IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received May 29, 2020, accepted June 14, 2020, date of publication June 19, 2020, date of current version June 30, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3003753

Prioritized Transmission Control of Point Cloud Data Obtained by LIDAR Devices

KEIICHIRO SATO[1], RYOICHI SHINKUMA[1], (Senior Member, IEEE), TAKEHIRO SATO[1], (Member, IEEE), EIJI OKI[1], (Fellow, IEEE), TAKANORI IWAI[2], DAI KANETOMO[2], AND KOZO SATODA[2], (Member, IEEE)

[1]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[2]System Platform Research Laboratories, NEC Corporation, Kawasaki 211-8666, Japan

Corresponding author: Ryoichi Shinkuma (shinkuma@i.kyoto-u.ac.jp)

This work was supported in part by the Japan Science and Technology Agency as PRESTO under Grant JPMJPR1854, and in part by the Japan Society for the Promotion of Science KAKENHI under Grant JP17H01732.

ABSTRACT Smart monitoring, particularly at intersections, is a promising service that is being considered for the concept of smart cities. A network of light detection and ranging (LIDAR) sensors, which generates point cloud data in real time, can be used to detect people's mobility in smart monitoring. Due to the sheer volume of point cloud data, data transmission requires a significant amount of communication resources. In order to monitor people's mobility in real time, it is necessary to reduce the amount of transmission data to shorten delay. Point cloud compression is one method for reducing the amount of data. However, prior works addressing point cloud compression mainly focused on accuracy for the compression of an entire point cloud without considering its spatial characteristics. The more dynamically a spatial region changes, the more important it is when detecting moving objects such as cars, trucks, pedestrians, and bikes in smart monitoring. This paper proposes a prioritized transmission scheme that applies multiple point cloud compression methods to point cloud data according to the spatial importance of the data, i.e., how dynamically spatial regions change. This paper assumes data transmission of point cloud data from multiple LIDAR devices to an edge server and addresses the intra-frame geometry compression of point cloud data. The proposed scheme splits the point cloud into multiple classes according to the spatial importance and applies multiple point cloud compression methods to each class. A numerical study using a real point cloud dataset obtained at an intersection demonstrates the dependencies of quality, volume, and processing time on possible compression format options. The results verify that the proposed scheme reduces the amount of point cloud data drastically while satisfying the quality and processing time requirements.

INDEX TERMS Point cloud, prioritized transmission, compression, smart monitoring.

I. INTRODUCTION

Smart monitoring, particularly at intersections, is a promising service being considered for the concept of smart cities [1]. With the steep increase in the number of cars in recent years, there is an increasing need for efficient traffic management to avoid traffic jams and optimize traffic flow, especially at intersections. As Datondji *et al.* suggested [2], intersection safety is a critical worldwide issue. In fact, accidents that occur at intersections are one of the major causes of road fatalities [3]. Intersections are particularly dangerous compared to highways as the potential for conflict is much higher at an intersection due to its design. About 30% to 60% (depending on country) of all injuries and about 16% to 36% of fatalities occur at intersections [4]. In addition, accidents at intersections are amongst the most complex since they involve different types of road users, various orientations, speeds, etc. According to the European Road Safety Observatory, more than 62,000 people were killed in traffic accidents at intersections between 1997 and 2006 [5]. More than 80% of accidents at intersections are caused by driver errors [3], [6]. In order to halve the number of road deaths in the near future, it is necessary to develop and implement innovative vehicle monitoring systems, especially at intersections.

Light detection and ranging (LIDAR) is a type of sensor that generates point cloud data in real time. LIDAR sensor

The associate editor coordinating the review of this manuscript and approving it for publication was Hongbin Chen.

networks can be used for smart monitoring at intersections in smart cities. A LIDAR sensor continually fires beams of laser light and measures how long it takes for the light to return to the sensor. The sensor includes special characteristics such as continuous 360-degree visibility and highly accurate depth information [7] which are used to observe the world. Since the sensor cannot acquire information from the back side of an object, more information can be acquired by using multiple sensors at different positions. The point cloud data is collected by multiple sensors and uploaded to an edge server that integrates sensor data and monitors dynamic objects on the road such as vehicles and pedestrians. Then the edge server provides user devices with 1st-level knowledge about the objects. The cloud server receives the aggregated sensor data and provides devices 2nd-level knowledge such as predictive and indicative information about people's mobility, such as road traffic volume and accidents, using 3D object recognition technology [8], [9]. In 2006, the University of Minnesota's ITS group developed a test bed system in which a network of radars and LIDAR sensors were placed near a rural intersection [10]. The network enabled the constant monitoring of an intersection and collection of traffic data from various aspects. In 2009, Zhao *et al.* proposed a system for sensing an intersection using a network of LIDAR sensors and video cameras [11].

Due to the volume of point cloud data, transmitting the data occupies a significant amount of communication resources. For raw dynamic point cloud frames, the data transmission rate for an application with 30 fps can be as high as 6 Gbps [12]. Since network capacity is strictly limited, a communication request for a large amount of point cloud data can cause transmission delays. In order to monitor people's mobility in real time, it is necessary to reduce the amount of transmission data to shorten delay.

Point cloud compression is one method for reducing the volume of transmission data. Most studies have focused on the quality of the decompressed point cloud, processing time, and compression ratio. Setting up a regular structure for the point cloud, such as a binary stream format, is an effective technique for compression [13]–[15]. However, prior works addressing the quality of a decompressed point cloud focused mainly on the compression of an entire point cloud without considering its spatial characteristics. The more dynamically a spatial region changes, the more important it is when detecting moving objects such as cars, trucks, pedestrians, and bikes.

This study proposes a prioritized transmission scheme that applies multiple point cloud compression methods to point cloud data according to the spatial importance of the data, i.e., how dynamic the spatial regions are. We assume data transmission of point cloud data from multiple LIDAR devices to an edge server and address the intra-frame geometry compression of point cloud data. The proposed scheme splits a point cloud into multiple classes according to spatial importance and applies multiple point cloud compression methods to each class. A numerical study using a real point cloud dataset obtained at an intersection demonstrates the dependencies of quality, volume, and processing time on possible compression format options. The results verify that the proposed scheme drastically reduces the amount of point cloud data while satisfying the quality and processing time requirements.

The remainder of this paper is as follows. Section II reviews prior studies conducted on point cloud compression. Section III presents the proposed scheme. Section IV presents performance evaluations using a point cloud dataset and point cloud compression methods. Finally, the conclusions are presented in Section V.

II. RELATED WORK

Smart monitoring has been studied from various aspects, e.g., service delivery, system architecture, privacy, security, detection and prediction models, networking, sensing, devices, communication and computing infrastructure, and energy conservation [16]–[19]. This paper addresses the transmission of point could data.

Point cloud compression has been an important research orientation since the increasing capability of 3D data sensing devices such as LIDAR and depth cameras. A new ad-hoc group has been initiated for MPEG Point Cloud Compression, or MPEG PCC [20], [21]. The group is focused on developing point cloud compression standards and has made significant progress in point cloud compression. There are three main types of point cloud compression [22]: geometry, attribute, and dynamic motion-compensated. Geometry compression codes 3D point coordinates in point clouds. Attribute compression reduces redundancy among point cloud attributes. Dynamic motion-compensated compression targets dynamic point cloud sequences.

One of the most effective techniques for point cloud compression is setting up a regular structure for the point cloud. An octree structure provides an efficient representation of the spatial point cloud distribution in a binary stream format [13], [14]. Schnabel *et al.* discussed an octree-based point cloud compression approach combined with a specialized prediction technique for point-sampled geometry and surface approximation [15]. By using exclusive disjunction operation (XOR) on the octree byte stream, a point cloud data stream can be compressed in real-time as the XOR prediction is relatively simple and can be performed quickly [23]–[25]. However, the approach can only be applied to scenarios with limited movement, which is not the case for the envisioned application with moving people. A kd tree is another structure commonly used to represent a point cloud. The tree is constructed recursively in a top-down fashion by picking the coordinate axis with the largest range (span) of point coordinates and splitting the set of points into two equally-sized subsets, subsequently recursing to each of them [26]. Devillers *et al.* adopted the kd-tree approach to recursively subdivide the bounding box of a point cloud [27] and Shao *et al.* devised an improved kd-tree scheme for uniform partitioning without empty blocks [28]. In 2017, Google

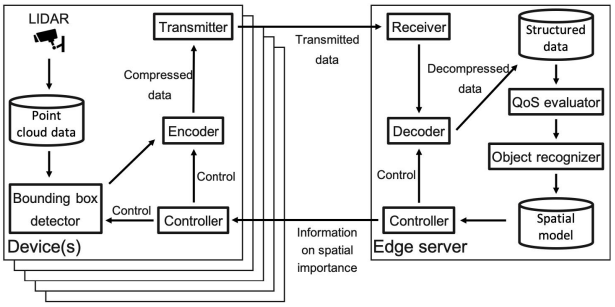


FIGURE 1. System overview of the scheme.

established the open source Draco [29], which uses a kd-tree structure and arithmetic coding to quantify and organize points in 3D space.

Some studies have tried converting 3D point cloud data into a 2D structure rather than decomposing one frame of point cloud data into multiple images. Houshiar and Nüchter used an equirectangular projection to map point clouds onto panorama images [30]. Kohira and Masuda mapped point cloud data onto 2D pixels using GPS time and the parameters of the laser scanner [31].

In recent years, learning-based methods using machine learning, such as recurrent neural networks [32] and autoencoders [33], have been developed. These methods have also been applied to image and video compression. Tu *et al.* proposed a method that uses a recurrent neural network composed of an encoder, a binarizer, and a decoder to progressively compress one frame of information from LIDAR [34]. The 3D information from LIDAR was stored in a 2D matrix as an image. Huang and Liu proposed a deep auto-encoder processing unorder point cloud data with lower reconstruction loss and more accurate detail reconstruction than previously developed unsupervised neural networks [35].

Other compression methods have been inspired by traditional hybrid video coding structures such as H.264/AVC [36] and HEVC [37], which are applied in attribute compression. However, current intra prediction schemes for attribute compression have not been successful [22].

III. PROPOSED SCHEME

A. SYSTEM MODEL

This section describes the system model of our proposed scheme, which is illustrated in Figure 1. This study assumes a system that transmits point cloud data from multiple LIDAR devices to an edge server via wireless networks. As shown in the figure, the system consists of an edge server and multiple devices that use LiDAR sensors to acquire spatial information in real time.

The devices continuously collect point cloud data frame-by-frame using the LIDAR sensors. The controller of each device receives information from the edge server regarding spatial importance and then compresses the point cloud data based on spatial importance. The transmitter then sends the compressed data to the edge server.

The edge server receives and decompresses the compressed data from the devices. The QoS evaluator measures the quality and delay of point cloud transmission. The object recognizer recognizes moving objects such as cars, trucks, pedestrians, and bikes from the structure data and scores the spatial importance of each spatial region on the basis of how dynamically the spatial regions change. The spatial model in the edge server represents the scores of spatial regions. The edge server controller extracts the spatial importance information from the spatial model and sends it to each device.

B. METHODOLOGY

The objective of the proposed scheme is to reduce the volume of point cloud data while satisfying the quality and processing time requirements. The proposed scheme is based on the principle that point cloud data has spatial characteristics; some spatial regions change dynamically, while others rarely change. We hypothesize that the former is important for detecting moving objects, such as pedestrians and cars, in smart monitoring. Therefore, spatial regions in point cloud data that change more dynamically are considered to be of higher importance when the data is transmitted and classified on the basis of the importance.

The key idea of the proposed scheme is to adaptively assign compression formats to each importance class of the spatial regions. In general, a higher compression ratio leads to a lower quality point cloud, so a compression format with a lower compression ratio is assigned to point cloud data of higher importance. For simplicity, we consider two classes in the following part: high and low importance. Therefore, compression formats with low and high compression ratios are used to transmit the high and low importance classes of point cloud data, respectively.

The algorithm of the proposed scheme is shown with the pseudo code format in Algorithm 1.

C. PROBLEM FORMULATION

The problem formulation minimizes the total volume of point cloud data obtained by LIDAR devices under the requirements for the decompressed point cloud quality and processing time (including encoding time), and is given as:

$$\min_{x_h, x_l} \; v(d_h, x_h) + v(d_l, x_l) \tag{1}$$

$$\text{s.t. } (\bar{d}_h(x_h), \bar{d}_l(x_l)) \geq \alpha \tag{2}$$

$$t(d_h, x_h) + t(d_l, x_l) \leq \Delta. \tag{3}$$

We first explain the given parameters in the problem formulation. d_h and d_l are raw point cloud data of high importance and low importance, respectively. α and Δ denote the quality requirement and the processing deadline, respectively.

We then explain the decision variables in the problem formulation. x denotes a compression format, which is composed of a compression method and the parameter(s). x_h and x_l indicate the compression formats used for high and low importance data, respectively. $\bar{d}(x)$ signifies the point cloud

Algorithm 1 Algorithm of Proposed Scheme

Modeling phase
$f_1, f_2, \ldots, f_k \leftarrow k$ frames of point cloud obtained by LIDARs in advance
$I \leftarrow$ spatial importance extracted from $f_1, f_2, \ldots, f_k$
Compressing phase
Send I to each device
$R_h \leftarrow$ a set of high-importance spatial regions determined by I
$R_l \leftarrow$ a set of low-importance spatial regions determined by I
$p_1, p_2, \ldots, p_n \leftarrow$ point cloud obtained by LIDARs in real time, where n is the number of points in the point cloud
for $i = 1, 2, \ldots, n$ **do**
    if p_i in R_h **then**
        Add p_i to the high-importance data from point cloud, d_h
    else if p_i in R_l **then**
        Add p_i to the low-importance data from point cloud, d_l
    end if
end for
Perform compression for d_h and d_l with the compression formats for each
Send the compressed data from devices to the edge server
Receive the compressed data and perform decompression

data obtained from d using x. $v(d, x)$ is the volume of the compressed point cloud data obtained from d using x. $q(\bar{d}(x))$ is the quality of the decompressed point cloud data decoded from $\bar{d}(x)$. $t(d, x)$ is the processing time for the compression of point cloud data d using x.

Eq. (1) describes the objective of the proposed scheme, which is to minimize the volume of the compressed point cloud data. Eq. (2) states that the quality of decompressed point cloud data needs to be higher than α. Eq. (3) states that the total processing time of point cloud compression needs to be shorter than Δ.

D. HOW TO DETERMINE OPTIMAL COMPRESSION FORMANTS

Next, we need to consider how to determine x_h and x_l in (1) to (3) optimally. It is not realistic to estimate $v(d, x)$, $q(\bar{d}(x))$, and $t(d, x)$ before actually performing compression in advance because, in general, point cloud data compression has non-linear characteristics; $v(d, x)$ depends on the redundancy of raw point cloud data; $q(\bar{d}(x))$ is not a linear function of a parameter of compression format x, and $t(d, x)$ depends computational power of the device. Therefore, a straightforward approach is to study the dependencies of $v(d, x)$, $q(\bar{d}(x))$, or $t(d, x)$ on possible options of compression format x. This approach is shown in the next section.

IV. NUMERICAL STUDY

The proposed scheme was evaluated in order to verify its effectiveness. We found that the scheme maintains the quality of the high-importance point cloud data while reducing the volume of compressed point cloud data and processing time.

A. POINT CLOUD DATASET

An experiment was conducted using the Ko-PER intersection dataset published by Strigel *et al.* [38]. The Ko-PER intersection dataset consists of Sequence1a-d, Sequence2, Sequence3, and comprises raw LIDAR data, undistorted camera images, reference data of selected vehicles, and object label information. Labeled objects are cars, trucks, pedestrians, and bikes.

Raw data from the Ko-PER intersection dataset was collected to identify a public four-way intersection in Aschaffenburg, Germany. Its main road features two straight ahead lanes and a separate left-turn lane for each direction. The branch roads have one lane per direction and a left-turn lane on one side. Additionally, the main road has a separate bicycle lane and the intersection is surrounded by sidewalks on all sides except one.

The intersection was observed by fourteen SICK LD-MRS 8-layer research LIDAR from different viewpoints. The sensors were installed on infrastructure components such as lamp posts and traffic lights and were mounted at least 5 m above the ground. Four LIDAR sensors covered the area of the central intersection, two scanners observed the sidewalks along the main road, and eight sensors observed three egresses of the intersection. The LIDAR sensors synchronously operated at a frequency of 12.5 Hz (80 ms).

This study used Sequence1a because Sequence2 and Sequence3 do not contain label information. Sequence1a contained 1211 frames. Each LIDAR sensor was calibrated in advance using calibration information from the Ko-PER intersection dataset and Open3D [39], an open source Python programming library. The number of points in the calibrated point cloud data of one frame was about 15,000.

B. POINT CLOUD COMPRESSION METHODS

This section describes two point cloud compression methods: octree-based compression and Draco. As we presented in Section II, and as Huang and Liu mentioned [35], octree and kd-tree are widely used point cloud representations. Therefore, using octree-based compression and Draco, which is based on kd-tree, was feasible for the proposed scheme.

1) OCTREE-BASED COMPRESSION

Octree-based compression is provided by an octree structure [15]. An octree is a tree data structure suitable for sparse 3D data, in which each branch node represents a certain cube or cuboid bounding volume in space. Starting at the root, each branch has up to eight children, one for each sub-octant of the node's bounding box. An example is shown in the left half of Figure 2. By traversing the tree in breadth-first order and

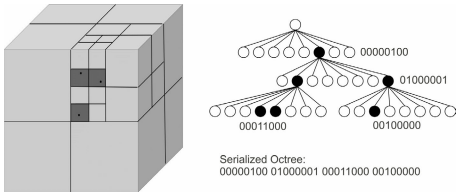


FIGURE 2. Example of octree structure and its serialization [23].

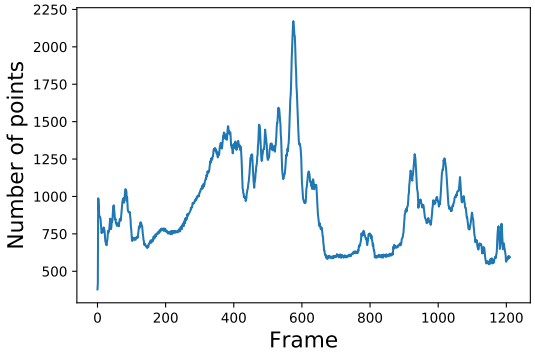
outputting every child node configuration byte encountered, we are able to efficiently encode the point distribution in space. Upon reading the encoded byte stream, the number of bits set in the first byte tells the decoder the number of consecutive bytes that are direct children. The bit positions specify the voxel/child in the octree they occupy. The righthand side of Figure 2 illustrates this byte stream representation for the octree in the left half.

This study used the Point Cloud Library (PCL) to implement octree-based compression. PCL is a large cross-platform open source C++ programming library that implements a large number of point cloud universal algorithms and efficient data structures [40], [41]. PCL provides several parameters for octree-based compression. This study used octree voxel resolution which defines the voxel size of the deployed octree. This parameter was set between 0.0001 and 9.0. Other parameters were set to default.

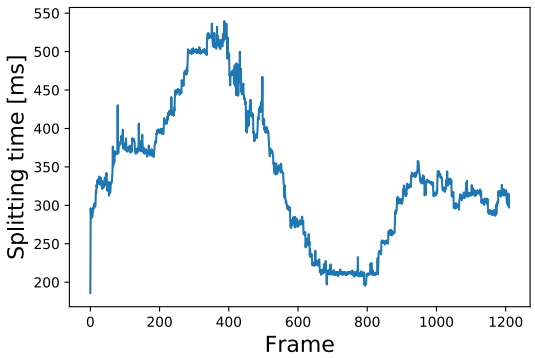
2) DRACO

Draco is a compression algorithm developed by Google [29], which enables the compression of meshes and point cloud data [42]. Due to the differences between the two data representations, Draco does not rely on a single compression algorithm but uses multiple techniques to compress both representations on the basis of compression ratio, decoding speed, and discretization losses. Such compression is much more optimal when compared with general purpose algorithms such as gzip due to this specialization. For point cloud data, Draco mainly relies on order-optimized encoding by rearranging the points using a kd-tree. Positional data is discretized by a configurable number of quantization bits. While this will naturally result in the loss of spatial resolution, it can be fine-tuned for accuracy or visual quality requirements. Draco also supports the compression of arbitrary point attributes, making it well-suited for heterogenous data. To compress mesh topology, Draco relies on the Edgebreaker algorithm, which encodes a mesh in the form of a spiral, encoding the connectivity of each triangular face in a string while keeping track of the already visited vertices and faces. These strings are then compressed separately by the library.

Draco provides two parameters: quantization parameter (qp) and compression level (cl). The qp is the number of quantization bits; a higher qp generally results in a more optimal compression rate. If the qp is set to 0, Draco will not perform any quantization, resulting in lossless compression. In this study, the qp was set between 1 and 20. Zero was not included in order to use PSNR (described later in IV-D) as an evaluation metric. The cl turns different compression features on or off. In general, the highest cl setting, 10, will have the most compression, and 0 will have the least compression. In this study, the cl was set to 10.



(a) Number of points in high-importance class



(b) Splitting time

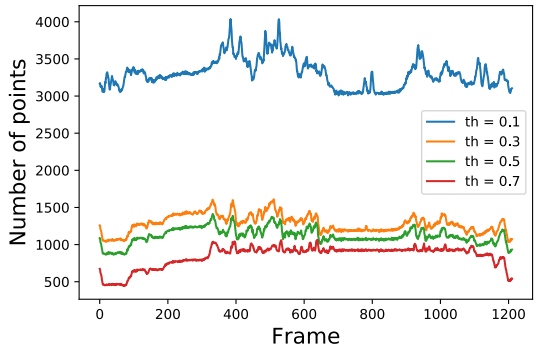
FIGURE 3. Bounding box-based case.

C. TWO CASES OF SPATIAL IMPORTANCE CLASSIFICATION

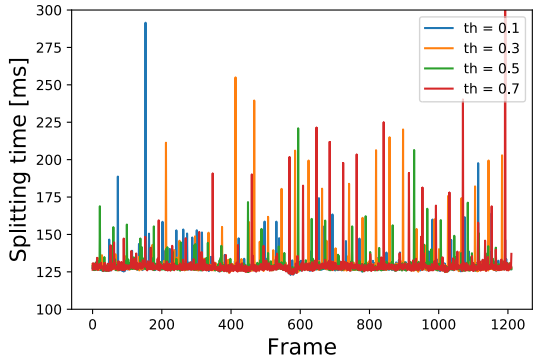
This section considers two cases of spatial importance classification using object label information available from the dataset. Shinkuma and Yamada previously studied a method to estimate the importance of spatial information using machine learning [43]–[45]. The application of the method to the proposed scheme in this paper is left as future work. Note that the computational environment for measuring splitting time below is Ubuntu OS version 18.04.3, which has twelve Intel(R) Core(TM) i7-8700 CPUs @ 3.2GHz and 32GB memory.

1) BOUNDING BOX-BASED CASE

In the bounding box-based case, the bounding box extracted from the object label information is the high-importance region. Ten features of each object were obtained by LIDAR devices in the object label information. This study used seven features, x position [m], y position [m], z position [m] (set to zero), width of the object [m], length of the object [m], height of the object [m], and the orientation angle [rad], to extract the bounding box of the object. Then, the point cloud was split into two, with the inside of the bounding box



(a) Number of points in high-importance class



(b) Splitting time

FIGURE 4. Voxel-based case.

as the high-importance region. Figure 3a shows the number of points in the high-importance region. Figure 3b shows the splitting time, which varied depending on the frame.

2) VOXEL BASED CASE

In the voxel-based case, the high-importance region was the voxel whose importance calculated from the object label information was higher than the threshold. First, the statistical score for each voxel was calculated by checking whether a bounding box belonged to each voxel in all frames and dividing by the number of frames. Then, the score was regarded as the voxel's importance. Finally, the point cloud was split into two. The voxel with a higher importance than the threshold was the high-importance region. This study set one side of the voxel to 1 [m]. Figure 4a shows the number of points in the high-importance region for each threshold. Figure 4b shows the splitting time for each threshold. The number of high-importance points decreased as the threshold increased.

D. PSNR OF POINT CLOUD

This section introduces the peak signal-to-noise ratio (PSNR) of the point cloud as an objective quality metric. PSNR has traditionally been used in analogue systems, digital image/video technology, and as a consistent quality metric [46], [47]. Prior works on point cloud compression used PSNR [48]–[50] for performance evaluation. PSNR is

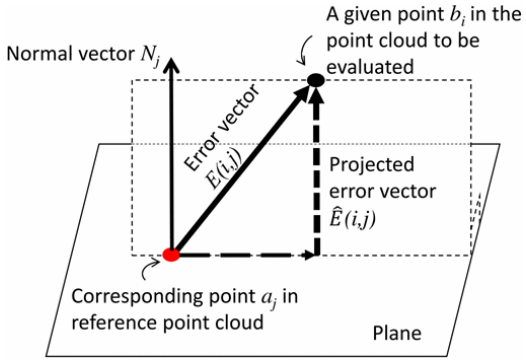


FIGURE 5. Point-to-point vs. point-to-plane [51].

defined by

$$\mathrm{PSNR}(A, B) = 10 \log 10 \frac{\mathrm{MAX}^2}{\mathrm{MSE}} [dB], \tag{4}$$

where A and B represent the original point cloud and compressed point cloud, respectively. On the other hand, MAX and MSE can have various definitions.

MAX is conventionally defined as the diagonal distance of a bounding box or the maximum of the three sides of the bounding box of point cloud A. According to Tian *et al.*, one disadvantage of these definitions is that, given the same amount of error for each point, a spatially larger point cloud would produce a higher PSNR than a spatially smaller point cloud. They proposed the maximum distance of the nearest neighbor points to point cloud A [51], which chooses MAX based on the intrinsic resolution of point cloud A. This study also defines MAX as the maximum distance of nearest neighbor points to point cloud A.

This study adapted two metrics for calculating MSE: point-to-point and point-to-plane. Figure 5 shows the differences between the two metrics.

1) POINT-TO-POINT

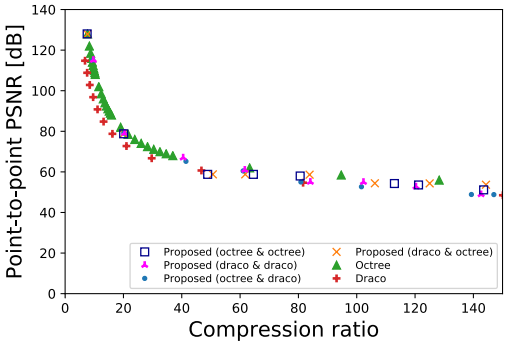
Point-to-point calculates the MSE from the distance between points [52]. The point-to-point MSE is defined by

$$\mathrm{MSE}(A, B) = \frac{1}{N_A} \sum_{i=1}^{N_A} (a_i - b_{nearest\ neighbour})^2, \tag{5}$$

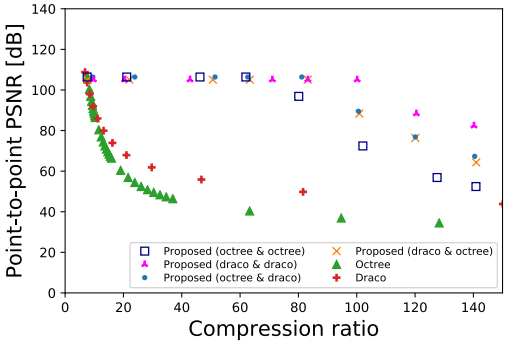
where N_A, a_i, and $b_{nearest\ neighbor}$ are the number of points in point cloud A, the point belonging to point cloud A, and the nearest neighbor point of a_i that belongs to point cloud B, respectively. Finally, let MSE be the maximum value of MSE (A, B) and MSE (B, A). This study used a nearest neighbor search algorithm implemented by PCL. This algorithm is based on the Fast Library for Approximate Nearest Neighbors (FLANN) [53].

2) POINT-TO-PLANE

Point-to-plane calculates the MSE from the distance between the plane and the point [51], [54]. The point-to-plane MSE is

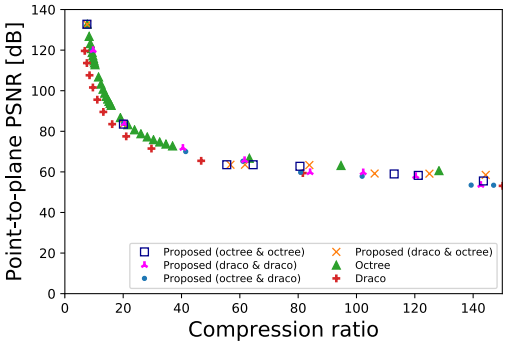


(a) Entire region

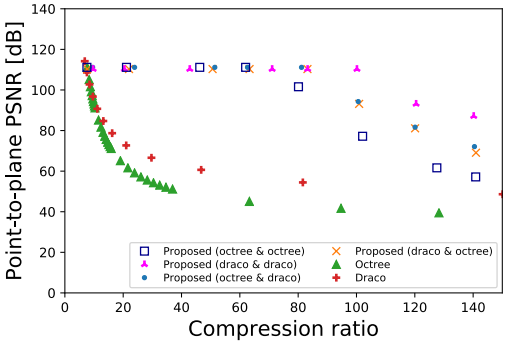


(b) High-importance region

FIGURE 6. Point-to-point PSNR vs. compression ratio in bounding box-based case.



(a) Entire region



(b) High-importance region

FIGURE 7. Point-to-plane PSNR vs. compression ratio in bounding box-based case.

defined by

$$\mathrm{MSE}(A, B) = \frac{1}{N_A} \sum_{i=1}^{N_A} (\vec{e}_i \cdot \vec{n}_i)^2, \tag{6}$$

where $\vec{e}_i$ and $\vec{n}_i$ are the error vector between a_i and $b_{nearest\ neighbour}$ and the normal vector of a_i, respectively. The error from the surfaces of the structures is represented by taking the inner product of $\vec{e}_i$ and $\vec{n}_i$. Compared to point-to-point, point-to-plane captures surface features of structures more effectively. Finally, let MSE be the maximum value of MSE (A, B) and MSE (B, A) as in point-to-point. In this study, the normal vector was calculated by using an algorithm implemented by PCL. This algorithm estimates the normal vector from the nearest neighbor points to the observation point. The number of the nearest neighbor points was set to 10 in this study.

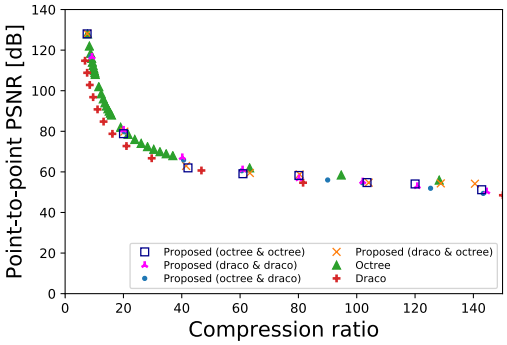
E. RESULTS

1) PSNR

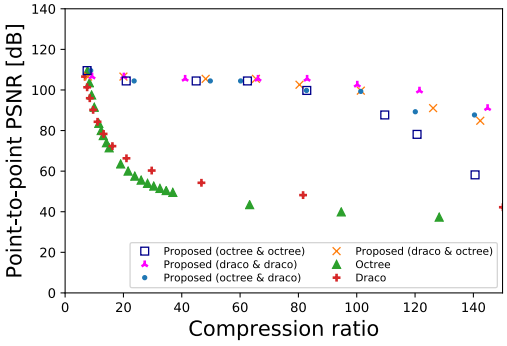
This section discusses the results of the numerical study using PSNR as the quality metric. We first selected the frames containing the label information of 25 objects from Sequence1a and randomly picked up 100 frames to plot the average values of the results. The proposed scheme used octree-based compression, Draco, or a combination of the two, while the benchmark used either one of the two methods. The comparison with the benchmark is meaningful because, as we presented in Section II and as Huang and Liu mentioned [35], octree and kd-tree, on which Draco is based, are widely-used point cloud representations.

In the following part, the proposed scheme (compression method A & compression method B) means that it used compression methods A and B for high and low importance regions, respectively. For improved visibility, we only plotted the results obtained from an optimal combination of the parameters used for compression methods A and B in the proposed scheme.

Figures 6 and 7 present PSNR compared with the compression ratio in the bounding box-based case. Note point-to-point and point-to-plane PSNRs, which were introduced in Section IV-D, are used in Figs. 6 and 7, respectively. PSNR decreased as compression ratio increased, which suggests that as the amount of transmission data decreases, point cloud quality also decreases monotonically. As seen in Figs. 6a and 7a, the benchmark using Draco gives a lower PSNR than octree-based compression for the entire region of point cloud data. The performance of the proposed scheme was comparable to the octree-based compression in terms of PSNR for the entire region. Additionally, we plotted the PSNR for the high-importance regions in Figs. 6b and 7b. The overall values of the point-to-plane PSNR were slightly larger than those of the point-to-point PSNR. This is likely because high-importance regions contain objects that affect point-to-plane measurement as explained in Section IV-D.

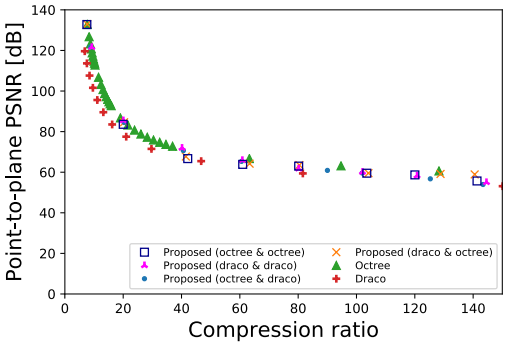


(a) Entire region

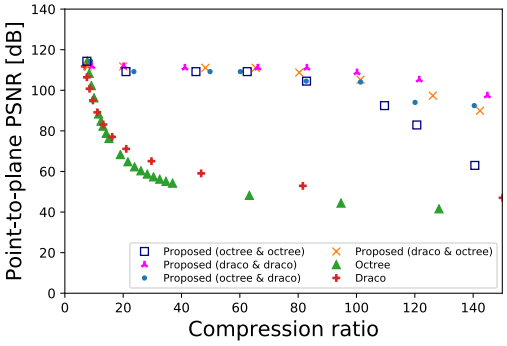


(b) High-importance region

FIGURE 8. Point-to-point PSNR vs. compression ratio in voxel-based case.



(a) Entire region



(b) High-importance region

FIGURE 9. Point-to-plane PSNR vs. compression ratio in voxel-based case.

These figures show that the proposed method performed more effectively than the benchmarks in terms of PSNR for the high-importance region. The proposed scheme successfully reduced the volume of point cloud data while maintaining PSNR; the proposed scheme (draco & draco) achieved 100 dB of PSNR up to a compression ratio of 100.

Figures 8 and 9 present the results in the voxel-based case. The trends seen in these figures are largely the same as those in Figures 6 and 7, which suggests that the proposed scheme was successful in both cases of spatial importance classification. The proposed scheme (draco & draco) reduced the volume of point cloud data with a high compression ratio while maintaining PSNR for the high importance region.

However, a question here is whether or not the proposed scheme (draco & draco) actually assigned low and high compression ratios to high and low importance regions, respectively. Table 1 shows the parameter sets used in the proposed scheme (draco & draco). As explained in Section IV-B, our study used the parameter qp to vary the compression ratio of Draco. In the bounding box-based case, the qp for high and low importance regions was set to 20 to 16 and 20 to 1, respectively, which suggests that a larger qp is used for high importance regions rather than low importance regions. In the voxel-based case, another parameter, *th*, was also adjusted to vary the compression ratio, but the larger qp was used for high importance regions rather than low importance regions, the same as in the bounding box case. The findings in this table demonstrate that the proposed scheme (draco & draco) actually assigned low and high compression ratios to high and low importance regions, respectively.

TABLE 1. Parameter sets used in proposed scheme (draco & draco).

(a) Bounding box-based case

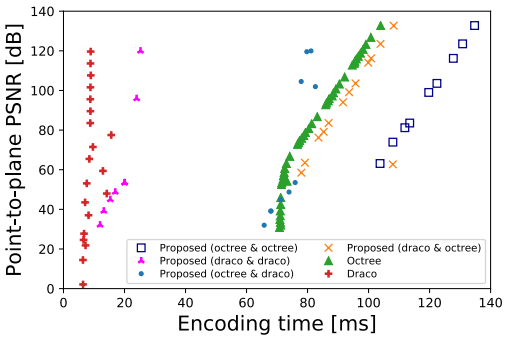
Compression ratio	Point-to-plane PSNR [dB]	qp for high importance	qp for low importance
9.7	110.3	20	20
20.4	110.3	20	14
42.8	110.3	20	11
71.1	110.3	20	9
83.2	110.3	20	8
100.2	110.3	20	3
120.4	93.1	17	6
140.2	87.1	16	1

(b) Voxel-based case

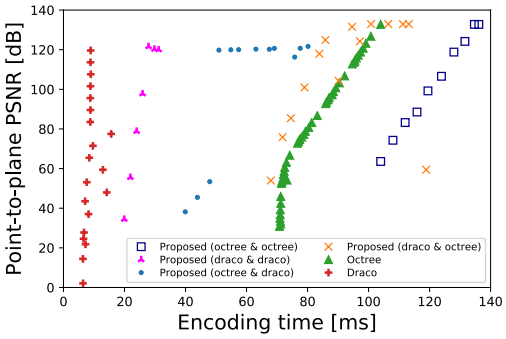
Compression ratio	Point-to-plane PSNR [dB]	qp for high importance	qp for low importance	th
9.2	111.8	20	20	0
20.2	111.8	20	8	0
41.2	111.1	20	11	0.5
66.1	111.1	20	9	0.5
83.0	111.1	20	7	0.5
100.2	108.8	20	7	0.6
121.5	105.2	19	5	0.7
144.9	97.4	20	7	0.9

2) ENCODING TIME

This section presents the encoding time of the proposed scheme and the benchmarks. We used the same computational environment as the one in Section IV-B. As seen in Figure 10, PSNR decreased as encoding time decreased,



(a) Bounding box-based case



(b) Voxel-based case

FIGURE 10. **Point-to-plane PSNR of entire region vs. encoding time.**

suggesting that as the compression processing time decreases, point cloud quality also decreases. An overall comparison revealed that Draco compresses faster than octree-based compression. As shown in Figure 10, in both of the bounding box and voxel-based cases, the encoding time of the proposed scheme (draco & draco) was the shortest and followed the same trend as that of Draco. As we mentioned in Section IV-B, Draco was designed using multiple techniques to optimize compression. Comparing the results of PSNR versus compression ratio and encoding time, the proposed scheme (draco & draco) was most effective in terms of reduction in the volume of point cloud data, the quality of the decompressed point cloud, and the compression encoding time.

F. DISCUSSION ABOUT COMPLEXITY

As we presented in Section III-B, the proposed scheme performs processing when splitting or compressing point could data.

In general, the complexity of encoding in point cloud compression is discussed in terms of time [23], [55], so the complexity can be described using the numerical result of the encoding time, shown in Section IV-E2. As we observed, the proposed scheme (draco & draco) performed more effectively than the proposed scheme with other compression formats and the benchmark methods.

A possible drawback of the proposed scheme is the splitting time for importance classification, which was shown in Figs. 3b and 4b as it could not be ignored unlike the encoding time presented in Section IV-E2. This drawback would become greater as the number of importance classes increases to three or more (this paper only assumed two). Let n and c denote the number of points and the number of importance classes, respectively. Using Algorithm 1, the order of the complexity of splitting is given as $\mathcal{O}(nc)$, which means that the processing time for splitting increases linearly against the increasing number of points obtained from LIDAR devices or the number of importance classes in the proposed scheme. Parallelizing processing is a possible solution for this, which will be investigated in future work.

V. CONCLUSION

In this paper, we proposed a hybrid scheme that applies multiple point cloud compression methods to point cloud data obtained by multiple LIDAR devices. The compression methods were applied according to the spatial importance of the data with a focus on intra-frame geometry compression of the point cloud. Two point cloud compression methods, octree-based compression and Draco, were applied to the high and low importance point clouds split by the proposed scheme. A numerical study using the Ko-PER intersection dataset demonstrated the dependencies of quality, volume, and processing time on possible compression formats. The results verified that the proposed scheme reduces the volume of point cloud data drastically while satisfying the quality and processing time requirements.

REFERENCES

[1] G. Hancke, B. Silva, and G. Hancke, Jr., "The role of advanced sensing in smart cities," *Sensors*, vol. 13, no. 1, pp. 393–425, Dec. 2012. [Online]. Available: https://www.mdpi.com/1424-8220/13/1/393

[2] S. R. E. Datondji, Y. Dupuis, P. Subirats, and P. Vasseur, "A survey of vision-based traffic monitoring of road intersections," *IEEE Trans. Intell. Transp. Syst.*, vol. 17, no. 10, pp. 2681–2698, Oct. 2016.

[3] S. Lefèvre, C. Laugier, and J. Ibañez-Guzmán, "Risk assessment at road intersections: Comparing intention and expectation," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2012, pp. 165–171.

[4] O. Aycard, Q. Baig, S. Bota, F. Nashashibi, S. Nedevschi, C. Pantilie, M. Parent, P. Resende, and T.-D. Vu, "Intersection safety using LiDAR and stereo vision sensors," in *Proc. IEEE Intell. Vehicles Symp. (IV)*, Jun. 2011, pp. 863–869.

[5] G. Yannis, C. Antoniou, and P. Evgenikos, "Comparative analysis of junction safety in Europe," in *Proc. 12th World Conf. Transp. Res. (WCTR)*, 2010, pp. 1–11.

[6] P. Subirats, Y. Dupuis, E. Violette, D. Doucet, and G. Dupre, "A new tool to evaluate safety of crossroad," in *Proc. 4th Int. Symp. Highway Geometric Design. Valence*, 2010, pp. 2–5.

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

[8] D. Maturana and S. Scherer, "VoxNet: A 3D convolutional neural network for real-time object recognition," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 922–928.

[9] B. Li, T. Zhang, and T. Xia, "Vehicle detection from 3D LiDAR using fully convolutional network," 2016, *arXiv:1608.07916*. [Online]. Available: http://arxiv.org/abs/1608.07916

[10] L. Alexander, P.-M. Cheng, A. Gorjestani, A. Menon, B. Newstrom, C. Shankwitz, and M. Donath, "The Minnesota mobile intersection surveillance system," in *Proc. IEEE Intell. Transp. Syst. Conf.*, Sep. 2006, pp. 139–144.

[11] H. Zhao, J. Cui, H. Zha, K. Katabira, X. Shao, and R. Shibasaki, "Sensing an intersection using a network of laser scanners and video cameras," *IEEE Intell. Transp. Syst. Mag.*, vol. 1, no. 2, pp. 31–37, Sep. 2009.

[12] E. d'Eon, B. Harrison, T. Myers, and P. Chou, *8I Voxelized Full Bodies—A Voxelized Point Cloud Dataset*, Standard ISO/IEC JTC1/SC29 Joint WG11/WG1 (MPEG/JPEG) input document WG11M40059/WG1M74006, Geneva, Switzerland, 2017.
[13] J. Peng and C. C. J. Kuo, "Octree-based progressive geometry encoder," *Proc. SPIE*, vol. 5242, pp. 301–311, Nov. 2003.
[14] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," in *Proc. SPBG*, vol. 6, 2006, pp. 103–110.
[15] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. SPBG*, vol. 6, 2006, pp. 111–120.
[16] L. Sanchez, L. Muñoz, J. A. Galache, P. Sotres, J. R. Santana, V. Gutierrez, R. Ramdhany, A. Gluhak, S. Krco, E. Theodoridis, and D. Pfisterer, "SmartSantander: IoT experimentation over a smart city testbed," *Comput. Netw.*, vol. 61, pp. 217–238, Mar. 2014.
[17] X. Liu, A. Liu, T. Wang, K. Ota, M. Dong, Y. Liu, and Z. Cai, "Adaptive data and verified message disjoint security routing for gathering big data in energy harvesting networks," *J. Parallel Distrib. Comput.*, vol. 135, pp. 140–155, Jan. 2020.
[18] M. Peng, W. Liu, T. Wang, and Z. Zeng, "Relay selection joint consecutive packet routing scheme to improve performance for wake-up radio-enabled WSNs," *Wireless Commun. Mobile Comput.*, vol. 2020, pp. 1–32, Jan. 2020.
[19] Y. Chen, W. Liu, T. Wang, Q. Deng, A. Liu, and H. Song, "An adaptive retransmit mechanism for delay differentiated services in industrial WSNs," *EURASIP J. Wireless Commun. Netw.*, vol. 2019, no. 1, p. 258, Dec. 2019.
[20] S. Schwarz, M. Preda, V. Baroncini, M. Budagavi, P. Cesar, P. A. Chou, R. A. Cohen, M. Krivokuca, S. Lasserre, Z. Li, J. Llach, K. Mammou, R. Mekuria, O. Nakagami, E. Siahaan, A. Tabatabai, A. M. Tourapis, and V. Zakharchenko, "Emerging MPEG standards for point cloud compression," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 9, no. 1, pp. 133–148, Mar. 2019.
[21] R. Mekuria and L. Bivolarsky, "Overview of the MPEG activity on point cloud compression," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2016, p. 620.
[22] Y. Shao, Q. Zhang, G. Li, Z. Li, and L. Li, "Hybrid point cloud attribute compression using slice-based layered structure and block-based intra prediction," in *Proc. ACM Multimedia Conf. (MM)*, 2018, pp. 1199–1207.
[23] J. Kammerl, N. Blodow, R. B. Rusu, S. Gedikli, M. Beetz, and E. Steinbach, "Real-time compression of point cloud streams," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2012, pp. 778–785.
[24] R. Mekuria, K. Blom, and P. Cesar, "Design, implementation, and evaluation of a point cloud codec for tele-immersive video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 27, no. 4, pp. 828–842, Apr. 2017.
[25] C. Moreno, Y. Chen, and M. Li, "A dynamic compression technique for streaming kinect-based point cloud data," in *Proc. Int. Conf. Comput., Netw. Commun. (ICNC)*, Jan. 2017, pp. 550–555.
[26] R. Klokov and V. Lempitsky, "Escape from cells: Deep Kd-networks for the recognition of 3D point cloud models," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Oct. 2017, pp. 863–872.
[27] O. Devillers and P.-M. Gandoin, "Geometric compression for interactive transmission," in *Proc. Vis. VIS*, 2000, pp. 319–326.
[28] Y. Shao, Z. Zhang, Z. Li, K. Fan, and G. Li, "Attribute compression of 3D point clouds using Laplacian sparsity optimized graph transform," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, Dec. 2017, pp. 1–4.
[29] Google. (2017). *Draco: 3D Data Compression*. [Online]. Available: https://github.com/google/draco
[30] H. Houshiar and A. Nuchter, "3D point cloud compression using conventional image compression for efficient data transmission," in *Proc. 25th Int. Conf. Inf., Commun. Automat. Technol. (ICAT)*, Oct. 2015, pp. 1–8.
[31] K. Kohira and H. Masuda, "Point-cloud compression for vehicle-based mobile mapping systems using portable network graphics," *ISPRS Ann. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 4, 2017, pp. 1–8.
[32] W. Zaremba, I. Sutskever, and O. Vinyals, "Recurrent neural network regularization," 2014, *arXiv:1409.2329*. [Online]. Available: http://arxiv.org/abs/1409.2329
[33] D. P Kingma and M. Welling, "Auto-encoding variational Bayes," 2013, *arXiv:1312.6114*. [Online]. Available: http://arxiv.org/abs/1312.6114
[34] C. Tu, E. Takeuchi, A. Carballo, and K. Takeda, "Point cloud compression for 3D LiDAR sensor using recurrent neural network with residual blocks," in *Proc. Int. Conf. Robot. Automat. (ICRA)*, May 2019, pp. 3274–3280.
[35] T. Huang and Y. Liu, "3D point cloud geometry compression on deep learning," in *Proc. 27th ACM Int. Conf. Multimedia*, Oct. 2019, pp. 890–898.
[36] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.
[37] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding (HEVC) standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.
[38] E. Strigel, D. Meissner, F. Seeliger, B. Wilking, and K. Dietmayer, "The ko-PER intersection laserscanner and video dataset," in *Proc. 17th Int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2014, pp. 1900–1901.
[39] Q.-Y. Zhou, J. Park, and V. Koltun, "Open3D: A modern library for 3D data processing," 2018, *arXiv:1801.09847*. [Online]. Available: http://arxiv.org/abs/1801.09847
[40] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Automat.*, May 2011, pp. 1–4.
[41] Y. Liu and R. Zhong, "Buildings and terrain of urban area point cloud segmentation based on PCL," *IOP Conf. Ser., Earth Environ. Sci.*, vol. 17, no. 1, 2014, Art. no. 012238.
[42] T. Wiemann, F. Igelbrink, S. Pütz, M. K. Piening, S. Schupp, S. Hinderink, J. Vana, and J. Hertzberg, "Compressing ROS sensor and geometry messages with draco," in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Sep. 2019, pp. 243–248.
[43] Y. Inagaki, R. Shinkuma, T. Sato, and E. Oki, "Prioritization of mobile IoT data transmission based on data importance extracted from machine learning model," *IEEE Access*, vol. 7, pp. 93611–93620, 2019.
[44] R. Shinkuma and T. Nishio, "Data assessment and prioritization in mobile networks for real-time prediction of spatial information with machine learning," in *Proc. IEEE 1st Int. Workshop Netw. Meets Intell. Comput. (NMIC)*, Jul. 2019, pp. 1–4.
[45] Y. Yamada, R. Shinkuma, T. Sato, and E. Oki, "Feature-selection based data prioritization in mobile traffic prediction using machine learning," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–6.
[46] Q. Huynh-Thu and M. Ghanbari, "Scope of validity of PSNR in image/video quality assessment," *Electron. Lett.*, vol. 44, no. 13, pp. 800–801, Jun. 2008.
[47] A. Hore and D. Ziou, "Image quality metrics: PSNR vs. SSIM," in *Proc. 20th Int. Conf. Pattern Recognit.*, Aug. 2010, pp. 2366–2369.
[48] C. Zhang, D. Florencio, and C. Loop, "Point cloud attribute compression with graph transform," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Oct. 2014, pp. 2066–2070.
[49] T. Golla and R. Klein, "Real-time point cloud compression," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Sep. 2015, pp. 5087–5092.
[50] R. L. de Queiroz and P. A. Chou, "Compression of 3D point clouds using a region-adaptive hierarchical transform," *IEEE Trans. Image Process.*, vol. 25, no. 8, pp. 3947–3956, Aug. 2016.
[51] D. Tian, H. Ochimizu, C. Feng, R. Cohen, and A. Vetro, "Geometric distortion metrics for point cloud compression," in *Proc. IEEE Int. Conf. Image Process. (ICIP)*, Sep. 2017, pp. 3460–3464.
[52] D. Girardeau-Montaut, M. Roux, R. Marc, and G. Thibault, "Change detection on points cloud data acquired with a ground laser scanner," *Int. Arch. Photogramm., Remote Sens. Spatial Inf. Sci.*, vol. 36, no. 3, p. W19, 2005.
[53] M. Muja and D. G. Lowe, "Flann, fast library for approximate nearest neighbors," in *Proc. Int. Conf. Comput. Vis. Theory Appl. (VISAPP)*, vol. 3, 2009.
[54] E. Alexiou and T. Ebrahimi, "Point cloud quality assessment metric based on angular similarity," in *Proc. IEEE Int. Conf. Multimedia Expo (ICME)*, Jul. 2018, pp. 1–6.
[55] K. Mamou, T. Zaharia, and F. Prêteux, "TFAN: A low complexity 3D mesh compression algorithm," *Comput. Animation Virtual Worlds*, vol. 20, nos. 2–3, pp. 343–354, Jun. 2009.

KEIICHIRO SATO received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2018, and the master's degree from the Graduate School of Informatics, Kyoto University, in 2020. His research interest includes machine learning to predict spatial information in mobile networks.

RYOICHI SHINKUMA (Senior Member, IEEE) received the B.E., M.E., and Ph.D. degrees in communications engineering from Osaka University, in 2000, 2001, and 2003, respectively. In 2003, he joined the Communications and Computer Engineering Faculty, Graduate School of Informatics, Kyoto University, where he is currently an Associate Professor. He was a Visiting Scholar with the Wireless Information Network Laboratory, Rutgers University, from fall 2008 to fall 2009. His research interest includes cooperation in heterogeneous networks. He is a Fellow of IEICE. He received the Young Researchers' Award from IEICE, in 2006, and the Young Scientist Award from Ericsson Japan, in 2007, the TELECOM System Technology Award from the Telecommunications Advancement Foundation, in 2016, and the Best Tutorial Paper Award from the IEICE Communications Society, in 2019. He was the Chairperson of the Mobile Network and Applications Technical Committee of the IEICE Communications Society, from 2017 to 2019.

TAKEHIRO SATO (Member, IEEE) received the B.E., M.E., and Ph.D. degrees in engineering from Keio University, in 2010, 2011, and 2016, respectively. From 2011 to 2012, he was a Research Assistant with the Keio University Global COE Program, High-level Global Cooperation for Leading-edge Platform on Access Spaces, established by the Ministry of Education, Culture, Sports, Science and Technology of Japan. From 2012 to 2015, he was a Research Fellow with the Japan Society for the Promotion of Science. From 2016 to 2017, he was a Research Associate with the Graduate School of Science and Technology, Keio University. He is currently an Assistant Professor with the Graduate School of Informatics, Kyoto University. His research interests include communication protocols and network architecture for next-generation optical networks. He is a member of IEICE.

EIJI OKI (Fellow, IEEE) received the B.E. and M.E. degrees in instrumentation engineering and the Ph.D. degree in electrical engineering from Keio University, in 1991, 1993, and 1999, respectively. From 1993 to 2018, he was with the Nippon Telegraph and Telephone Corporation (NTT) Communication Switching Laboratories, Tokyo, Japan, where he was involved in researching network design and control, traffic control methods, and switching systems. From 2000 to 2001, he was a Visiting Scholar with the Polytechnic Institute, New York University, where he was involved in designing switch/router systems. From 2008 to 2017, he was with the University of Electro-Communications in Tokyo. He joined Kyoto University, in 2017, where he is currently a Professor. He has been active in the standardization of the path computation element and GMPLS in the IETF. He has authored over ten IETF RFCs. He has authored/coauthored five books, such as *Broadband Packet Switching Technologies* (Wiley, 2001), *GMPLS Technologies* (CRC Press, 2005), *Advanced Internet Protocols, Services, and Applications* (Wiley, 2012), *Linear Programming and Algorithms for Communication Networks* (CRC Press, 2012), and *Routing and Wavelength Assignment for WDM-based Optical Networks* (Springer, 2016). He is a Fellow of IEICE. He was a recipient of several prestigious awards, including the 1999 IEICE Excellent Paper Award, the 2001 IEEE Communications Society Asia-Pacific Outstanding Young Researcher Award, the 2010 Telecom System Technology Prize awarded by the Telecommunications Advanced Foundation, the IEEE HPSR 2012 Outstanding Paper Award, the IEEE HPSR 2014 Best Paper Award Finalist (first runner-up), the 2015 IEICE Achievement Award, the IEEE Globecom 2015 Best Paper Award, the 2016 Fabio Neri Best Paper Award (runner-up), and the 2018 Excellent Paper Award for Information and Communication Technology on Convergence.

TAKANORI IWAI received the B.E. and M.E. degrees in electrical and electronic engineering from Shinshu University, in 2002 and 2004, respectively. He is currently pursuing the Ph.D. degree in integrated design engineering with Keio University. In 2004, he joined NEC Corporation. He is also a Principal Researcher with the System Platform Research Laboratories. His research interests include system control, mobile and wireless networking, and the IoT service networks. He received the Electrical Science and Engineering Award from the Promotion Foundation for Electrical Science and Engineering, in 2016, and the Best Paper Award from the IEEE ComSoc International Communications Quality and Reliability Workshop (CQR '17).

DAI KANETOMO received the B.E. and M.E. degrees in administration engineering from Keio University, in 1994 and 1996, respectively. From 1999 to 2000, he was a Visiting Researcher with Stanford University. He joined NEC Corporation, in 2004, where he has been involved in research on multimedia communication. His research interests include remote machine control and live video streaming on cellular networks and their QoE improvement.

KOZO SATODA (Member, IEEE) received the B.E. and M.E. degrees in electrical engineering from Kyoto University, in 1991 and 1993, respectively. He joined NEC Corporation, in 1993. He has been a Senior Manager with the System Platform Research Laboratories, NEC Corporation, since 2016. His research interests include multimedia communication, streaming, and mobile traffic management. He is a member of IEICE. He received the Best Paper Award at IEEE CQR 2010, the Best Paper Award at IEEE CCNC 2017, the IEICE Communications Society Excellent Paper Award, in 2016, 63rd Electrical Science and Engineering Promotion Awards, and the 2016 IPSJ Industrial Achievement Award.