Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Distortion Metrics in Segmented LiDAR Point Clouds

Alexsander Correa de Oliveira

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Orientador
Prof. Dr. Bruno Luiggi Macchiavello Espinoza

Brasília
2023

Universidade de Brasília
Instituto de Ciências Exatas
Departamento de Ciência da Computação

Distortion Metrics in Segmented LiDAR Point Clouds

Alexsander Correa de Oliveira

Monografia apresentada como requisito parcial
para conclusão do Bacharelado em Ciência da Computação

Prof. Dr. Bruno Luiggi Macchiavello Espinoza (Orientador)
CIC/UnB

Prof. Dr. Diogo Caetano Garcia
Universidade de Brasília

Renam Castro da Silva
Universidade de Brasília

Prof. Dr. Marcelo Grandi Mandelli
Coordenador do Bacharelado em Ciência da Computação

Brasília, 28 de Julho de 2023

Dedicatória

Na *dedicatória* o autor presta homenagem a alguma pessoa (ou grupo de pessoas) que têm significado especial na vida pessoal ou profissional. Por exemplo (e citando o poeta): *Eu dedico essa música a primeira garota que tá sentada ali na fila. Brigado!*

Agradecimentos

Nos *agradecimentos*, o autor se dirige a pessoas ou instituições que contribuíram para elaboração do trabalho apresentado. Por exemplo: *Agradeço aos gigantes cujos ombros me permitiram enxergar mais longe. E a Google e Wikipédia.*

O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES), por meio do Acesso ao Portal de Periódicos.

Resumo

O *resumo* é um texto inaugural para quem quer conhecer o trabalho, deve conter uma breve descrição de todo o trabalho (apenas um parágrafo). Portanto, só deve ser escrito após o texto estar pronto. Não é uma coletânea de frases recortadas do trabalho, mas uma apresentação concisa dos pontos relevantes, de modo que o leitor tenha uma ideia completa do que lhe espera. Uma sugestão é que seja composto por quatro pontos: 1) o que está sendo proposto, 2) qual o mérito da proposta, 3) como a proposta foi avaliada/validada, 4) quais as possibilidades para trabalhos futuros. É seguido de (geralmente) três palavras-chave que devem indicar claramente a que se refere o seu trabalho. Por exemplo: *Este trabalho apresenta informações úteis a produção de trabalhos científicos para descrever e exemplificar como utilizar a classe LaTeX do Departamento de Ciência da Computação da Universidade de Brasília para gerar documentos. A classe* `UnB-CIC` *define um padrão de formato para textos do CIC, facilitando a geração de textos e permitindo que os autores foquem apenas no conteúdo. O formato foi aprovado pelos professores do Departamento e utilizado para gerar este documento. Melhorias futuras incluem manutenção contínua da classe e aprimoramento do texto explicativo.*

Palavras-chave: LaTeX, metodologia científica, trabalho de conclusão de curso

Abstract

O *abstract* é o resumo feito na língua Inglesa. Embora o conteúdo apresentado deva ser o mesmo, este texto não deve ser a tradução literal de cada palavra ou frase do resumo, muito menos feito em um tradutor automático. É uma língua diferente e o texto deveria ser escrito de acordo com suas nuances (aproveite para ler `http://dx.doi.org/10.6061%2Fclinics%2F2014(03)01`). Por exemplo: *This work presents useful information on how to create a scientific text to describe and provide examples of how to use the Computer Science Department's LaTeX class. The `UnB-CIC` class defines a standard format for texts, simplifying the process of generating CIC documents and enabling authors to focus only on content. The standard was approved by the Department's professors and used to create this document. Future work includes continued support for the class and improvements on the explanatory text.*

Keywords: LaTeX, scientific method, thesis

Sumário

Referências

Distortion Metrics in Segmented LiDAR Point Clouds

1st Alexsander Correa de Oliveira
Departamento de Ciência da Computação (CiC)
Universidade de Brasília (UnB)
Brasília, Brasil
alexsanderco@outlook.com

***Abstract*— Point cloud compression has been the object of wide study recently, and has seen big leaps in development, such as MPEG's standartization in the form of TMC13 G-PCC. Concurrently, segmentation has also been steadly improving and showing big promise to real-world applications. Segmentation of compressed point clouds, however, have received little attention as a research object. In this paper, we propose two different metrics to tackle this open problem, with the objective of measuring distortion both in terms of geometric errors and in terms of segmentation. In order to test the metrics proposed, we first made lossy compressions to our dataset, using TMC13 and then segmented them with RandLA-Net. To the best of our knowledge, this is the first paper proposing content quality as a distortion metric. The performance of our metrics are tested and validated against an accepted distortion metric, quantity of points and bitrate.**

I. Introduction

Light Detection and Ranging (LiDAR) is a sensor that works by scanning its surrounding space into a list of points, indicating spatial coordinates and other aspects, such as reflectance and color.

LiDAR is a flexible technology, and as such, has been used in several areas [1]–[3] showing great potential in automobile automation [4]. However, the proposals and developments made in this regard have not yet been adapted into the market. This fact is seen in companies such as Tesla, which currently use image processing as the base for their products [5].

Even though LiDAR studies for autonomous cars are newer than image processing studies in the same context [6], there is an effort in building and researching new technologies based on LiDAR [7]. One of the fields of interest is segmentation, which is the process of classifying objects in the input data. This means that for autonomous cars using LiDAR, each sensor scan requires segmentation of the data being collected, and only after that, the automobile can make decisions [7].

The developments regarding segmentation can be seen in competitions such as SemanticKITTI's [8], which offers an open dataset for segmentation tasks, and offers a list displaying the best performing networks, with regards to the intersection over union (mIOU). After two years of developments the mIOU has risen from 55.9% to 72.9% and time performance has fallen by an order of magnitude [9], [10].

Although developments in segmentation have been made, there are some concerns regarding the usage of these technologies in a real-world application, such as the processing times of the point clouds. For example, if we consider the Moving Pictures Experts Group's (MPEG) Ford_01_q_1mm, scans were performed at 100 ms intervals, so that there is less than 100ms available for point cloud processing. When only considering segmentation times from networks such as RandLA-NET, which was marketed as time-efficient, they take 40ms in average for point clouds with 10^5 points with a NVIDIA 2080TI [9]. However, such performance may not be sufficient, as autonomous cars have other tasks, redundancy built into them, and a bigger focus on accuracy as a safety-critical application [11].

One solution to this time complexity problem is to lossy compress our data, as to quantize our input point cloud with some information loss. This has already been used in research to decrease processing hurdles, and may be obligatory in high-traffic scenarios [12]. Although this encompasses the time problem, after compressing other data, such as images and videos, artifacts emerge, directly affecting segmentation quality [13], [14]. In images and videos there is a fixed ammount of pixels available, and each one has a set position. With this constraint, the artifacts are due to the fact that attribute information has been lost [15]. However, in point clouds the complexity of this analysis is higher, as contrary to images, spatial coordinates are not fixed, and as such, there may be not only attribute artifacts but also differences in spatial information. This fact can be analyzed by looking at point cloud sizes, such as in Table I, in which after compression in codecs such as MPEG's TMC13 Geometry Point Cloud Compression (G-PCC), numerous points are lost after decompression.

This means that for compressed images, segmentation errors are bound pixel-wise, and analysis is made by comparing the original's pixel to the compressed one in the same position in the image. In compressed point clouds, this assessment can't be done by simply analyzing points at the same spatial coordinates, as after decompression there is no guarantee each point remains at the same place.

In this context, there is at least one study on the quality of segmentation of compressed point clouds [16], but none, of our knowledge, that tries to give new tools to analyze segmentation quality as a distortion metric.

The remainder of this paper is structured as follows. In Section II, a typical pointcloud pipelined process is presented,

	Original	r01	r02	r03	r04	r05	r06
Average Point Cloud Size	82,627.105	14,389.334	29,533.026	69,685.673	78,033.783	80,739.55	81027.786

TABLE I: Table displaying average point cloud size in all 1500 frames in the dataset Ford_01_q_1mm. r01-r06 are quantization levels provided by TMC13 GPCC, and they go from most quantized to least.

up until the moment in which it is ready for use. Section III is used to show and explain each metric's equations, as well as briefly state some differences between the two metrics. Section IV contains the data and analysis of each metric's performance. And finally in Section V we conclude our paper and briefly discuss possible future works.

II. Point Cloud Processing Pipeline

The point cloud processing pipeline will be illustrated with MPEG's Ford_01_q_1mm dataset. For time performance sake's, all the tests where made in a computer with the following specifications: Intel(R) Core(TM) i7-8700 CPU @ 3.20GHz processor, 24 GB of RAM memory, GeForce GTX 1080 8 GB Graphics card. The compression method was TMC13's octree-RAHT, that is, the codec uses the octree data structure and RAHT transform [17].

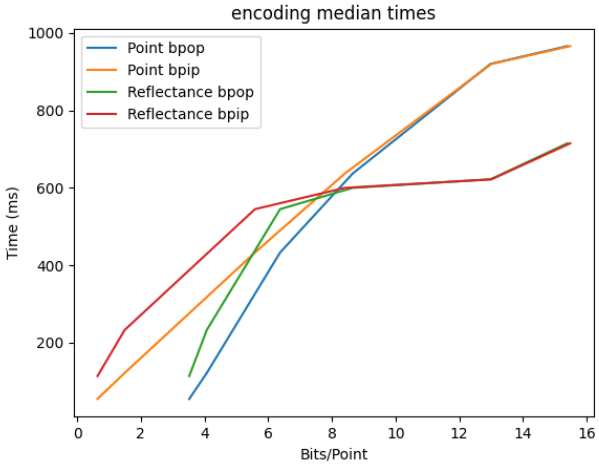

Fig. 1: Median time spent compressing, in terms of bits per point. Bpip and bpop are bits/input point and bits/output point respectively. What this means is, the first uses the ammount of points of the original point cloud and the latter uses the ammount of points of the decompressed point cloud.

A. Data Acquisition

Every frame in the operation of the autonomous car is defined by a complete scan of the surrounding area by a LiDAR sensor. As is indicated in the Table I, initially, the point clouds captured, in any instant, has aproximately 82 thousand points. Each point has a 25-byte representation, with twelve bytes for (x, y, z) coordinates, one for reflectance and the other 12 are for the normal coordinates (nx,ny,nz). In other words, every scan produces a file size 2.05 megabytes!

This quantity of information is worry inducing, as we can see in Figures 1 and 2, the difference in processing times

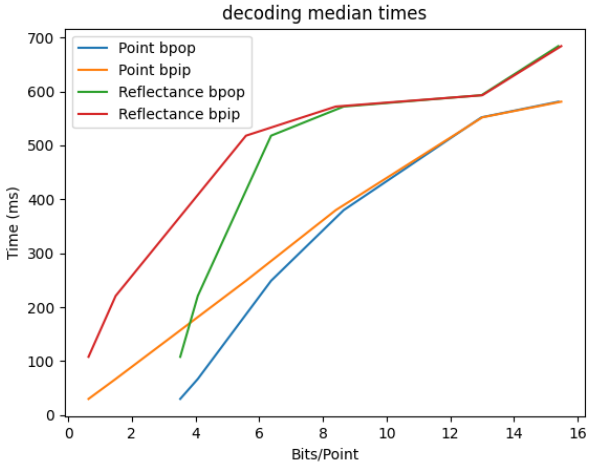

Fig. 2: Median time spent decompressing, in terms of bits per point.

of each quantization value is big and indicates a necessity for the compression of the point clouds, otherwise, processing the point cloud will take up much more than 2 frames only considering the enconding step. Add to this the fact that this dataset presents lower density point clouds, if compared to other widely used ones [8], [18], and compression appears even more attractive.

B. Compressing the point cloud

As said earlier in Section II, the *codec* chosen was the TMC13 G-PCC, which is MPEG's test model. Even though its time performance needs improvement, as other methods have made developments in that area [12], we can still use the data regarding processing times, due to time constraints in the automobile environment [11].

For the compression task all we need is the input point cloud, received from the LiDAR sensor. With it in hands we compress it with G-PCC in any of the six quantization levels offered by the test model. As we can see in Figures 1 and 2, the levels with least quantization take too much time to make processing these point clouds in real time possible.

C. Transmiting the compressed point cloud

As the compressed point cloud is made available, there are several applications in which it is useful as is. For example, in the context of level 4 and 5 automation [19], one of the procedures made is the communication between vehicles [19]. Preferentially with the compressed file, as the state-of-the-art of data transmission vehicle-to-vehicle (V2V) is too limited in a city traffic environment [12].

In our case, we only use the compression as a step to make educated guesses to which point to delete or reallocate with minimal information loss.

D. Decompressing the point cloud

After compressing, and potentially transmitting the data, we can decompress our point cloud. In our case, this operation is done using TMC13 as well.

With the decompressed point cloud in hands, some changed aspects can be seen, such as its size, which has shrunk, and attribute information point-wise, which may have been lost. This implies that the geometry of objects in the scanned area possibly had their geometries changed. One additional caveat of G-PCC's compression is that the normal coordinates are scrapped, making each point only occupy 13 bytes versus the original 25.

E. Point cloud segmentation

Finally, to fully process the point cloud, we just need to segment it. RandLANet, in its creation, showed great time performance and accuracy [9]. For these reasons, it was chosen for our tests.

Here, we are only using a semantic segmentation on the point cloud, and so make no distinction among objects of the same class.

When segmenting the point cloud, we receive the data in the form of labels for everything and each point gets classified under 19 different classes, ranging from cars, roads to pedestrians and buildings [9].

With the decompressed point cloud and label data in hands, all information is available and no further procedures are needed.

III. Content Quality as a Distortion Metric

As briefly stated in Section I, point clouds present more singularities in the context of objective quality assessments of segmented decompressed data than videos and images, because there is reallocation and deletion of points, contrary to the fixed ammount of pixels in images. This fact makes it so that there are two major concerns which we want to tackle in this paper:

- Compressed point clouds change the geometry of objects.
- Each point may be incorrectly segmented. For example, a point which was originally a part of a car was labelled as something else, such as a tree, after compression and segmentation.

This being the case, we propose two distinct metrics, with each addressing a different problem.

For both metrics, we define $P = \{p_1, p_2 ... p_N\}$ as a LiDAR point cloud, in which $p_i = (x_i, y_i, z_i)$, with $0 < i <= N$ and (x, y, z) representing some point in 3D volumetric space.

After segmentation, each point $p \in P$ is related to some class k, where K indicates the total ammount of available classes and $0 < k <= K$. Therefore, $P_k = \{p \mid p \in P \wedge p \in k\}$.

The process of lossy compression receives P as an input as has as an output $Q = \{q_1, q_2 ... ql\}$, where l can be of any value, lower, higher or equal to N. $q_j = (x_j, y_j, z_j)$ and $0 < j <= l$. It is to be noted that for any q_j it may or may not be equal to any p_i.

For the comparison to be made between Q and P, we need some form of quantization of P and sorting, as to compare the two point clouds as closely as possible. For this, lets define some function $Nearest(P, Q) = \bigcup_{i=1}^{l} \{u \mid u \in P \wedge \forall t \in P dist(u, q_i) <= dist(t, q_i)\}$, where dist is euclidean distance.

A. Class-MSE

For the geometry distortion task, we propose the usage of D1(MSE_sym), but imbued in a segmentation context. And finally, our metric, which is defined as:

$$Class\text{-}MSE(F, G, k) = \frac{1}{size(F)} \sum_{c=1}^{size(F)} MSE_{sym}((F_c)_k, (G_c)_k),$$

where F is a collection of point clouds, and G is a collection of all point clouds of F but lossy compressed, F_c *and* G_c are point clouds, $0 < c <= size(F)$.

This metric has the objective of analyzing intraclass distortion for compressed point clouds. We achieve this by comparing the original segmented point cloud to a decompressed and segmented version of itself.

It is to be noted that different classes tend to operate under different magnitudes of values. This means that we can't guarantee trivially that comparisons between Class-MSE of two different classes are fair, as shown in Section IV.

B. Point Cloud Cross Entropy

For the problem of segmentation mistakes in compressed point clouds, we propose the usage of Cross Entropy to compare the point clouds. It is defined as $CrossEntropy(S, H) = \sum_{e=1}^{size(S)} S_e * log(H_e)$. The two inputs for Cross Entropy are matrixes composed of vectors of probabilities. In our case, each vector has the following property: $sum(vector) = 1$.

This data was obtained during segmentation, that is, instead of getting a single label for each point, here, we acquire the chance of the point of belonging to any class.

Considering this, our metric is defined as:

$$PCCE(F, G) = \frac{1}{size(F)} \sum_{c=1}^{size(F)} CrossEntropy(Nearest(F_c, G_c)_s, (G_c)_s),$$

where s denotes that we are getting the matrix of softmax probabilities from some point cloud.

IV. Experimental results

In this section, for each metric proposed, we'll analyze the data collected from experimentation on the dataset as well as briefly highlighting some of the problems of using only geometric distortions as basis for analysing the segmentation of compressed point clouds.

		r01	r02	r03	r04	r05	r06
Class 1	Total Points	640,228	1,946,443	5,402,181	5,733,759	6,431,068	6,605,259
	MSE average	157.150	31.354	12.115	11.444	8.472	8.026
Class 2	Total Points	1,411	2,991	92,200	79,027	96,072	105,589
	MSE average	7,789.692	5,843.317	1,720.703	2,061.760	1,773.026	1,512.234
Class 3	Total Points	4,618,773	10,887,819	14,750,715	11,038,163	17,105,011	19,027,484
	MSE average	16.176	6.692	5.375	13.507	1.189	0.983
Class 4	Total Points	752	17,196	2,117,199	3,509,807	1,924,200	1,980,027
	MSE average	6,603.984	2,066.569	56.805	46.359	24.146	21.932
Class 5	Total Points	1,202,837	3,475,175	24,907,418	21,121,630	23,269,768	23,877,142
	MSE average	64.931	15.588	3.126	3.497	1.095	0.984
Class 6	Total Points	2,977,751	9,408,839	20,944,490	24,006,780	29,656,976	30,063,799
	MSE average	179.560	14.051	2.612	1.722	0.941	0.960
Class 7	Total Points	5,295,498	9,285,328	21,534,977	23,971,995	21,058,082	20,981,513
	MSE average	19.165	9.038	3.585	2.271	1.502	1.403
Class 8	Total Points	215,788	539,112	1,327,169	1,999,067	1,981,445	1,850,659
	MSE average	606.049	100.774	30.703	21.976	18.598	18.359
Class 9	Total Points	5,489,682	7,266,452	10,714,221	22,642,967	16,328,397	13,694,607
	MSE average	14.518	10.734	6.706	6.377	1.988	1.891
Class 10	Total Points	1,013,253	1,231,528	2,044,873	2,058,782	2,149,828	2,197,277
	MSE average	29.621	18.839	8.483	8.275	7.376	7.347
Class 11	Total Points	128,028	238,656	693,067	888,698	1,108,478	1,158,323
	MSE average	1,166.910	624.207	159.210	136.808	120.542	97.934

TABLE II: Comparison between all points in the dataset in each class and each quantization level, and the average MSE of each class and each quantization level

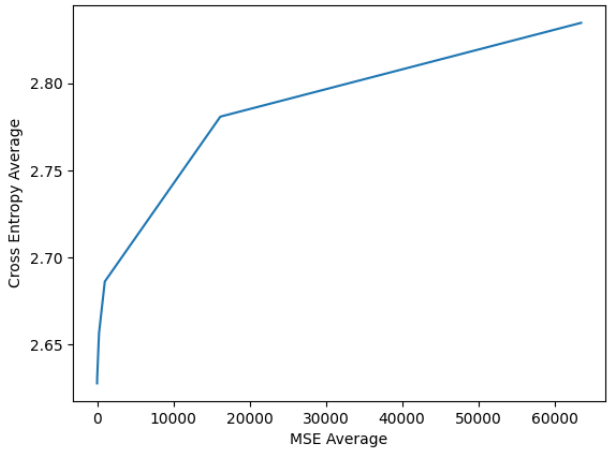

Fig. 3: Relation between average Cross entropy and average MSE(D1) of all 1500 frames of the dataset Ford_01_q_1mm in all GPCC quantization levels (r01-r06).

A. Class-MSE data

For this metric, the data was collected by segmenting all Ford_01_q_1mm frames, compressing and segmenting the same frames in all quantization levels from G-PCC. After this, we separate each class in each point cloud to its own file. This way we can trivially apply our Class-MSE metric. Figure 4 shows how the dataset was processed for the data's acquisition.

As per the results, we can see in Figure 5 that there is a trend that Class-MSE follows that of more points in a given point cloud tends to lower MSE values. This, however, isn't guaranteed as various points may be incorrectly segmented, leading to comparisons between true elements of a class and false elements of the same class. Which in turn means that we aren't necessarily comparing spatially close points, leading to higher error values.

Note that Table II has the data for every class, whereas 5 only contains information for the four first classes.

The convexity of this metric appears in most of the classes but with different intensities, as observed in Figure 6. Although the function is not monotonic in all cases, it appears to be stemming from the segmentation mistakes rather than from geometric distortions, as said earlier in this section.

B. Point Cloud Cross Entropy data

Differently from Class-MSE, for this metric we dont use one label for each point, but rather the softmax vector of probabilities, therefore there is no need to separate classes, and we only need to compare the segmentation of each compressed point cloud to the original.

This metric's average results in terms of bits/point can be seen in Figure 7. The graphic shows a monotic curve that is convex and descending. The observation to be made is that there is a relation between segmentation quality and quantization level.

In addition, comparing Cross Entropy trends to the MSE point-to-point (D1) averages from Figure 3 we see that, although both metrics are related to bits/point, and both are monotic, convex, and decreasing, the curves are different. This means that geometric distortions are related to segmentation quality, but we can't guarantee with certainty by how much.

V. Conclusion

In this paper, we proposed two metrics, both of which are inserted in distinct contexts within the distortion analysis of the segmentation of compressed point clouds. We showed that both approaches work, but one is used for geometric distortions, using MSE as our base, and the other for segmentation

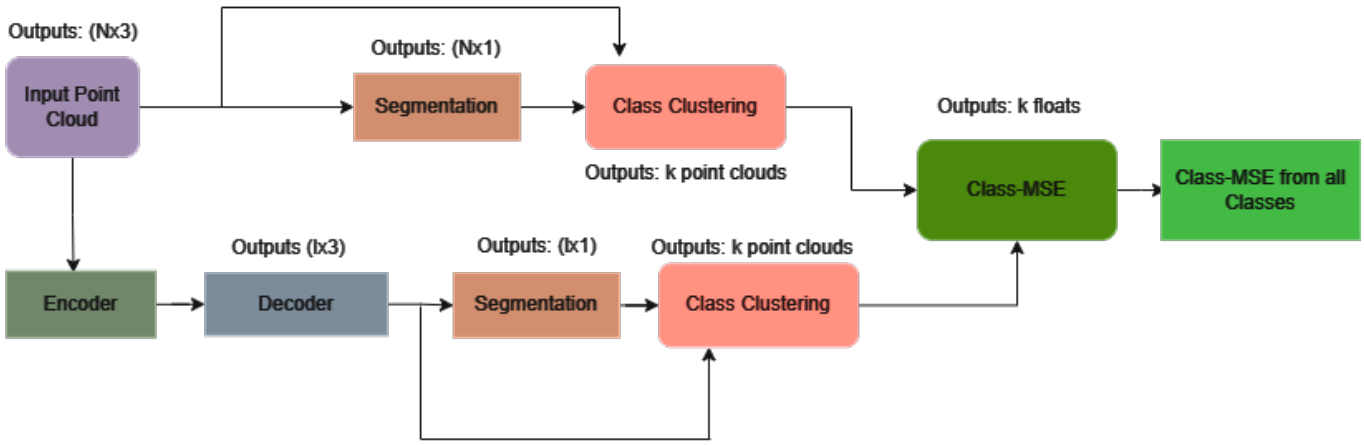

Fig. 4: Diagram showing each step to getting the Class-MSE for all classes of an input point cloud, for any quantization level.

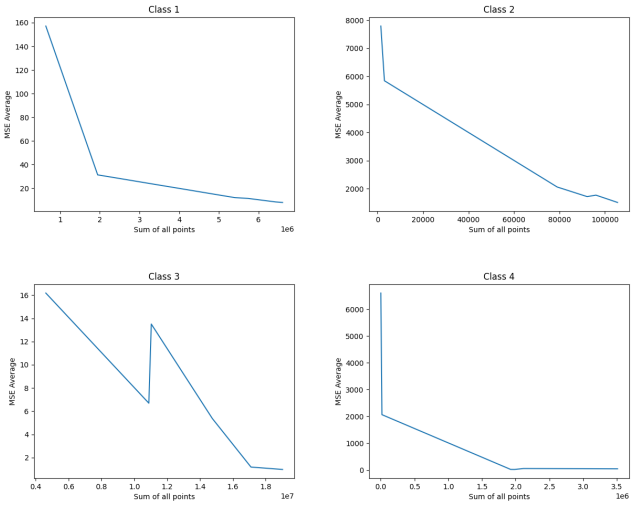

Fig. 5: MSE(D1) average in terms of total points in a given class in all dataset's frames combined.

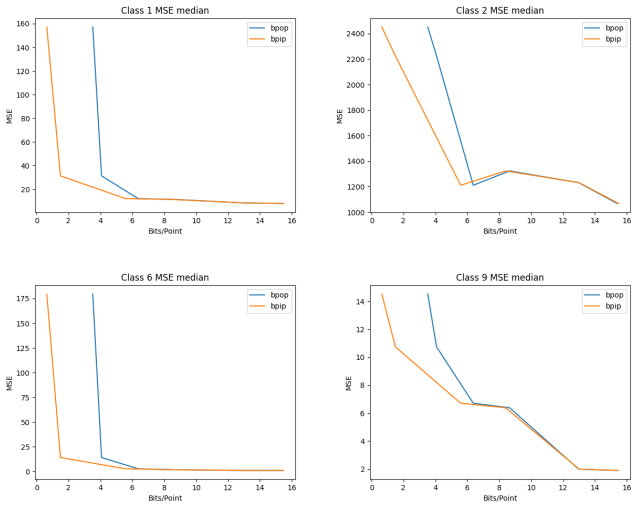

Fig. 6: Average Class-MSE for 4 separate classes: Vehicles, Cyclists and pedestrians, Buildings, and Road.

mistakes, using Cross entropy. Therefore, both should be used to arrive at more precise conclusions. Our tests demonstrated that the metrics are correlated to bitrate, with convexity being observed primarely in our Point Cloud Cross Entropy.

Further tests include the use of others codecs, segmentation tecnologies and LiDAR datasets.

References

[1] G. G. Goyer and R. Watson, "The Laser and its Application to Meteorology," *Bulletin of the American Meteorological Society*, vol. 44, no. 9, pp. 564–570, Sep. 1963. DOI: 10.1175/1520-0477-44.9.564.

[2] J. M. Bonisteel, A. Nayegandhi, C. W. Wright, J. Brock, and D. Nagle, "Experimental advanced airborne research lidar (EAARL) data processing manual," Tech. Rep. 2009-1078, 2009.

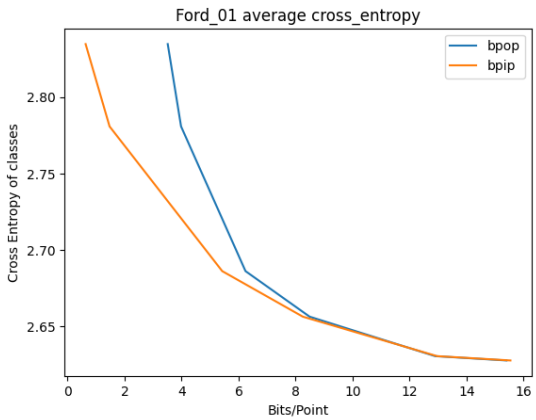

Fig. 7: Average Cross entropy in terms of bitspoint. The lower the value, the better.

[3] F. Amzajerdian, D. Pierrottet, L. Petway, G. Hines, and V. Roback, "Lidar systems for precision navigation and safe landing on planetary bodies," *Proceedings of SPIE - The International Society for Optical Engineering*, vol. 8192, Jun. 2011. DOI: 10.1117/12.904062.

[4] M. Elhousni and X. Huang, "A survey on 3d lidar localization for autonomous vehicles," in *2020 IEEE Intelligent Vehicles Symposium (IV)*, 2020, pp. 1879–1884. DOI: 10.1109/IV47402.2020.9304812.

[5] N. Nrip and B. Patil, "An analysis of the use of artificial intelligence in tesla and mercedes driverless cars," *International Journal of Innovative Research in Science Engineering and Technology*, vol. 11, pp. 13 407–13 411, Nov. 2022. DOI: 10.15680/IJIRSET.2022.1111022.

[6] M. Krödel and K.-D. Kuhnert, "Autonomous driving through intelligent image processing and machine learning," Oct. 2001, pp. 712–718, ISBN: 978-3-540-42732-2. DOI: 10.1007/3-540-45493-4_70.

[7] Y. Li and J. Ibanez-Guzman, "Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems," *IEEE Signal Processing Magazine*, vol. 37, no. 4, pp. 50–61, Jul. 2020. DOI: 10.1109/msp.2020.2973615. [Online]. Available: https://doi.org/10.1109%2Fmsp.2020.2973615.

[8] J. Behley, M. Garbade, A. Milioto, *et al.*, *Semantickitti: A dataset for semantic scene understanding of lidar sequences*, 2019. arXiv: 1904.01416 [cs.CV].

[9] Q. Hu, B. Yang, L. Xie, *et al.*, *Randla-net: Efficient semantic segmentation of large-scale point clouds*, 2020. arXiv: 1911.11236 [cs.CV].

[10] X. Yan, J. Gao, C. Zheng, *et al.*, *2dpass: 2d priors assisted semantic segmentation on lidar point clouds*, 2022. arXiv: 2207.04397 [cs.CV].

[11] S. Sudhakar, V. Sze, and S. Karaman, "Data centers on wheels: Emissions from computing onboard autonomous vehicles," *IEEE Micro*, vol. 43, no. 1, pp. 29–39, 2023. DOI: 10.1109/MM.2022.3219803.

[12] A. Varischio, F. Mandruzzato, M. Bullo, M. Giordani, P. Testolina, and M. Zorzi, *Hybrid point cloud semantic compression for automotive sensors: A performance evaluation*, 2021. arXiv: 2103.03819 [cs.NI].

[13] Y. Chen, A. Janowczyk, and A. Madabhushi, "Quantitative assessment of the effects of compression on deep learning in digital pathology image analysis," en, *JCO Clin Cancer Inform*, vol. 4, pp. 221–233, Mar. 2020.

[14] T. Gandor and J. Nalepa, "First gradually, then suddenly: Understanding the impact of image compression on object detection using deep learning," en, *Sensors (Basel)*, vol. 22, no. 3, Feb. 2022.

[15] A. Unterweger, "Compression artifacts in modern video coding and state-of-the-art means of compensation," *Multimedia Networking and Coding*, pp. 28–49, Jan. 2012. DOI: 10.4018/978-1-4666-2660-7.ch002.

[16] L. Garrote, J. Perdiz, L. A. da Silva Cruz, and U. J. Nunes, "Point cloud compression: Impact on object detection in outdoor contexts," *Sensors*, vol. 22, no. 15, 2022, ISSN: 1424-8220. DOI: 10.3390/s22155767. [Online]. Available: https://www.mdpi.com/1424-8220/22/15/5767.

[17] H. Liu, H. Yuan, Q. Liu, J. Hou, and J. Liu, *A comprehensive study and comparison of core technologies for mpeg 3d point cloud compression*, 2019. arXiv: 1912.09674 [eess.IV].

[18] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression: A survey," in *The 24th International Conference on 3D Web Technology*, ser. Web3D '19, LA, CA, USA: Association for Computing Machinery, 2019, pp. 1–9, ISBN: 9781450367981. DOI: 10.1145/3329714.3338130. [Online]. Available: https://doi.org/10.1145/3329714.3338130.

[19] J. Wang, H. Huang, K. Li, and J. Li, "Towards the unified principles for level 5 autonomous vehicles," *Engineering*, vol. 7, no. 9, pp. 1313–1325, 2021, ISSN: 2095-8099. DOI: https://doi.org/10.1016/j.eng.2020.10.018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2095809920304008.