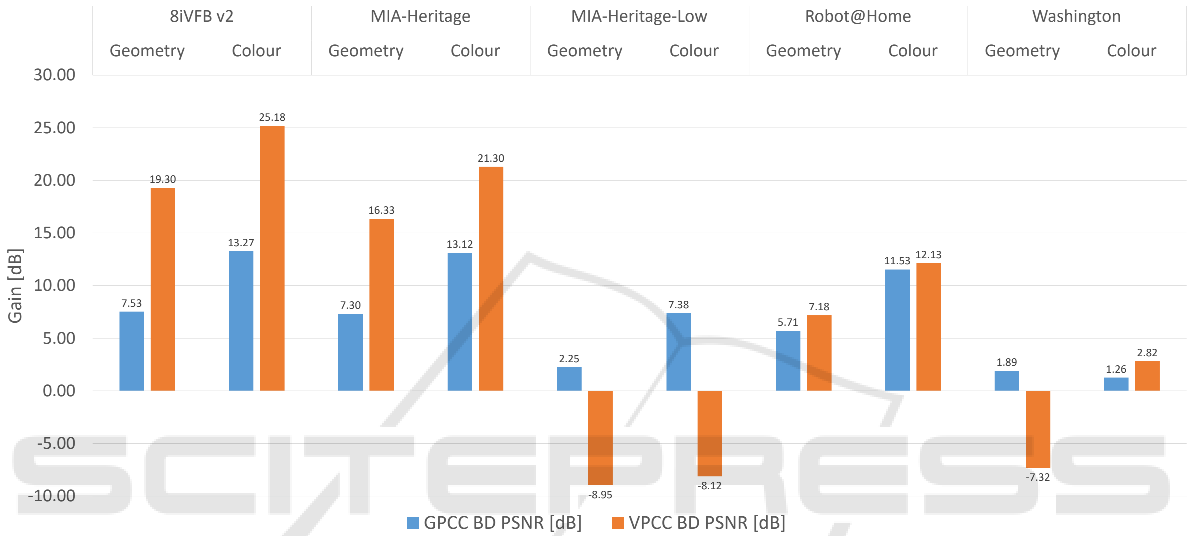

Figure 3: The collected results across the different datasets, with performance difference expressed via the BD PSNR values. The Draco codec was used as the anchor.

implementation github [6]. The latest reference implementation version 14.0 was used.

Table 5: The V-PCC parameters that varied between the various bit rates. RX: various bitrate configurations. GQ: Geometry Quantization, AQ: Attribute Quantization.

	R1	R2	R3	R4
GQ	32	24	20	0
AQ	42	32	27	-12

4 DISCUSSION

It can be seen in Figure 3 that MPEG's V-PCC outperforms both Draco and MPEG's G-PCC in terms of reconstruction quality of the decompressed point cloud for geometry and colour on the *Robot@home*, *8iVFB-v2* and *MIA-Heritage* datasets. Common to all of these datasets is that they consist of dense point clouds, where the distances between occupied voxels are low. V-PCC was designed for compressing dense dynamic object point clouds by utilizing optimized 2D video codecs to compress the colour information, which harmonizes well with the obtained results. This also explains why the reported BD-PSNR results for *Robot@Home* are lower compared to those for *8iVFB-v2* and *MIA-Heritage*, as those are objects and *Robot@Home* are scenes. However, V-PCC still performs better than Draco and G-PCC in terms of reconstruction quality on the *Robot@Home* dataset.

It can also be seen from Figure 3 that V-PCC compresses geometric information of sparse point clouds, such as those found in the *MIA-Heritage-Low* and the *Washington* datasets poorly, compared to G-PCC and Draco. This is possibly because V-PCC makes use of 2D projection methods that require a high resolution for the 2D projection planes, which is not true for sparse point clouds.

The results on the *Washington* dataset suggests that V-PCC is not only troubled by compressing sparse point clouds but also by compressing the geometric information in single-view point clouds. This

[6]https://github.com/MPEGGroup/mpeg-pcc-tmc2

might be because the V-PCC codec utilizes at least 6 projection planes which are redundant for single-view point clouds. The normal vector segmentation in the V-PCC algorithm should have been able to take care of this issue, however, the results suggest otherwise. G-PCC achieves higher BD PSNR gains for geometry which suggest that G-PCC is better suited for compressing single-view point clouds.

As seen in Figure 3, MPEG's G-PCC outperforms both Draco and V-PCC on both *MIA-Heritage-Low* and *Washington* in terms of both geometry and colour reconstruction quality. For the *Washington* dataset Draco performs also most identical to the G-PCC and has a higher BD PSNR gain than V-PCC for geometry reconstruction quality. Both the *MIA-Heritage-Low* and the *Washington* datasets consist of sparse point clouds where the distances between occupied voxels are large, see Figure 1. For G-PCC the octree data structure was chosen since it was made to compress sparse point clouds well which seems to be confirmed by the results obtained in this paper.

It is worth noting that while G-PCC outperforms Draco on the *MIA-Heritage-Low* and the *Washington* datasets, it does so by a margin that is smaller than those obtained on *8iVFB-v2* and *MIA-Heritage*. This might suggest that Draco is also suitable for compression of sparse point clouds. Both Draco and G-PCC outperform V-PCC heavily when compressing geometric information in sparse point clouds. Furthermore, Figure 3 shows that G-PCC performs well on a wide variety of datasets and does so consistently, e.g. G-PCC performs almost similar on dense object and dense scene datasets.

5 CONCLUSION

This paper proposes a set of binary properties to describe point clouds and argues that point cloud compression methods should be evaluated on a diverse set of datasets with different properties. To this end, an evaluation framework with an associated open source evaluation pipeline has been proposed with publicly available datasets.

Furthermore we also propose the MIA-Heritage dataset as a static dense point cloud compression dataset benchmark, as well as a metric for surface density to evaluate whether a point cloud is sparse or dense.

The evaluation of 3D compression methods finds that V-PCC provides good reconstruction quality on dense static and dense dynamic point clouds. It performs the strongest on objects but also outperforms Draco and G-PCC on dense scenes in terms of reconstruction quality. V-PCC is outperformed by G-PCC and Draco on sparse datasets, with Draco and G-PCC performing somewhat equally. Furthermore, the results suggest that V-PCC is challenged on single-view datasets in terms of geometric reconstruction quality.

REFERENCES

Bello, S. A., Yu, S., and Wang, C. (2020). Review: deep learning on 3d point clouds. *CoRR*, abs/2001.06280.

Bjøntegaard, G. (2001). Calculation of average psnr differences between rd-curves.

Bruder, G., Steinicke, F., and Nüchter, A. (2014). Poster: Immersive point cloud virtual environments. In *2014 IEEE Symposium on 3D User Interfaces (3DUI)*, pages 161–162.

Cao, C., Preda, M., and Zaharia, T. (2019). 3d point cloud compression: A survey. In *The 24th International Conference on 3D Web Technology*, pages 1–9.

Chen, L.-C., Hoang, D.-C., Lin, H.-I., and Nguyen, T.-H. (2016). Innovative methodology for multi-view point cloud registration in robotic 3d object scanning and reconstruction. *Applied Sciences*, 6(5).

d'Eon, E., Harrison, B., Myers, T., and Chou, P. A. (2017). 8i voxelized full bodies-a voxelized point cloud dataset. *ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006*, 7:8.

Fan, Y., Huang, Y., and Peng, J. (2013). Point cloud compression based on hierarchical point clustering. In *2013 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference*, pages 1–7.

Graziosi, D., Nakagami, O., Kuma, S., Zaghetto, A., Suzuki, T., and Tabatabai, A. (2020). An overview of ongoing point cloud compression standardization activities: video-based (v-pcc) and geometry-based (g-pcc). *APSIPA Transactions on Signal and Information Processing*, 9:e13.

Gu, S., Hou, J., Zeng, H., Yuan, H., and Ma, K.-K. (2019). 3d point cloud attribute compression using geometry-guided sparse representation. *IEEE Transactions on Image Processing*, 29:796–808.

Huang, Y., Peng, J., Kuo, C.-C. J., and Gopi, M. (2008). A generic scheme for progressive point cloud coding. *IEEE Transactions on Visualization and Computer Graphics*, 14(2):440–453.

Ingale, A. K. and J., D. U. (2021). Real-time 3d reconstruction techniques applied in dynamic scenes: A systematic literature review. *Computer Science Review*, 39:100338.

JPEG (2020). Final call for evidence on jpeg pleno point cloud coding. ISO/IEC JTC 1/SC 29/WG 1 (ITU-T SG16).

Kim, J., Im, J., Rhyu, S., and Kim, K. (2020). 3d motion estimation and compensation method for video-based point cloud compression. *IEEE Access*, 8:83538–83547.

Krivokuca, M., Chou, P. A., and Savill, P. (2018). 8i voxelized surface light field (8ivslf) dataset. *ISO/IEC JTC1/SC29/WG11 MPEG, input document m42914*.

Lai, K., Bo, L., Ren, X., and Fox, D. (2011). A large-scale hierarchical multi-view rgb-d object dataset. In *2011 IEEE international conference on robotics and automation*, pages 1817–1824. IEEE.

Li, L., Li, Z., Zakharchenko, V., Chen, J., and Li, H. (2020). Advanced 3d motion prediction for video-based dynamic point cloud compression. *IEEE Transactions on Image Processing*, 29:289–302.

Li, Y., Ma, L., Zhong, Z., Liu, F., Chapman, M. A., Cao, D., and Li, J. (2021). Deep learning for lidar point clouds in autonomous driving: A review. *IEEE Transactions on Neural Networks and Learning Systems*, 32(8):3412–3432.

Liu, H., Yuan, H., Liu, Q., Hou, J., and Liu, J. (2020). A comprehensive study and comparison of core technologies for mpeg 3-d point cloud compression. *IEEE Transactions on Broadcasting*, 66(3):701–717.

Mammou, K., Chou, P. A., Flynn, D., Krivokuća, M., Nakagami, O., and Sugio, T. (2019). G-pcc codec description v2. *ISO/IEC JTC1/SC29/WG11 N18189*.

Mekuria, R., Blom, K., and Cesar, P. (2017a). Design, implementation, and evaluation of a point cloud codec for tele-immersive video. *IEEE Transactions on Circuits and Systems for Video Technology*, 27(4):828–842.

Mekuria, R., Laserre, S., and Tulvan, C. (2017b). Performance assessment of point cloud compression. In *2017 IEEE Visual Communications and Image Processing (VCIP)*, pages 1–4.

MPEG (2017). Call for proposals for point cloud compression v2. ISO/IEC JTC1/SC29/WG11 MPEG2017/N16763.

MPEG (2020). V-pcc codec description. ISO/IEC JTC 1/SC 29/WG 7.

Quach, M., Valenzise, G., and Dufaux, F. (2019). Learning convolutional transforms for lossy point cloud geometry compression. *2019 IEEE International Conference on Image Processing (ICIP)*.

Que, Z., Lu, G., and Xu, D. (2021). Voxelcontext-net: An octree based framework for point cloud compression.

Ruiz-Sarmiento, J., Galindo, C., and Gonzalez-Jimenez, J. (2017). Robot@home, a robotic dataset for semantic mapping of home environments. *The International Journal of Robotics Research*, 36(2):131–141.

Santos, C., Gonçalves, M., Corrêa, G., and Porto, M. (2021). Block-based inter-frame prediction for dynamic point cloud compression. In *2021 IEEE International Conference on Image Processing (ICIP)*, pages 3388–3392. IEEE.

Schwarz, S., Preda, M., Baroncini, V., Budagavi, M., Cesar, P., Chou, P. A., Cohen, R. A., Krivokuća, M., Lasserre, S., Li, Z., Llach, J., Mammou, K., Mekuria, R., Nakagami, O., Siahaan, E., Tabatabai, A., Tourapis, A. M., and Zakharchenko, V. (2019). Emerging mpeg standards for point cloud compression. *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, 9(1):133–148.

Wang, J., Ding, D., Li, Z., and Ma, Z. (2021). Multiscale point cloud geometry compression. In *2021 Data Compression Conference (DCC)*, pages 73–82. IEEE.

Wu, C.-H., Hsu, C.-F., Kuo, T.-C., Griwodz, C., Riegler, M., Morin, G., and Hsu, C.-H. (2020). Pcc arena: a benchmark platform for point cloud compression algorithms. *Proceedings of the 12th ACM International Workshop on Immersive Mixed and Virtual Environment Systems*.

Xiong, J., Gao, H., Wang, M., Li, H., and Lin, W. (2021). Occupancy map guided fast video-based dynamic point cloud coding. *IEEE Transactions on Circuits and Systems for Video Technology*.