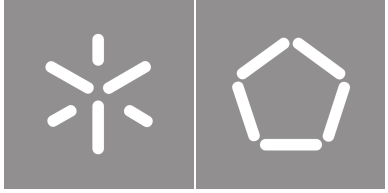
Universidade do Minho
Escola de Engenharia

Francisco André Oliveira Dias

ALFA-Pc: Streaming Point Cloud Data Compression

Janeiro de 2022



Universidade do Minho
Escola de Engenharia

Francisco André Oliveira Dias

ALFA-Pc: Streaming Point Cloud Data Compression

Dissertação de Mestrado
Mestrado em Engenharia Eletrónica Industrial e Computadores

Trabalho efetuado sob a orientação de:
Professor Doutor João Monteiro
Professor Doutor Tiago Gomes

Janeiro de 2022

Agradecimentos

Agradecimentos.....

Acknowledgments.....

STATEMENT OF INTEGRITY

I hereby declare having conducted this academic work with integrity. I confirm that I have not used plagiarism or any form of undue use of information or falsification of results along the process leading to its elaboration. I further declare that I have fully acknowledged the Code of Ethical Conduct of the University of Minho.

Resumo

Resumo....

Palavras-chave:

Abstract

Keywords:

Contents

List of Figures

List of Tables

Glossary

ADAS Advanced Driver-Assistance Systems

AG Angular Resolution

AMCW Amplitude Modulated Continuous Wave

ASPRS American Society of Photogrammetry and Remote Sensing

AVC Advanced Video Coding

CfP Call for Proposals

CNN convolution neural network

CW Continuous Waveform

DARPA Defense Advanced Research Projects Agency

DL Deep Learning

EDC Extended Delta Compression

FMCW Frequency Modulated Continuous Wave

FoV Field of View

FPGA Field Programmable Gate Array

G-PCC Geometry-based Point Cloud Compression

HEVC High Efficiency Video Coding

hFoV horizontal FoV

JPEG Joint Photographic Experts Group

JPEG-LS Joint Photographic Experts Group - Lossless

LiDAR Light Detection And Ranging

LZ77 Lempel-Ziv Coding 1977

LZMA Lempel–Ziv–Markov chain algorithm

MEMS Microelectromechanical

MPEG Moving Picture Experts Group

OPA Optical Phased Array

PNG Portable Graphics Format

RADAR RAdio Detection And Ranging

RAHT Region Adaptive Hierarchical Transform

ROI regions of interest

SAE Society of Automotive Engineers

SNR Signal-to-Noise Ratio

SWaP Size, Weight, and Power

ToF Time of Flight

V-PCC Video-based Point Cloud Compression

XOR Exclusive Disjunction Operation

1. Introduction

1.1 Motivation

1.2 Main Goal

1.3 Structure

2. Background and State of the Art

This chapter begins by discussing Light Detection And Ranging (LiDAR) technology in automotive field. Section 2.1.1 describes some of the fundamental principles of LiDAR technology, beginning with the working principle of a LiDAR sensor and the main measuring techniques, followed by some of the major metrics used to evaluate the performance of a LiDAR sensor. Section 2.1.2 discusses existing types of imaging techniques for collecting data from the environment, followed by Section 2.1.3, which describes some of the applications of LiDAR technology in the automotive field. Next, some of the challenges of LiDAR technology are addressed, and then in Section 2.2 several implementations related to the work presented in this dissertation are studied and discussed.

2.1 LiDAR Technology for Automotive

Since the last century, LiDAR has been a well-established measurement technology. Established as a useful and appropriate measurement tool for atmospheric aerosol research [1] and mapping of aeronautical and aerospace terrain [2, 3], it has evolved into one of the important technologies for autonomous vehicles [4, 5]. Despite the fact that LiDAR systems were already in use for military purposes, the use of this technology for the automotive industry only appeared with the Defense Advanced Research Projects Agency (DARPA) Grand Challenges in the year 2004 [6] and the winning of DARPA Challenge in the next year with Stanford's autonomous car, named Stanley, equipped with five SICK LiDARs [7]. Since then, the race to develop fully autonomous and self-driving vehicles has been on, with this technology being used in a variety of applications in recent years [8] resulting in an improvement of these LiDAR systems with a design that has remarkably low Size, Weight, and Power (SWaP)[9].

For a vehicle to navigate autonomously, it needs a perception system capable of providing an accurate mapping of the the surroundings of the vehicle. By combining RAdio Detection And Ranging (RADAR), Camera and LiDAR sensors it is possible to create a multi-sensor perception system offering the ability to navigate the environment effectively [10]. The combination of these technologies is critical to the success of the new automotive era [11] with the rising deployment of Advanced Driver-Assistance Systems (ADAS) by major automotive companies helping to achieve higher Society of Automotive Engineers (SAE) Levels of driving automation, ranging from no driving automation (level 0) to complete driving automation (level 5) [12]. Considering all of this, it is reasonable to conclude that LiDAR sensors are a fundamental technology

for the future of completely autonomous vehicles since they can offer a 3D representation of the vehicle's surroundings in real-time with high resolution.

2.1.1 Basic Concepts of a LiDAR System

A LiDAR system is composed by a transmitter that emits light signals within a range of wavelengths, and a receiver that receives the reflected light from the objects. With the round-trip travel time of the emitted signal, known as Time of Flight (ToF), it is possible to measure distances to objects as depicted in Fig. 2.1.

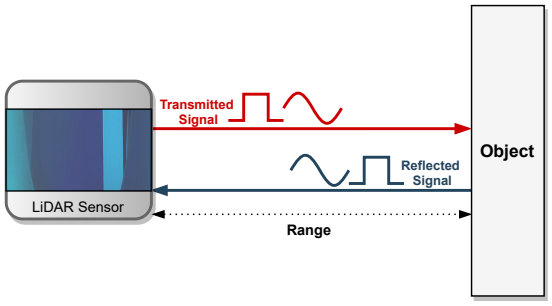


Figure 2.1: LiDAR working principle.

The maximum range, distance to the object, of a LiDAR system is calculated following equation 2.1 where c is the speed of light in the medium and τ is the round-trip time, which corresponds to the difference between the time of signal emission and reception of the reflected signal at the target. With the ToF calculated for each point, it is possible to construct a 3D representation of the vehicle's surroundings, known as the point cloud.

$$R = \frac{1}{2}c\tau \tag{2.1}$$

The value of τ can be obtained in a several ways, including pulsed or continuous wave measurement techniques [5]. Pulsed ToF measurement techniques use the time interval between the emitted laser pulse and the reflected laser pulse on the target to calculate the distance between them with the round-trip delay being calculated using timers [13], as depicted in Fig 2.2. Since the ToF is measured directly from the reflected pulse on the target, pulses must be emitted with high optical power and short as possible with quick rise and fall times [5].

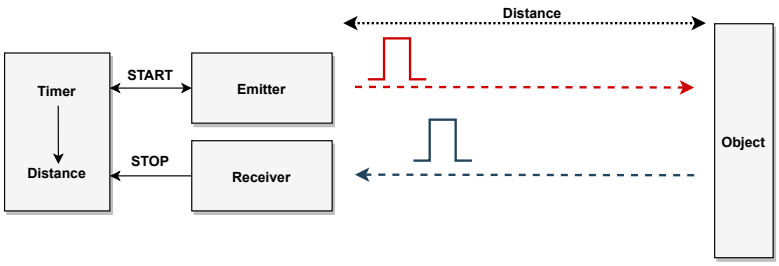


Figure 2.2: Pulsed ToF measurement technique.

The Amplitude Modulated Continuous Wave (AMCW) technique, Fig. 2.3, uses the Continuous Waveform (CW) modeling principle. This approach is similar to pulsed ToF but differs in transmission mode, which uses modulation of the intensity of light, transmitting continuous optical power [13]. The distance value is obtained by utilizing the phase-shift caused in an intensity-modulated periodic signal during its round-trip to the target [14].

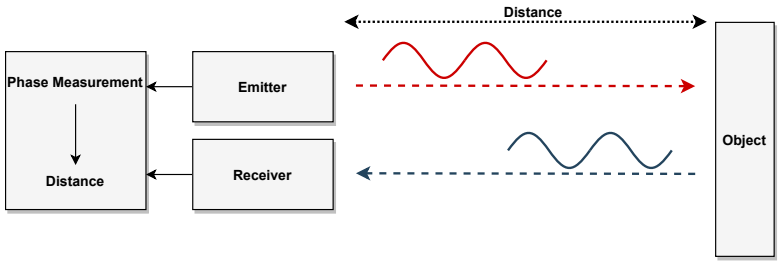


Figure 2.3: AMCW measurement technique.

The Frequency Modulated Continuous Wave (FMCW) approach, in contrary to the techniques described above, utilizes a method that generates short chirps of frequency modulated laser light [15]. By utilizing wave light properties, these systems can measure the phase and frequency of the return chirp allowing the system to calculate distance and velocity.

Although pulsed-wave techniques are easier and less expensive to implement, they lack precision in measurements, and their immunity to noise is compromised, with a generally low Signal-to-Noise Ratio (SNR) induced by mutual interference from external light sources [16]. Continuous-wave measurement techniques, on the other hand, produce better results, though there is always a trade-off between the quality of the results and the complexity of the electronic systems.

Regarding the measurement techniques used, there are certain crucial performance metrics for assessing LiDAR systems. Starting with Field of View (FoV) which is the area that can be observed by a LiDAR scan and is generally expressed in degrees [5], defining also the angle from which the LiDAR signals are emitted [13]. A 3D LiDAR sensor includes both horizontal and vertical planes, with some sensors capable of delivering 360° horizontal FoV (hFoV) by employing mechanical rotating components or by combining sensors with a smaller FoV and merging their output. Alongside FoV, Angular Resolution (AG) is an important metric as well. A LiDAR system's AG represents the smallest angle between two points that the sensor can distinguish. As a result, higher resolution sensors can generate denser point clouds with a larger volume of data.

Accuracy and precision are essential range measurements when it comes to LiDAR evaluation. Accuracy is a qualitative notion defined as the degree of agreement between a measurement's outcome and its true value [17]. As a result, LiDAR's output, a point cloud, is accurate when it is close to the true position of the environment. Precision can be described as the repeatability of measurement with a defined point cloud being produced by a precise LiDAR sensor, whereas noisy point clouds being produced by a low-precision LiDAR sensor. Finally, the frequency of time it takes for a LiDAR sensor to scan the complete

FoV is referred to as the frame rate. This metric is also important, particularly in real-time applications, since an automotive sensor must cope with real-time data from dynamic surroundings [18] with the frame rate being significantly influenced by the imaging technique employed.

2.1.2 LiDAR Imaging Systems

There are several imaging techniques [9] for representing the 3D environment around the vehicle in such a way that it is possible to form an image with a multitude of points, known as a point cloud, within an FoV. These imaging systems can be divided into beam steering sensors [19, 20, 21] and full solid-state sensors [22]. In the beam steering category, LiDAR systems can be mechanically rotated to scan the 3D environment or they can be solid-state, requiring no spinning mechanical components.

Rotor-based mechanical LiDAR systems can provide a 360° horizontal FoV. This is made possible by a spinning mechanical system that rotates the scanner component and a rotating mirror or prism resulting in a 3D 360° hFoV. Despite its widespread adoption in the automotive industry, this type of LiDAR sensor has significant disadvantages, including a hefty design, added inertia from the rotating part and also being the most expensive LiDAR system. In the beam steering solid-state category, technologies such as Microelectromechanical (MEMS) and Optical Phased Array (OPA) systems are employed. MEMS-based LiDAR systems employ MEMS technology instead of external rotating components to mechanically move the laser position within the FoV. Tiny electromechanical mirrors are employed for this purpose, with their tilt angle being adjusted by a stimulus such as voltage . In the OPA-based system, phase modulators are used to optically steer the laser beam. By controlling the phase of the optical emitters, the electromagnetic field near them can be completely controlled [23], thus resulting in a direction-controlled emitting light. Since solid-state systems do not have rotating mechanical elements, they are able to achieve higher frame rates but have a limited FoV.

In the full solid-state category, LiDAR systems do not contain mechanical moving components with Flash LiDAR being one of the most promising sensors [22]. The behavior of this system is similar to a camera in that a single laser pulse illuminates the environment in its front and a set of focal plane photodetectors placed in the vicinity of the laser captures the scattered light behind [15]. The data capture rate is much faster since this method captures all FoV points at the same time when compared to the mechanical laser scanning method. However they require more power to illuminate the entire FoV compared to other scanning techniques, thus limiting the maximum range.

2.1.3 LiDAR Applications

Although LiDAR sensors are new to the automotive industry, they are becoming increasingly relevant for autonomous driving since these sensors can represent the environment around them in real-time with high resolution. Fusing their data with other sensors such as RADAR and cameras helps getting closer to

a fully autonomous system with features such as pedestrian recognition and tracking [24, 25, 26], object detection and classification [27, 28, 29], and road perception [30, 31, 32].

Pedestrian Detection and Tracking

The accurate identification and tracking of pedestrians in autonomous vehicles and ADAS seek to improve vehicles' awareness for possible threats in their area enhancing traffic safety [33]. In [25] a multi-sensor system combining LiDAR, camera, and semantic information is presented for pedestrian recognition in urban situations. This approach provides improved detection performance by using a LiDAR sensor for object detection and generation of regions of interest (ROI), followed by an image-based module that validates the presence of pedestrians in the ROIs. More recently, in [26] a method for pedestrian recognition based on score fusion is proposed, merging data from two LiDARs and delivering extremely accurate pedestrian detection results in real-time, outperforming traditional raw data fusion algorithms in most circumstances.

Object Detection

Object detection is a critical task in an autonomous system since being unable to identify and distinguish road elements might lead to safety-related incidents [34]. In H. Cho et al. [28], a technique for identifying and tracking moving objects in urban driving environments is suggested, using a combination of RADAR and LiDAR sensors. This fusion demonstrates an enhancement and performance gain of the tracking system in terms of obstacle detection rate improving the previous work used in 2007 on the DARPA Urban Challenge. X. Zhao et al. [29] proposed an object detection and identification method that uses complementary data from a LiDAR sensor and a camera to achieve accurate detection while also obtaining target distance information. To provide precise object-region candidates a 3D LiDAR is employed, then these candidates are projected into image space where the ROI is selected and finally a convolution neural network (CNN) is used for additional object recognition.

Road Perception

The ability to recognize road boundaries such as curbs or lane marks in real-time is critical for an autonomous vehicle [35]. Y. Zhang et al. [31] proposes a real-time curb recognition approach using a LiDAR sensor to separate the road and determine its curbs. The point cloud data from the LiDAR sensor is processed to distinguish between on-road and off-road areas, and then the road is later segmented. Finally, the curb detection algorithm is used to determine the location of curbs for each road segment. In [32], a method for combining LiDAR and camera lane marker identification algorithms is suggested, with the goal of creating a lane map using point cloud data gathered independently from both sources. When the results of two algorithms for detecting lane markings are combined, the map construction and matching process becomes more effective and the performance of lane marking detection is improved.

2.1.4 LiDAR Challenges

Although LiDAR sensors are being increasingly used in the automotive industry, with one of the goals being autonomous driving, this technology still has to face some challenges that are being addressed both at the industrial and the academic level. Weather denoising is one of these challenges, as LiDAR sensors can be substantially influenced by bad temporal conditions [13] such as rain [36], fog [37], or snow [38]. This challenge is addressed in [10], where a new solution is presented in which two state-of-the-art algorithms are combined, giving a superior filtering ratio and improving processing time when implemented in an Field Programmable Gate Array (FPGA). Ground segmentation is another important challenge for road surface detection since it identifies and separates the ground from the point cloud points [13]. This is a key step for an autonomous vehicle since after the segmentation of the ground it allows tasks such as object tracking and 3D reconstruction to be performed more efficiently [39]. Even though there are several good performing ground segmentation methods, these still need to be improved for real-time applications [40].

Data Compression

An important challenge is dealing with the enormous amount of data that a LiDAR sensor can output, reaching a throughput of several Gbit/s in point clouds with high resolution [13] requiring a large portion of bandwidth for transmission. With the advancement of LiDAR technology, the surrounding environment can be represented with higher resolutions and wider FoVs, resulting in denser point clouds and an increase in the amount of data involved. As one hour of point cloud data from a Velodyne HDL-64 [41] LiDAR sensor potentially representing over 100GB of data, existing techniques do not have the capacity to transmit all of this data in its original format. As a result, developing methods for compressing the output of a LiDAR sensor becomes an essential task [42].

The goal of LiDAR data compression is to decrease the amount of data required to represent digital data gathered by the LiDAR system, and compression is achieved when redundancies, spatial and/or temporal, in the data are eliminated [43, 44]. Spacial redundancy occurs when elements inside a structure, such as pixels in an image/frame and sequences of data, are duplicated. When elements in two frames have the same values and are in the same location, it is referred to as temporal redundancy. Exploiting both of these redundancies is a critical step in the compression of streaming point clouds. There are several algorithms for compression that can minimize the size of the point cloud tackling spatial and/or temporal redundancy for data storage or streaming, with these algorithms being categorized into lossy or lossless schemes [45]. Lossy compression techniques reduce the data size by removing useless information, whereas lossless compression techniques maintain the original content while removing redundant statistical data [46]. Although data compression might add overhead to the processing unit and thus increase processing time [45], transmitting point cloud data in real-time is a challenging task due to the sheer volume of data involved, making compression an essential step before transmission [44]. That being the case, the work

presented in this dissertation addresses this challenge by using data compression algorithms for point cloud data streaming purposes.

2.2 Related Work

This dissertation focuses on the analysis and development of an FPGA-based approach for streaming point cloud data compression for LiDAR sensors in the automotive industry. In the following section, some implementations that tackle LiDAR data compression will be analyzed and discussed in order to understand what sort of algorithm is ideal for addressing the challenge of streaming point cloud data compression. Fig. 2.4 depicts some of the existing categories and their respective compression algorithms that will be analyzed in this work. In addition to approaches that use low-level compression algorithms [18, 47, 48] and methods based on the LAS format [49, 50], new techniques are emerging, with space decomposition [51, 52, 53] and projection-based ones [44, 54, 55] having been proved in the literature to be efficient for encoding dense and sparse point clouds [56]. Also, some Deep Learning (DL)-based approaches [44, 53, 57] have begun to disclose some breakthroughs in efficient point cloud compression.

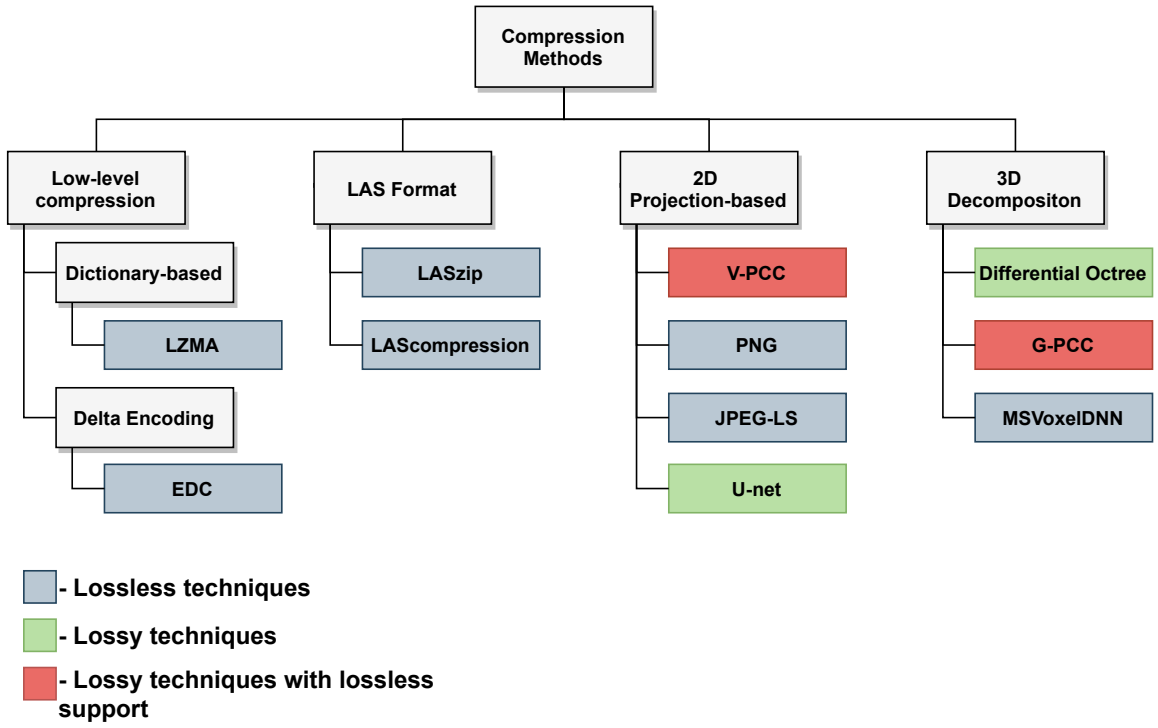


Figure 2.4: Data compression methods overview.

2.2.1 Low-level Compression

There are several low-level compression methods that could be used for compressing raw LiDAR data. These techniques can be based on entropy encoding or data differentiating algorithms. Entropy encoding

is a lossless coding technique that replaces data elements with coded representations that in conjunction with transformation and quantization, results in substantially decreased data size [58]. Approaches based on dictionaries are a type of entropy encoding schemes, being beneficial in instances when the original data has more repeating patterns. When a pattern is encountered in the input sequence, it is coded with an index to the dictionary. When a pattern is not found in the dictionary, it is coded using any less efficient methods. It is classified into two types: frequently and infrequently recurring patterns [59]. However, it may have certain limitations in terms of storage because multiple iterations are required to form the dictionary, which must then be stored or transmitted together with the compressed data for effective data decompression [48]. Delta encoding is a method of storing or transmitting data that makes use of estimated differences (deltas) between files accepting as input old and new versions of a file and then computing the difference between them. This method can be employed for data compression when the values in the original information are uniform, that is, when there is normally only a minor difference between consecutive values. [60].

In [61] is demonstrated an FPGA implementation of the Lempel–Ziv–Markov chain algorithm (LZMA) [47], an entropy-based algorithm. This algorithm compresses unknown data using a sliding window method and a dynamic dictionary. Fig 2.5 depicts the stages of the LZMA algorithm where the delta encoding stage organizes the incoming data stream for efficient compression via the sliding window, which stores or transmits data as differences between consecutive data rather than entire files. Then the dynamic dictionary based on the Lempel-Ziv Coding 1977 (LZ77) [62] algorithm is applied, coding byte sequences from former contents rather than the original data. Finally, to achieve higher compression ratios a range encoder is used, which encodes all of the message's symbols into a single number [61]. With this method, it is possible to achieve a compression ratio of roughly 50% [18]. In [48], a lossless compression technique called Extended Delta Compression (EDC) is suggested, which is based on a derivative of the delta encoding algorithm [63]. When compared to other low-level compression methods, the results demonstrated the capacity to compress LiDAR raw data with good compression ratios, however these results were obtained in a simulation and further work would be required to optimize this approach for LiDAR sensors in a real scenario.

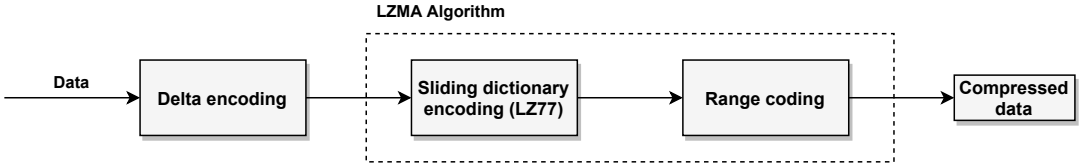


Figure 2.5: LZMA algorithm overview.

2.2.2 LAS-based Methods

To address the issue of LiDAR data compression, the American Society of Photogrammetry and Remote Sensing (ASPRS) developed a standard binary exchange format known as the LAS format [64] to enable the

practical interchange of LiDAR data. This format has evolved throughout time, but the following sections remain constant: (1) a mandatory single header containing generic LiDAR metadata identifying the project, the LAS format version, the number and type of points captured, and dimensions of X, Y, Z range ; (2) variable-length records including extra information such as projection information, metadata, and user application data; and (3) a series of fixed-length point data records containing X, Y, and Z coordinates as well as other point properties such as GPS time or point type [45, 64]. Following the release of the LAS format, methods such as LAScompression [49] and LASzip [50] have emerged to compress LiDAR points stored in LAS format in a lossless manner. These algorithms use prediction coding to compress blocks of LiDAR points forecasting the attributes of a new point from a group of past points [45] with the correcting deltas being subsequently compressed using arithmetic coding [65] as depicted in Fig. 2.6. Although these are mature methods used in LiDAR systems, they do not provide real-time compression and require the point cloud to be transformed to LAS format before compression can occur.

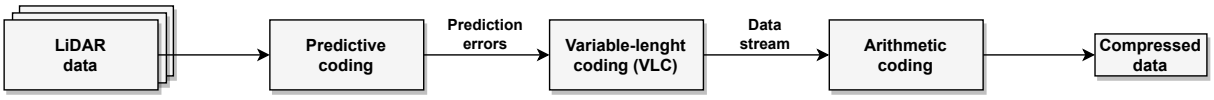


Figure 2.6: LAScompression algorithm overview.

2.2.3 2D Projection-based Techniques

In the projection-based compression methods, the 3D point cloud data is converted into 2D images or patches by projection or mapping. Then this 2D images or patches are compressed using efficient coding technologies such as standard image or video codecs.

For more than three decades, the Moving Picture Experts Group (MPEG) has been creating internationally successful compression standards such as Advanced Video Coding (AVC) and High Efficiency Video Coding (HEVC) [55]. MPEG initially investigated the usage of point clouds for immersive telepresence applications in 2013, and further debates on how to compress this sort of data took place. In 2017, a Call for Proposals (CfP) [66] was released and based on the submissions to this CfP, two separate compression methods were chosen for point cloud compression being one of them Video-based Point Cloud Compression (V-PCC). V-PCC is a lossy technique with lossless support and uses a projection-based approach as it converts the point cloud data from 3D to 2D and then the resultant is encoded by 2D video encoders thus taking advantage of 2D video compression [55]. The 3D geometry of the point cloud and its attributes are transformed into 2D patches which are then mapped to 2D planes using orthogonal projections. Then these 2D patches are placed on a 2D image of size W x H. Furthermore, an occupancy map is generated to indicate which sections of the image may or may not be used later. Finally, after the creation of those patches and the 2D frames, both geometry and attributes can be compressed using HEVC codec [67]. In Fig 2.7 is depicted a block diagram of V-PCC encoder.

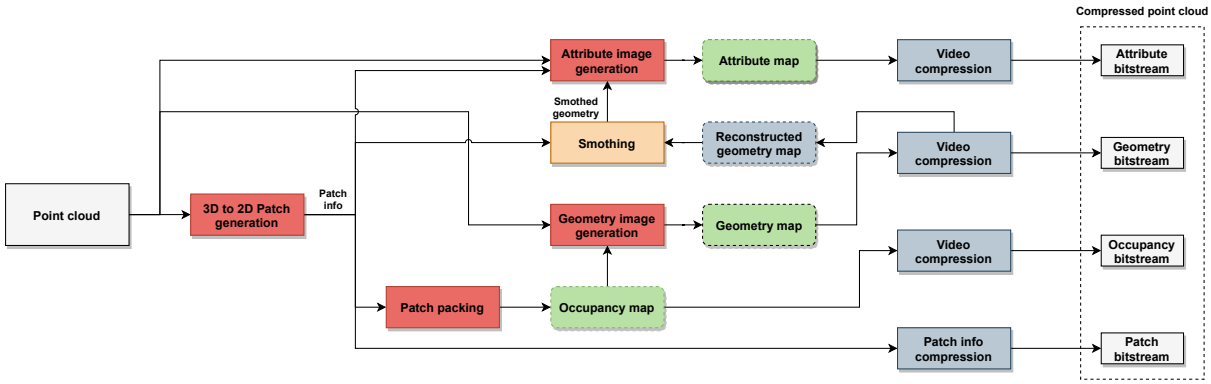


Figure 2.7: V-PCC encoder block diagram.

In addition to the methods already described, image compression techniques such as Joint Photographic Experts Group (JPEG) [68] and Portable Graphics Format (PNG) [69] can be used for LiDAR data compression. In [54], a method based on organizing and packing LiDAR data into 2D image arrays and then applying image compression techniques is proposed, reducing development time by using long-established compression techniques and showing good results using Joint Photographic Experts Group - Lossless (JPEG-LS) [70] in terms of compression efficiency. Fig. 2.8 depicts the stages for image-based compression of LiDAR data. The initial step is to transform raw LiDAR data into X, Y, and Z point locations. This is a standard technique that translates the laser beam's raw distance and both elevation and azimuth angle into X, Y, and Z values. The X, Y, and Z values are then quantized and transformed from floating-point to integer value point representation, with each point represented by a 16-bit value. These values are then transformed into 2D image arrays and subsequently compressed using an existing lossless image compression technique such as PNG or JPEG [54]. Nevertheless, image compression methods are designed for image and human recognition and thus there is a demand for better effective prediction approaches for autonomous driving systems [71]. In addition, there is always the drawback of having to transform the LiDAR sensor data into 2D image array.

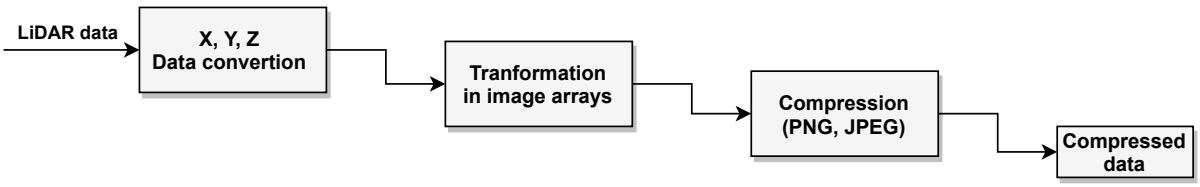


Figure 2.8: Encoding process using image-based compression.

In et al. C. Tu [44], a U-net-based deep learning network is used for real-time compression of streaming point clouds. Using raw data from LiDAR sensor, the point cloud information is stored in a 2D matrix thereby converting the point cloud data to a video-like format. Figure 2.9 [44] depicts the raw data layout combined in a 2D matrix with each row containing a laser ID, and each column representing a laser beam emission, with the distance information represented by the value in each cell [44].

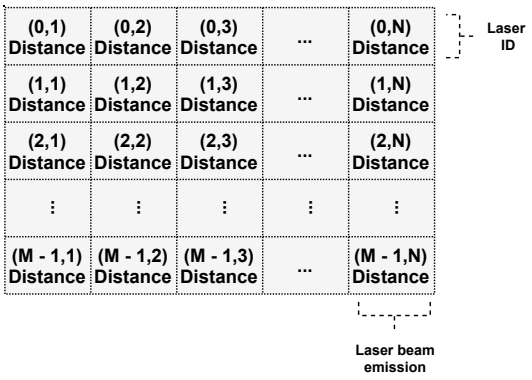


Figure 2.9: Storing technique for point cloud information.

After the conversion/projection in a 2D format, some frames are designated as references, and then using U-net [72] which consists on a convolutional network designed for image segmentation that is both quick and accurate, these are interpolated with the remaining frames. The residual between the predicted and true frames is then computed. Next, an encoder network is used for compression of the point cloud, as depicted in Fig. 2.10. To maintain maximum information on the reference frames, these are losslessly coded whereas the residual frames are quantized first and then coded. By converting raw LiDAR data into a 2D matrix and then using JPEG-LS for data compression spatial redundancy can be reduced. [44].

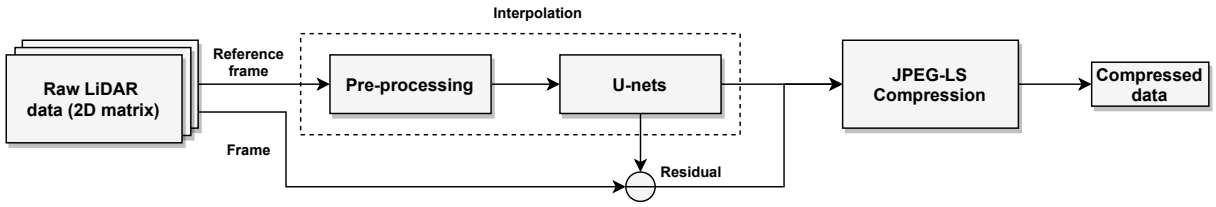


Figure 2.10: Encoding process block diagram.

2.2.4 3D Space Decomposition Techniques

One of the most popular approaches for encoding point cloud data is to employ space partitioning trees [73], with the most common being an octree [74]. Starting from the root, each branch can contain up to eight children, where each branch node represents a specific cube or cuboid from the 3D space. In the octree construction, point coordinates are added iteratively with a breadth-first traversal [75] beginning at the root. In J. Kammerl et al. [51] a differential approach was adopted for streaming applications. By using consecutive frames from data of the octree structures the difference between them was encoded. Applying an Exclusive Disjunction Operation (XOR) to compare the two frames permits to detect the part that changes, Fig. 2.11, and then the resulting vector can be entropy encoded. This is done using an integer-arithmetic coder called range coder [76] which is a modification of an arithmetic coder. By performing a whole compression once for the first point cloud and then just encoding extracted differences within the

point distribution, temporal redundancy can be exploited [51]. The primary advantage of this technique is that it significantly reduces computational and memory resources [56], resulting in good compression ratios. However, as coordinate precision increases, the compression ratio starts decreasing significantly.

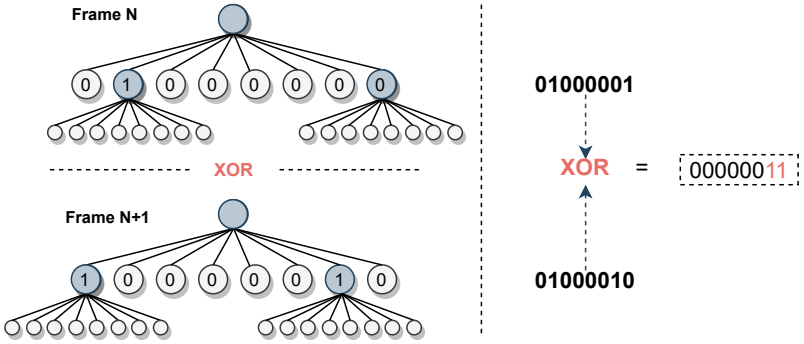


Figure 2.11: Differential technique using XOR operation between two consecutive frames.

The other method chosen in the CfP [66] done by MPEG was the Geometry-based Point Cloud Compression (G-PCC) method. Similar to V-PCC, the G-PCC method uses a lossy scheme having also a lossless support for data compression. G-PCC explores a geometry-based approach covering a wider range of cases, including both sparse and dense point clouds and also dynamic and static objects [56]. This method encodes the 3D space directly in sequence, i.e., it first processes the geometry and then its attributes because the coding of the attributes depends on the decoding of the geometry [52]. Before the encoding itself, a conversion to integer values is made of the coordinates provided by the acquisition system, which are usually expressed in floating-point numbers being this conversion similar to the voxelization of the input point cloud. When it comes to encoding the geometry, there are two modes, one that uses a decomposition of the 3D space into octrees and another that is called "triangle soup" (trisoup) where the surface of the object is approximated by a series of triangles with the resultant being arithmetically encoded. For the encoding of the attributes, one of the three available methods is used, Region Adaptive Hierarchical Transform (RAHT), Predicting Transform, or the Lifting Transform [55]. The RAHT [77] method predicts the values at the subsequent level by using attribute values from a lower octree level. The Predicting Transform implements a hierarchical nearest-neighbor prediction technique based on interpolation, while the Lifting Transform is constructed on top of the Predicting Transform with an extra lifting step [55]. The G-PCC encoder's block diagram is depicted in Fig. 2.12.

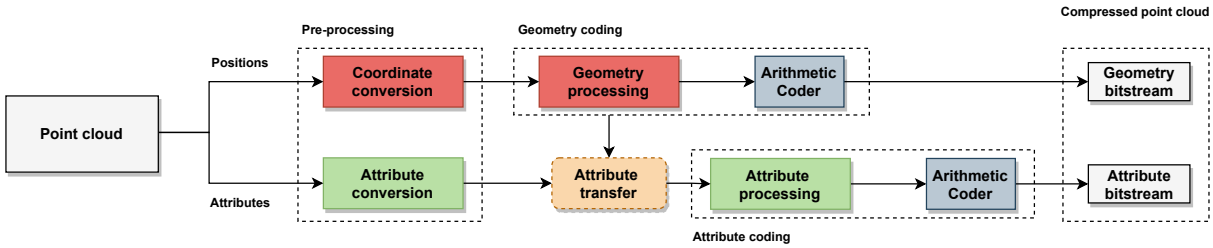


Figure 2.12: G-PCC encoder block diagram.

A DL-based approach has been proposed in [53] for lossless compression of the point cloud geometry called MSVoxelDNN, which is based on earlier work by the same researchers [78]. The proposed solution is centered on voxelized point clouds employing a multiscale architecture that models the voxel occupancy in a coarse-to-fine order. MSVoxelDNN divides the voxels into eight independent groups and uses only one network per group instead of one per voxel predicting multiple voxels in parallel.

Since an octree is formed by recursively dividing the voxel volume into eight sub-cubes until the required precision is achieved, an n level octree can represent a point cloud of size $2^n \cdot 2^n \cdot 2^n$. The suggested technique is shown in Fig. 2.13 [53], where the base resolution is encoded using the VoxelDNN [78] context model and higher resolutions are predicted from the lower resolutions. The probability block is fed through an arithmetic coder, which encodes the voxel occupancies, and the resultant bitstream is concatenated for all bits at all scales [53], allowing a significant reduction in computing complexity. However, despite achieving a significant compression ratio, this method performs poorly on sparse point clouds.

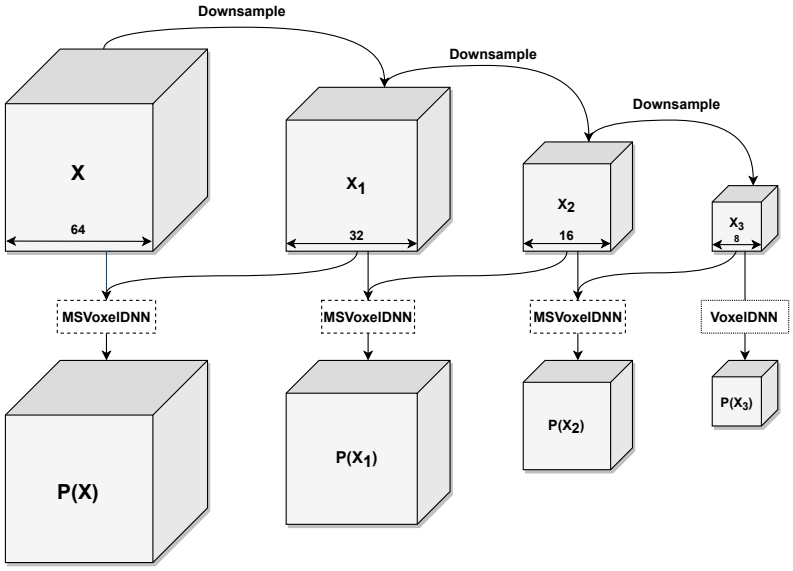


Figure 2.13: Overview of the MSVoxelDNN architecture.

2.2.5 Discussion

Some state-of-the-art algorithms were presented, demonstrating that there are different approaches for streaming point cloud data compression. Table 2.1 summarizes the characteristics of these state-of-the-art algorithms, emphasizing their advantages and disadvantages as well as their performance when doing compression. Analyzing the described algorithms reveals that are approaches based on the LAS format and on low-level compression algorithms that can compress raw LiDAR data. For compression of 3D point cloud data, there are approaches based on spatial decomposition, using spatial partitioning trees and approaches based on 3D to 2D projections and subsequent application of image/video codecs. Also, there are techniques that can apply deep learning in their algorithms as well.

Table 2.1: Analysis of the data compression algorithms.

	Purpose	Compression Type	Real-time	Characteristics	Advantages	Disadvantages	Point Cloud	Compression Ratio [a]	Bit Rate(bpp)[b]
Differential Octree [51] (2012)	Streaming Storage	Lossy	✓	- Uses Point Cloud Library (PCL) - Compares octrees of consecutive point cloud (XOR) - Range coder	- Low complexity - Reduction of computational and memory requirements - Good candidate for parallelization	- Low compression rate at high precision	230k to 270k points 15s 30Hz 7.5M points/s	27:1	3.6 5mm precision
LZMA [47] (2013)	Streaming Storage	Lossless	✓	- Low level compression - Dictionary-based - LZ77	- Implemented in FPGA	- Not applied in LiDARs - Dictionary-based methods not appropried for real-time applications	-	1.86:1	4.6
LAScompression [49] (2013)	Streaming Storage	Lossless	✗	- LAS format	- Implemented in FPGA - Used in LiDAR systems	- Lower compression ratio than LASzip and lower compression rate - Convertion to LAS format - No real-time support	Files represented in LASformat	6:1	-
LASzip [50] (2013)	Streaming Storage	Lossless	✗	- LAS format	- Slightly higher compression ratio than LAScompression - Used in LiDAR systems	- Convertion to LAS format - No real-time support	Files represented in LASformat	6:1	1.3
EDC [48] (2019)	Storage	Lossless	✓	- Low level compression - Delta encoding	- Low complexity - Used in LiDAR systems (robotics)	- Results obtained in simulation	-	1.76:1	4.96
JPEG-LS [54] (2019)	Storage	Lossless	✓	- Image-based - 2D image array - LOCO-I algorithm	- Low complexity - Used in LiDAR systems - Slightly higher compression rate (vs PNG)	- Convertion point cloud to a 2D image array	HDL-32E sensor 59k points p/scan 10Hz	29:1	2.1 bytes pp
PNG [54] (2019)	Storage	Lossless	✓	- Image-based - 2D image array - DEFLATE - LZ77 and Huffman coder	- Low complexity - Used in LiDAR systems - Slightly higher compression ratio (vs JPEG-LS)	- Convertion point cloud to a 2D image array	HDL-32E sensor 59k points p/scan 10Hz	32:1	1.9 bytes pp
U-net [44] (2019)	Streaming	Lossy	✓	- U-net - Deep learnig - Store 3D point cloud in 2D matrix - JPEG-LS	- Reduce temporal redundancy - Pre-processing to alleviate noise	- ML complexity - Training dataset	HDL-64ES2 133k points p/scan 10Hz	-	2.5
G-PCC [55] (2020)	Streaming Storage	Lossless Lossy	✓	- Geometry-based - Encodes directly in 3D space - Octrees - Arithmetic coder	- For sparce and dense point-clouds - For static and dynamic objects - Encoding rates	- No temporal prediction - Limited attribute prediction - Lower encoding rate than V-PCC	1.1M points (static)	10:1 to 35:1	2
V-PCC [55] (2020)	Streaming	Lossless Lossy	✓	- Video-based - 3D to 2D projection - HEVC	- Encoding rates - Video codecs - Rate-distortion performance - Optimized for dense dynamic point clouds	- High complexity - No specific decoding process - Not suitable for large scale sparse point clouds	1.5M points/frame 30 fps 300 frames ∼4.7G	114:1	1.25
MSVoxelDNN [53] (2021)	Streaming	Lossless	✓	- Deep learning - Voxel prediction	- Reduces the bitrate compared to G-PCC - Removes some dependencies between voxels in the same point cloud (parallel).	- ML complexity - Training dataset - Performance on sparse point clouds	1.5M points	27:1	-

[a] Approximated values.
[b] Bits per point.

The implementation proposed in [51] delivers good real-time results with respectable compression ratios but has the disadvantage of being a lossy technique, losing information during compression. Nevertheless it is a well mature method with low complexity and with potential to be explored in an FPGA-based approach. The LAS-based methods [49, 50] do not provide real-time compression and are therefore not the most suitable for streaming point cloud data compression in a real-time application such as autonomous driving. Furthermore, in order for compression to occur, the LiDAR sensor data must be converted into LAS format, which incurs additional overhead. The LZMA method proposed in [61] has a significant compression ratio however, it is a dictionary-based algorithm, which limits its efficiency in real-time applications. The proposed EDC technique in [48] might be used to compress LiDAR raw data, but because the results presented were achieved in a simulation, additional work would be required to evaluate them in a real scenario. Image-based compression methods rely on long-established methodologies and produce strong results, but since they must transform the point cloud to 2D images before compressing it, they are better suited for storage compression rather than streaming point cloud data compression.

The methods proposed by MPEG are relatively new and very promising. With the G-PCC method [52] it is possible to cover a wide range of point clouds, from sparse to dense, presenting itself as the best all-around method as can be seen in Table 2.1, being able to apply both lossy or lossless schemes. V-PCC is more oriented towards streaming point cloud compression and as mentioned in [46] is optimized for dense dynamic point clouds taking advantage of efficient video technologies. These are two recent algorithms from MPEG that show good results and are considered by many authors to be state-of-the-art. Finally, DL-based methods such as those proposed in [44] and [53], provide promising results in terms of

compression ratios and rates, but they are complex and are not yet mature enough to be able to surpass and cover the various use cases that traditional methods can achieve. Additionally, training the dataset is required for the algorithm to work and that is beyond the scope of this work.

In conclusion, some algorithms presented for streaming point cloud data compression will be explored starting with G-PCC and V-PCC, both algorithms proposed by MPEG. These are the most promising algorithms showing a wide range of use cases and high compression ratios. As result, they will be the subjects of this work with a software approach initially, followed by an FPGA-based approach.

References

[1] A. Ansmann, M. Riebesell, and C. Weitkamp, "Measurement of atmospheric aerosol extinction profiles with a raman lidar," *Optics Letters*, vol. 15, no. 13, p. 746, 1990.

[2] W. Krabill, J. Collins, L. Link, R. Swift, and M. Butler, "Airborne laser topographic mapping results," *Photogrammetric Engineering and remote sensing*, vol. 50, no. 6, pp. 685–694, 1984.

[3] R. Cowen, "Clementine begins moon-mapping mission," *Science News*, vol. 145, no. 11, p. 167, 1994.

[4] D. J. Yeong, G. Velasco-Hernandez, J. Barry, and J. Walsh, "Sensor and sensor fusion technology in autonomous vehicles: A review," *Sensors*, vol. 21, no. 6, p. 2140, 2021.

[5] S. Royo and M. Ballesta-Garcia, "An overview of lidar imaging systems for autonomous vehicles," *Applied Sciences*, vol. 9, no. 19, p. 5–5, 2019.

[6] M. E. Warren, "Automotive lidar technology," *2019 Symposium on VLSI Circuits*, p. 1–1, 2019.

[7] S. Thrun, M. Montemerlo, H. Dahlkamp, D. Stavens, A. Aron, J. Diebel, P. Fong, J. Gale, M. Halpenny, G. Hoffmann, *et al.*, "Stanley: The robot that won the darpa grand challenge," *Journal of field Robotics*, vol. 23, no. 9, pp. 661–692, 2006.

[8] C. I. Rablau, "Lidar: a new self-driving vehicle for introducing optics to broader engineering and non-engineering audiences," *Fifteenth Conference on Education and Training in Optics and Photonics: ETOP 2019*, p. 1–1, 2019.

[9] T. Raj, F. H. Hashim, A. B. Huddin, M. F. Ibrahim, and A. Hussain, "A survey on lidar scanning mechanisms," *Electronics*, vol. 9, no. 5, p. 1–1, 2020.

[10] R. Roriz, A. Campos, S. Pinto, and T. Gomes, "Dior: A hardware-assisted weather denoising solution for lidar point clouds," *IEEE Sensors Journal*, p. 1–2, 2021.

[11] J. Guerrero-Ibáñez, S. Zeadally, and J. Contreras-Castillo, "Sensor technologies for intelligent transportation systems," *Sensors*, vol. 18, no. 4, p. 1212, 2018.

[12] SAE International, "Taxonomy and definitions for terms related to driving automation systems for on-road motor vehicles." `https://www.sae.org/standards/content/j3016_202104/`.

[13] R. Roriz, J. Cabral, and T. Gomes, "Automotive lidar technology: A survey," *IEEE Transactions on Intelligent Transportation Systems*, p. 1–16, 2021.

[14] B. Behroozpour, P. A. M. Sandborn, M. C. Wu, and B. E. Boser, "Lidar system architectures and circuits," *IEEE Communications Magazine*, vol. 55, no. 10, p. 135–142, 2017.

[15] M. Khader and S. Cherian, "An introduction to automotive lidar," *Texas Instruments*, 2018.

[16] M. Koskinen, J. T. Kostamovaara, and R. A. Myllylae, "Comparison of continuous-wave and pulsed time-of-flight laser range-finding techniques," *SPIE Proceedings Optics, Illumination, and Image Sensing for Machine Vision VI*, 1992.

[17] O. Steinvall and T. Chevalier, "Range accuracy and resolution for laser radars," *Electro-Optical Remote Sensing SPIE Proceedings*, 2005.

[18] I. Maksymova, C. Steger, and N. Druml, "Review of lidar sensor data acquisition and compression for automotive applications," *Proceedings*, vol. 2, no. 13, p. 852, 2018.

[19] J. Lambert, A. Carballo, A. M. Cano, P. Narksri, D. Wong, E. Takeuchi, and K. Takeda, "Performance analysis of 10 models of 3d lidars for automated driving," *IEEE Access*, vol. 8, p. 131699–131722, 2020.

[20] H. W. Yoo, N. Druml, D. Brunner, C. Schwarzl, T. Thurner, M. Hennecke, and G. Schitter, "Mems-based lidar for autonomous driving," *e & i Elektrotechnik und Informationstechnik*, vol. 135, no. 6, p. 408–415, 2018.

[21] C.-P. Hsu, B. Li, B. Solano-Rivas, A. R. Gohil, P. H. Chan, A. D. Moore, and V. Donzella, "A review and perspective on optical phased array for automotive lidar," *IEEE Journal of Selected Topics in Quantum Electronics*, vol. 27, no. 1, p. 1–16, 2021.

[22] C. H. Jang, C. S. Kim, K. C. Jo, and M. Sunwoo, "Design factor optimization of 3d flash lidar sensor based on geometrical model for automated vehicle and advanced driver assistance system applications," *International Journal of Automotive Technology*, vol. 18, no. 1, p. 147–156, 2016.

[23] M. J. Heck, "Highly integrated optical phased arrays: photonic integrated circuits for optical beam shaping and beam steering," *Nanophotonics*, vol. 6, no. 1, p. 93–107, 2017.

[24] T. Ogawa, H. Sakai, Y. Suzuki, K. Takagi, and K. Morikawa, "Pedestrian detection and tracking using in-vehicle lidar for automotive application," *2011 IEEE Intelligent Vehicles Symposium (IV)*, 2011.

[25] C. Premebida and U. Nunes, "Fusing lidar, camera and semantic information: A context-based approach for pedestrian detection," *The International Journal of Robotics Research*, vol. 32, no. 3, p. 371–384, 2013.

[26] T. Wu, J. Hu, L. Ye, and K. Ding, "A pedestrian detection algorithm based on score fusion for multi-lidar systems," *Sensors*, vol. 21, no. 4, p. 1159, 2021.

[27] V. Magnier, D. Gruyer, and J. Godelle, "Automotive lidar objects detection and classification algorithm using the belief theory," *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[28] H. Cho, Y.-W. Seo, B. V. Kumar, and R. R. Rajkumar, "A multi-sensor fusion system for moving

object detection and tracking in urban driving environments," *2014 IEEE International Conference on Robotics and Automation (ICRA)*, 2014.

[29] X. Zhao, P. Sun, Z. Xu, H. Min, and H. Yu, "Fusion of 3d lidar and camera data for object detection in autonomous vehicle applications," *IEEE Sensors Journal*, vol. 20, no. 9, p. 4901–4913, 2020.

[30] E. Marti, M. A. D. Miguel, F. Garcia, and J. Perez, "A review of sensor technologies for perception in automated driving," *IEEE Intelligent Transportation Systems Magazine*, vol. 11, no. 4, p. 94–108, 2019.

[31] Y. Zhang, J. Wang, X. Wang, and J. M. Dolan, "Road-segmentation-based curb detection method for self-driving via a 3d-lidar sensor," *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 12, p. 3981–3991, 2018.

[32] H. Lee, S. Kim, S. Park, Y. Jeong, H. Lee, and K. Yi, "Avm / lidar sensor based lane marking detection method for automated driving on complex urban roads," *2017 IEEE Intelligent Vehicles Symposium (IV)*, 2017.

[33] X. Peng and J. Shan, "Detection and tracking of pedestrians using doppler lidar," *Remote Sensing*, vol. 13, no. 15, p. 2952, 2021.

[34] E. Arnold, O. Y. Al-Jarrah, M. Dianati, S. Fallah, D. Oxtoby, and A. Mouzakitis, "A survey on 3d object detection methods for autonomous driving applications," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 10, p. 3782–3795, 2019.

[35] P. Sun, X. Zhao, Z. Xu, R. Wang, and H. Min, "A 3d lidar data-based dedicated road boundary detection algorithm for autonomous vehicles," *IEEE Access*, vol. 7, p. 29623–29638, 2019.

[36] M. Byeon and S. W. Yoon, "Analysis of automotive lidar sensor model considering scattering effects in regional rain environments," *IEEE Access*, vol. 8, p. 102669–102679, 2020.

[37] Y. Li, P. Duthon, M. Colomb, and J. Ibanez-Guzman, "What happens for a tof lidar in fog?," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 11, p. 6670–6681, 2021.

[38] N. Charron, S. Phillips, and S. L. Waslander, "De-noising of lidar point clouds corrupted by snowfall," *2018 15th Conference on Computer and Robot Vision (CRV)*, 2018.

[39] P. M. Chu, S. Cho, J. Park, S. Fong, and K. Cho, "Enhanced ground segmentation method for lidar point clouds in human-centric autonomous robot systems," *Human-centric Computing and Information Sciences*, vol. 9, no. 1, 2019.

[40] J. Cheng, D. He, and C. Lee, "A simple ground segmentation method for lidar 3d point clouds," *2020 2nd International Conference on Advances in Computer Technology, Information Science and Communications (CTISC)*, p. 1–1, 2020.

[41] Velodyne, "USER'S MANUAL AND PROGRAMMING GUIDE, HDL-64E S3." `https://gpsolution.oss-cn-beijing.aliyuncs.com/manual/LiDAR/MANUAL%2CUSERS%`

2CHDL-64E_S3.pdf.

[42] C. Tu, E. Takeuchi, A. Carballo, C. Miyajima, and K. Takeda, "Motion analysis and performance improved method for 3d lidar sensor data compression," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 1, p. 243–256, 2021.

[43] M. M. Abdelwahab, W. S. El-Deeb, and A. A. A. Youssif, "Lidar data compression challenges and difficulties," *2019 5th International Conference on Frontiers of Signal Processing (ICFSP)*, p. 1–1, 2019.

[44] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, "Real-time streaming point cloud compression for 3d lidar sensor using u-net," *IEEE Access*, vol. 7, p. 113616–113625, 2019.

[45] A. Biasizzo and F. Novak, "Hardware accelerated compression of lidar data using fpga devices," *Sensors*, vol. 13, no. 5, p. 6405–6422, 2013.

[46] C. Cao, M. Preda, and T. Zaharia, "3d point cloud compression," *The 24th International Conference on 3D Web Technology*, p. 3–3, 2019.

[47] "LZMA SDK." `https://www.7-zip.org/sdk.html`.

[48] I. Maksymova, C. Steger, and N. Druml, "Extended delta compression algorithm for scanning lidar raw data handling," in *International Conference on Intelligent Robots and Systems*, 2019.

[49] D. Mongus and B. Žalik, "Efficient method for lossless lidar data compression," *International Journal of Remote Sensing*, vol. 32, no. 9, p. 2507–2518, 2011.

[50] M. Isenburg, "Laszip: lossless compression of lidar data," *Photogrammetric engineering and remote sensing*, vol. 79, no. 2, pp. 209–217, 2013.

[51] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," *2012 IEEE International Conference on Robotics and Automation*, 2012.

[52] "G-pcc codec description v12," Standard ISO/IEC JTC 1/SC 29/WG 7 N0151, International Organization for Standardization, Virtual, July 2021.

[53] D. T. Nguyen, M. Quach, G. Valenzise, and P. Duhamel, "Multiscale deep context modeling for lossless point cloud geometry compression," *2021 IEEE International Conference on Multimedia & Expo Workshops (ICMEW)*, 2021.

[54] P. V. Beek, "Image-based compression of lidar sensor data," *Electronic Imaging*, vol. 2019, no. 15, 2019.

[55] D. Graziosi, O. Nakagami, S. Kuma, A. Zaghetto, T. Suzuki, and A. Tabatabai, "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*, vol. 9, p. 1–1, 2020.

[56] C. Cao, M. Preda, V. Zakharchenko, E. S. Jang, and T. Zaharia, "Compression of sparse and dense dynamic point clouds—methods and standards," *Proceedings of the IEEE*, vol. 109, no. 9, p. 13–13,

2021.

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

[58] M. Ezhilarasan, P. Thambidurai, K. Praveena, S. Srinivasan, and N. Sumathi, "A new entropy encoding technique for multimedia data compression," *International Conference on Computational Intelligence and Multimedia Applications (ICCIMA 2007)*, 2007.

[59] U. Jayasankar, V. Thirumal, and D. Ponnurangam, "A survey on data compression techniques: From the perspective of data quality, coding schemes, data type and applications," *Journal of King Saud University - Computer and Information Sciences*, vol. 33, no. 2, p. 119–140, 2021.

[60] N. Samteladze and K. Christensen, "Delta: Delta encoding for less traffic for apps," *37th Annual IEEE Conference on Local Computer Networks*, 2012.

[61] X. Zhao and B. Li, "Implementation of the lzma compression algorithm on fpga," *2017 International Conference on Electron Devices and Solid-State Circuits (EDSSC)*, 2017.

[62] J. Ziv and A. Lempel, "A universal algorithm for sequential data compression," *IEEE Transactions on Information Theory*, vol. 23, no. 3, p. 337–343, 1977.

[63] T. Suel, "Delta compression techniques," *Encyclopedia of Big Data Technologies*, p. 658–665, 2019.

[64] ASPRS, "Las specification version 1.4 – r13 15 july 2013." `https://www.asprs.org/wp-content/uploads/2010/12/LAS_1_4_r13.pdf`.

[65] G. G. Langdon, "An introduction to arithmetic coding," *IBM Journal of Research and Development*, vol. 28, no. 2, p. 135–149, 1984.

[66] "Call for proposals for point cloud compression v2," Standard ISO/IEC JTC1/SC29/WG11 MPEG2017/N16763, International Organization for Standardization, Geneva, CH, April 2016.

[67] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (hevc) standard," *IEEE Transactions on circuits and systems for video technology*, vol. 22, no. 12, pp. 1649–1668, 2012.

[68] G. Wallace, "The jpeg still picture compression standard," *IEEE Transactions on Consumer Electronics*, vol. 38, no. 1, pp. xviii–xxxiv, 1992.

[69] T. Boutell, "Png (portable network graphics) specification version 1.0," 1997.

[70] M. Weinberger, G. Seroussi, and G. Sapiro, "The loco-i lossless image compression algorithm: principles and standardization into jpeg-ls," *IEEE Transactions on Image Processing*, vol. 9, no. 8, p. 1309–1324, 2000.

[71] C. Tu, E. Takeuchi, C. Miyajima, and K. Takeda, "Compressing continuous point cloud data using image compression methods," *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, 2016.

[72] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," in *International Conference on Medical image computing and computer-assisted intervention*, pp. 234–241, Springer, 2015.

[73] Y. Feng, S. Liu, and Y. Zhu, "Real-time spatio-temporal lidar point cloud compression," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, p. 1–1, 2020.

[74] D. Meagher, "Geometric modeling using octree encoding," *Computer Graphics and Image Processing*, vol. 19, no. 1, p. 85, 1982.

[75] A. Bundy and L. Wallen, "Breadth-first search," *Catalogue of Artificial Intelligence Tools*, p. 13–13, 1984.

[76] G. N. N. MARTIN, "Range encoder range encoding : An algorithm for removing redundancy from a digitized message," *Video & Data Recording Conference, Southampton. UK, July 2427*, vol. 43, 1979.

[77] R. L. D. Queiroz and P. A. Chou, "Transform coding for point clouds using a gaussian process model," *IEEE Transactions on Image Processing*, vol. 26, no. 7, p. 3507–3517, 2017.

[78] D. T. Nguyen, M. Quach, G. Valenzise, and P. Duhamel, "Learning-based lossless compression of 3d point cloud geometry," *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2021.