IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 22, 2020, accepted April 7, 2021, date of publication April 23, 2021, date of current version May 5, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075275

Data-Importance-Aware Bandwidth-Allocation Scheme for Point-Cloud Transmission in Multiple LIDAR Sensors

RYO OTSU[1], RYOICHI SHINKUMA[1,2], (Senior Member, IEEE), TAKEHIRO SATO[1], (Member, IEEE), AND EIJI OKI[1], (Fellow, IEEE)

[1]Graduate School of Informatics, Kyoto University, Kyoto 606-8501, Japan
[2]Faculty of Engineering, Shibaura Institute of Technology, Tokyo 135-8548, Japan

Corresponding author: Ryoichi Shinkuma (ryoichi_shinkuma@ieee.org)

This work was supported in part by the Japan Science and Technology Agency as PRESTO under Grant JPMJPR1854, and in part by the National Institute of Information and Communications Technology (NICT), Japan.

ABSTRACT This paper addresses bandwidth allocation to multiple light detection and ranging (LIDAR) sensors for smart monitoring, which a limited communication capacity is available to transmit a large volume of point-cloud data from the sensors to an edge server in real time. To deal with the limited capacity of the communication channel, we propose a bandwidth-allocation scheme that assigns multiple point-cloud compression formats to each LIDAR sensor in accordance with the spatial importance of the point-cloud data transmitted by the sensor. Spatial importance is determined by estimating how objects, such as cars, trucks, bikes, and pedestrians, are likely to exist since regions where objects are more likely to exist are more useful for smart monitoring. A numerical study using a real point-cloud dataset obtained at an intersection indicates that the proposed scheme is superior to the benchmarks in terms of the distributions of data volumes among LIDAR sensors and quality of point-cloud data received by the edge server.

INDEX TERMS Smart monitoring, LIDAR sensor, bandwidth allocation, point cloud compression.

I. INTRODUCTION

Smart monitoring systems, particularly at intersections, are expected to be a road-safety solution from the perspective of smart cities. As Datondji *et al.* suggested [1], intersection safety is a critical global issue since accidents at intersections are a major cause of road fatalities. Smart monitoring for obtaining spatial information in real time from 3D image sensors called light detection and ranging (LIDAR) sensors is a promising to prevent accidents at intersections [2], [3]. A smart monitoring system using multiple LIDAR sensors is effective in detecting objects; a single LIDAR sensor cannot obtain information of the back side of the target object, and the target object can be easily blocked by obstacles.

An issue with this system is that, since multiple LIDAR sensors share the limited capacity of the communication bandwidth from the sensors to an edge server, the communication channel becomes a bottleneck in real-time transmission of point-cloud data as the number of LIDAR sensors or the volume of point-cloud data increases. Due to this bottleneck, some of the data are dropped in real-time transmission; only successfully received data are used at the edge server, which leads to performance deterioration in smart monitoring. Although Sato et al. previously discussed this issue, bandwidth allocation to each LIDAR sensor was not taken in account [3].

In this paper, we propose a bandwidth-allocation scheme that allocates multiple point-cloud compression formats to each LIDAR sensor in accordance with the spatial importance of the point-cloud data transmitted from a LIDAR sensor. Spatial importance is determined by estimating how objects, such as cars, trucks, bicycles, and pedestrians, are likely to exist; regions where objects are more likely to exist are more useful for smart monitoring. The proposed scheme indirectly controls bandwidth by assigning the compression ratio of point-cloud data; as a higher compression ratio is assigned (data is less compressed), larger bandwidth is allocated from the limited capacity of the total bandwidth. We evaluate the proposed bandwidth-allocation scheme and compare it with benchmark schemes through a numerical study using a real

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

point-cloud dataset obtained at an intersection in terms of the distributions of data volumes among LIDAR sensors and quality of point-cloud data received by the edge server.

The remainder of this paper is as follows. Section II reviews previous studies conducted on bandwidth allocation for image-sensor networks. Section III presents the proposed scheme, and Section IV presents an extension of the proposed scheme. Section V presents the performance evaluation using a point-cloud dataset. Finally, Section VI concludes the paper.

II. RELATED WORK

A. 3D-IMAGE SENSOR NETWORK

LIDAR sensors are often used for autonomous vehicles [4]. However, the deployment of a network with multiple LIDAR sensors has been also investigated. In 2006, the intelligent transportation systems (ITS) group at the University of Minnesota developed a testbed system that placed a network of radar and LIDAR sensors near a rural intersection [5]. The network constantly monitored this intersection and collected road-traffic data from various perspectives. In 2009, Zhao *et al.* proposed a system for monitoring an intersection using a network of LIDAR sensors and video cameras [6]. The expected output of the system is motion trajectories of each moving object that entered the intersection and an estimation of its class (e.g., car, bus, bicycle, or pedestrian). The monitoring system can estimate a transformation within the horizontal plane (i.e., two translation and one rotation parameters) between the coordinate systems of two neighboring LIDAR sensors. Using a LIDAR sensor as the reference frame, all laser points from different LIDAR sensors can be transformed into a common coordinate system. In addition, all client computers are connected through a network to a server computer. The server computer broadcasts its local time periodically to synchronize the time of all computers. In 2014, Strigel *et al.* published a dataset collected at a public intersection in Aschaffenburg, Germany [2]. Four LIDAR sensors covered a wide 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).

B. POINT-CLOUD COMPRESSION

Point-cloud compression has been extensively studied to tackle the problem of the limitation capacity of the communication channel becoming a severe bottleneck affecting real-time transmission of a large volume of point-cloud data.

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 [7], [8]. A kd tree is another structure commonly used to represent a point cloud. The tree is constructed recursively in a top-down fashion by selecting 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 [9]. Devillers *et al.* used the kd-tree approach to recursively subdivide the bounding box of a point cloud [10]. In 2017, Google established the open source Draco [11], which uses a kd-tree structure and arithmetic coding to quantify and organize points in 3D space. A new ad-hoc group has recently been initiated for MPEG Point Cloud Compression (MPEG PCC) [12], [13]. The group focuses on developing point-cloud compression standards and has made significant progress in point-cloud compression. Sato *et al.* compared octree and Draco in the context of the spatial characteristics of point-cloud data [3].

Some studies have attempted to convert 3D point-cloud data into 2D structures 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 [14]. Kohira and Masuda mapped point-cloud data onto 2D pixels using GPS time and the parameters of the laser scanner [15].

C. DATA-IMPORTANCE-BASED BANDWIDTH ALLOCATION

After the 3rd-generation cellular system and IEEE802.11 were standardized around 2000, quality-of-service (QoS) differentiation technology started attracting attention because both standards were promising for providing multimedia services in wireless environments. Xiao *et al.* presented a bandwidth-allocation scheme for multi-class multimedia services in wireless/mobile networks [16]. They defined the QoS degradation metric and discussed inter-fairness and intra-fairness among QoS classes and in each QoS class. Shinkuma *et al.* presented a scheme that uses a more bandwidth-efficient transmission format to transmit web content less likely requested by users, which results in efficient bandwidth usage in wireless/mobile web services [17]. They also proposed a differentiated video-transmission scheme that allocates bandwidth in accordance with the error sensitivity of video data; the more sensitive the data, the larger the amount of bandwidth is assigned to ensure the robustness of the transmission [18]. Nasser *et al.* considered diverse bandwidth requirements of real-time multimedia traffic and proposed a scheme that prioritizes bandwidth allocation and call-admission control for multiple classes [19].

Much research has led to the standardization of QoS differentiation in commercialized wireless/mobile networks [20]. The recent development of network virtualization technology in the 5G context enables flexible bandwidth allocation to groups of communication traffic that belong to the same class by assigning a virtual network slice to each group [21].

Although there has been much research on QoS differentiation, bandwidth allocation of importance-aware transmission of point-cloud data has not been investigated. Adaptive encoding is an indirect means of bandwidth allocation; data

with a lower coding ratio require a smaller bandwidth for transmission [22]. This is the principle upon which the proposed scheme is built.

III. PROPOSED SCHEME

A. SYSTEM MODEL

Figure 1 illustrates the system model we assume in this paper. The system consists of an edge server and multiple devices equipped with image sensor units (hereafter "sensor devices"). Each sensor device generates streaming point-cloud data frame by frame sequentially. Point-cloud data are transmitted from multiple sensor devices to the edge server via a wireless network in this system. The capturer of each sensor device acquires point-cloud data at every frame delivered from the sensor device and stores them in the buffer. The encoder compresses point-cloud data using the compression format instructed by the controller of the edge server. The transmitter then sends the compressed data to the edge server. The receiver of the edge server receives the compressed data from the sensor devices. The decoder decompresses the compressed data with the compression format instructed by the controller. The estimator recognizes objects, such as cars, trucks, bicycles, and pedestrians, from the point-cloud data and estimates the spatial importance of the point-cloud data provided from each sensor device and stores them in the database. The controller determines the bandwidth allocation for each sensor device on the basis of the spatial importance stored in the database.

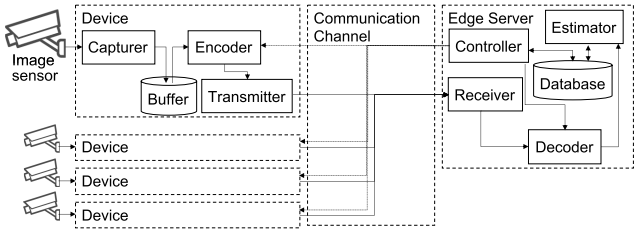


FIGURE 1. Overview of system model for the proposed scheme.

B. PROBLEM FORMULATION

This section presents the problem formulation of the system model in Fig. 1, which is given as

$$\max \sum_{i \in I} q(p_i(x_i)) \tag{1}$$

$$\text{s.t.} \sum_{i \in I} b(p_i(x_i)) \leq C. \tag{2}$$

The given parameters in the problem formulation are the set of all sensor devices I and the total bandwidth capacity from the transmitter to the receiver C. The decision variables in the problem formulation are the point-cloud data x_i and the compression format p_i of sensor device i. $q()$ is the quality of decompressed point-cloud data that depends on p_i, and $b()$ is the bandwidth allocated to sensor device i by selecting p_i as the compression format. Equations (1) and (2) are the objective and constraint of the problem formulation of the proposed scheme, respectively. Equation (1) means that the objective of the proposed scheme is to maximize the total quality of decompressed point-cloud data by selecting the appropriate p_i. Equation (2) means that the total volume of data transmitted from sensor devices does not exceed C.

Obviously, Eq. (1) cannot be solved because the quality of the point-cloud data becomes measurable after the point cloud is decompressed. In the proposed scheme, the quality of the decompressed point cloud in specific spatial regions where objects are likely to exist should be prioritized. Therefore, in the proposed scheme, Eq. (1) is replaced with

$$\max \sum_{i \in I} s_i b(p_i(x_i)). \tag{3}$$

This means that the proposed scheme maximizes the sum of the bandwidth for sensor device i weighted by s_i, which is the spatial importance score of sensor device i and explained in detail in Section III-D3. A simple heuristic approach for this optimization is to select the higher compression ratio to the sensor device with a higher importance score to improve the quality of the spatial regions with high importance scores. The algorithm for the control procedure of bandwidth allocation is described in detail in Section III-C. Figure 2 illustrates an example of bandwidth allocation in the proposed scheme based on Eqs. (3) and (2). In this example, sensor devices I to IV transmit the point-cloud data with importance scores 1 to 4, respectively. Sensor devices with higher scores compress the point-cloud data with a higher compression ratio. Eventually, the compressed point-cloud data with high importance scores are transmitted while satisfying the requirement of C.

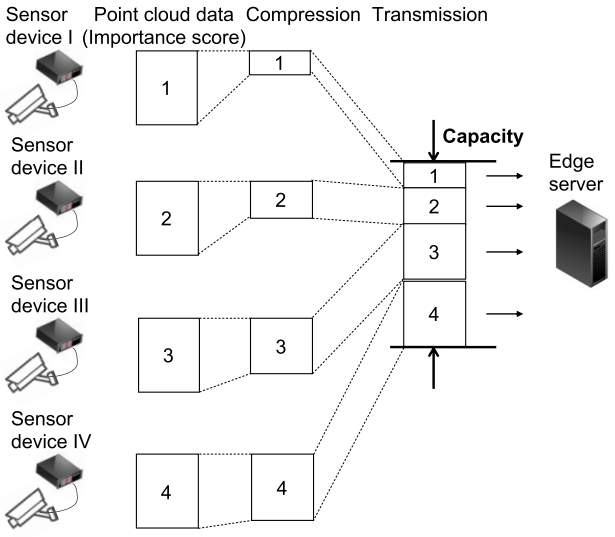


FIGURE 2. Example of bandwidth allocation in proposed scheme.

C. CONTROL PROCEDURE

The control procedure of bandwidth allocation of the proposed scheme is shown in Algorithm 1 and Figure 3.

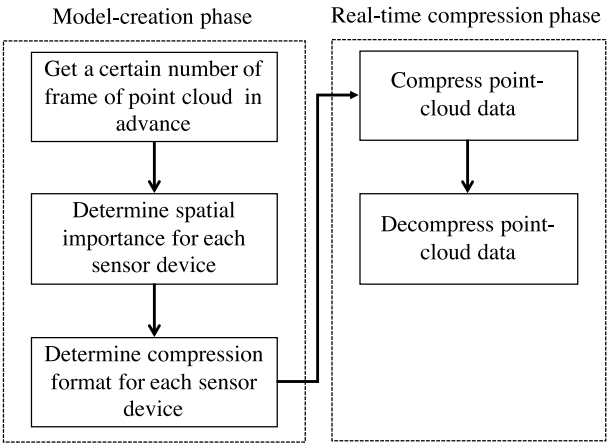


FIGURE 3. Block diagram of control procedure of proposed scheme.

Algorithm 1 Control Procedure of Proposed Scheme

Model-creation phase

1: $f_1, f_2, \ldots, f_l \leftarrow l$ frames of point cloud collected in advance
2: $s_i \leftarrow$ spatial importance of the sensor device estimated from $f_1, f_2, \ldots, f_l$, where i is the unique number of the sensor device
3: $p_i \leftarrow$ the compression format is determined from $s_1, s_2, \ldots, s_i$

Real-time compression phase

4: Send p_i to each device
5: Compress point-cloud data obtained from sensor device i, x_i with the p_i for i.
6: Send the compressed data from devices to the edge server
7: Receive the compressed data and carry out decompression with p_i

The control procedure roughly consists of two phases: model creation and real-time compression. In the model-creation phase, the importance score of each sensor device is estimated. The details of estimating the importance score are given later in Section III-D3. In the real-time-compression phase, compression is performed with the compression format determined for each sensor device.

Considering p_i, we assume octree-based compression, which is a commonly used compression method for point-cloud data [8], [23]. The compression of octree-based compression corresponds to setting the voxel size as a parameter; the smaller (larger) the voxel size is, the higher (lower) the compression ratio becomes while maintaining the high (low) quality of the point-cloud. Therefore, the proposed scheme uses a smaller voxel size for the sensor devices with higher importance scores. In the proposed scheme, the voxel size p_i is given as

$$p_i = p_{\max} - (p_{\max} - p_{\min})N((s_i)^k), \tag{4}$$

where $p_{\max}$ and $p_{\min}$ are the maximum and minimum voxel size in the range of p_i, respectively, $N()$ is a normalization function that scales from 0.0 to 1.0, and k is a given parameter that affects the normalization of s_i. In Eq. (4), k affects the distribution of $N()$. As k is set larger, smaller p_i is assigned to sensor devices with high-importance scores. As $p_{\max}$ and $p_{\min}$ are set larger, p_i increases for all sensor devices. It is difficult to optimize the determination of k, $p_{\max}$, and $p_{\min}$; we compared several combinations of them in the numerical study discussed in Section V.

D. NUMERICAL EXAMPLE OF BANDWIDTH ALLOCATION

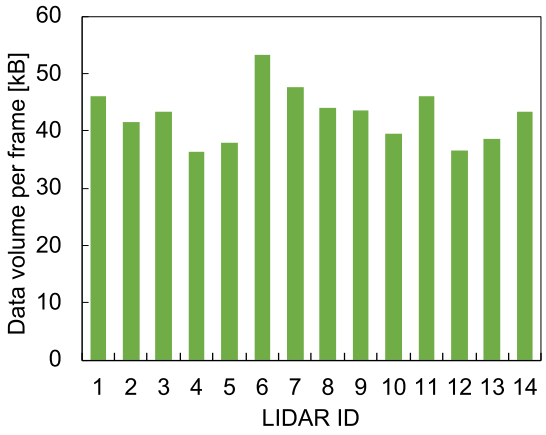
This section presents a numerical example of the bandwidth allocation of the proposed scheme.

1) SCENARIO AND DATASET

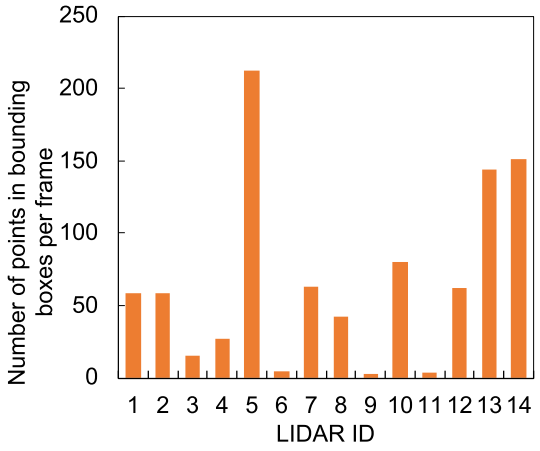
We assumed a scenario in which ten or more sensor devices at an intersection acquire point-cloud data. A dataset called the Ko-PER intersection dataset, which was published by Strigel *et al.* [2], was most suitable for our numerical study because it contains 3D image data obtained with 14 SICK LD-MRS 8-layer LIDAR sensor devices. Raw data in the dataset were 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 using the 14 sensor devices from different viewpoints. The sensor devices were installed on infrastructure components such as lamp posts and traffic lights and were mounted at least 5 m above the ground. Four sensor devices covered the area of the center of the central intersection, two sensors (scanners) monitored the sidewalks along the main road, and eight sensors monitored three egresses of the intersection. The sensor devices synchronously operated at a frequency of 12.5 Hz (80 ms). The dataset consists of Sequence1a-d, Sequence2, and Sequence3 and contains raw LIDAR data, undistorted camera images, reference data of selected vehicles, and object-label information. The object labels are cars, trucks, pedestrians, and bicycles. We used Sequence1a because it contains label information and has 1,211 frames. Each sensor device was calibrated in advance using calibration information from the Ko-PER intersection dataset and Open3D [24], an open source python programming library. The number of points in the calibrated point-cloud data of one frame was around 15,000.

2) COMPRESSION METHOD

Huang and Liu mentioned that octree-based compression is a commonly used compression method [25]. We used this compression method in our numerical study though



(a) Data volume before compression



(b) Number of points in bounding box

FIGURE 4. Distribution of point cloud data of each sensor device in Ko-PER dataset.

other compression methods mentioned in Section II such as Draco [11] and MPEG PCC [12], [13] are applicable for the proposed scheme. Octree-based compression allows us to easily control the compression ratio by just a few parameters.

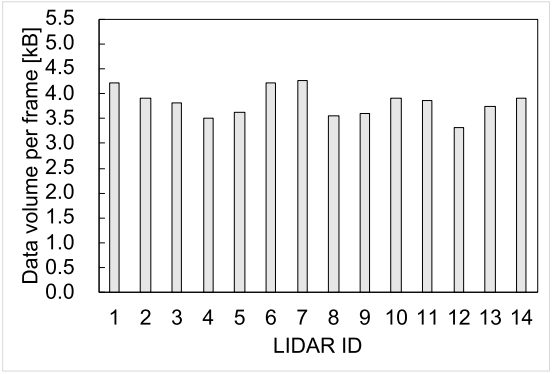
Octree-based compression is provided by an octree structure [23]. An octree is a tree with a structure suitable for sparse 3D data, in which each branch node represents a certain cube or cuboid bounding volume in space. Starting at the root, each branch has up to eight children, each one is a sub-octant of the node's bounding box. By traversing the tree in breadth-first order and outputting every child node configuration byte encountered, we are able to efficiently encode a point distribution in space. Our study involved using the Point Cloud Library (PCL) to implement octree-based compression. The 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 [26], [27]. The PCL provides parameters for octree-based compression. We used octree voxel size resolution, which corresponds to p_i in Eq. (4).

3) IMPORTANCE ESTIMATION

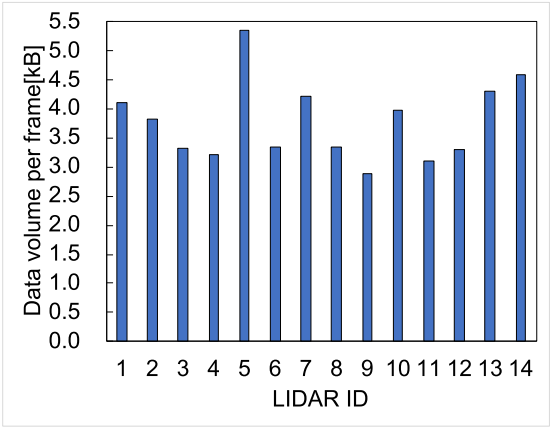
In our numerical study, we assumed that the bounding boxes given by the dataset are spatially important because the proposed scheme takes into account the regions where objects are likely to exist as important regions. Ten features of each object were obtained with sensor devices from the object-label information of the dataset. We used seven features, x position [m], y position [m], z position [m] (set to zero), object width [m], object length [m], object height [m], and orientation angle [rad], to extract the bounding box of the object. The point cloud was then divided into high-importance regions inside the bounding boxes and low-importance regions outside the bounding boxes. We counted the number of points inside the bounding boxes for each sensor device and defined the total number of points inside the bounding boxes as the importance score. This is reasonable because the sensor devices with a larger number of points inside the bounding boxes have more rich information about objects. Figures 4a and 4b show the point-cloud data volumes before compression of each sensor device and number of points in the bounding boxes per frame of each sensor device. As can be seen in Figure 4a, sensor devices 1, 6, 7, and 11 were top ranked in terms of data volume. As shown in Fig. 4b, however, data with a large volume did not always contain many points in the bounding boxes.

4) RESULTS

This section presents the data-volume results after compression, which suggests how bandwidth was allocated to each sensor device. Figures 5a and 5b show the data volumes of each sensor device after compression in the uniform and proposed schemes. The uniform scheme uses octree-based compression. However, different from the proposed scheme, the uniform scheme uses the same voxel size for all sensor devices without taking into account the importance of the point cloud. For comparison, we chose parameters of the uniform and proposed schemes that give almost the same normalized data rate. The normalized data rate was about 0.090. Note that the definition of normalized data rate is the data volumes after compression divided by the data volumes before compression. In the uniform scheme, data volumes uniformly decreased, as shown in Figure 5a, compared with those in Figure 4a. In the proposed scheme, compression was carried out in accordance with the number of points in the bounding boxes in Figure 5b. For example, the data volumes after compression of sensor devices 1, 2, 5, 7, 10, 13, and 14 were kept large because the numbers of points in the bounding boxes of those sensor devices were large. On the other hand, the data volumes after compression of sensor devices 6, 9, and 11 decreased because of the small numbers of points in the bounding boxes of those sensor devices. Thus, in the proposed scheme, the data of sensor devices with higher importance are compressed with a higher compression ratio, which means larger bandwidth is allocated to sensor devices



(a) Uniform ($p_i = 1.0 \times 10^{-2}$)



(b) Proposed ($p_{\max} = 1/32$, $p_{\min} = 5.0 \times 10^{-3}$, $k = 1/8$)

FIGURE 5. Data volume after compression.

with higher importance data, as expected with the proposed scheme.

IV. EXTENSION OF PROPOSED SCHEME

The proposed scheme presented in Section III performs as expected for prioritized transmission of point-cloud data in accordance with their spatial characteristics. However, there is still room for improvement because sensor devices with high (low) priority always use a high (low) compression ratio even when they are sending low (high) importance data. Therefore, this section presents an extension of the proposed scheme, which introduces the virtual device concept.

A. SYSTEM MODEL

Figure 6 illustrates the system model for the extended scheme. The architecture of the system is basically the same as that presented in Section III-A. However, we consider splitting sensor device i into two virtual sensor devices: high importance and low importance. Different compression parameters are set to each on the basis of the importance of the spatial regions. Compression of point cloud data is then carried out independently in each virtual sensor device. On the edge-server side, the controller determines the bandwidth

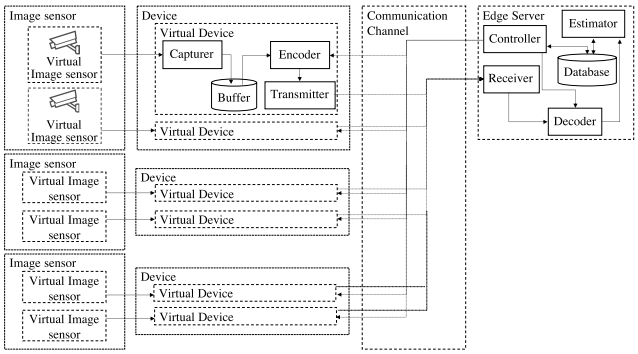


FIGURE 6. Overview of system model for extended scheme.

allocation to virtual sensor devices on the basis of the spatial importance of each virtual sensor device estimated by the estimator.

B. PROBLEM FORMULATION

The problem formulation presented in Section III-B is extended as follows. Equations (1) and (2) are redefined as

$$\max \sum_{i \in I} (q(p_{i,h}(x_{i,h}) + q(p_{i,l}(x_{i,l}))) \tag{5}$$

$$\text{s.t.} \sum_{i \in I} (b(p_{i,h}(x_{i,h})) + b(p_{i,l}(x_{i,l}))) \leq C. \tag{6}$$

The subscripts i, h and i, l denote virtual sensors of device i with high- and low- importances, respectively. The other notations are the same as in Eqs. (1) and (2).

Equation (5) is redefined as

$$\max \sum_{i \in I} (s_{i,h} b(p_{i,h}(x_{i,h})) + s_{i,l} b(p_{i,l}(x_{i,l}))). \tag{7}$$

This means that the extension maximizes the sum of the bandwidth for virtual sensor devices i, h and i, l weighted by $s_{i,h}$ and $s_{i,l}$, which are the spatial importance scores of the high- and low-importance regions of sensor device i, respectively. Again, we took a simple heuristic approach that selects the higher compression ratio to the region of the sensor device with the higher importance score to improve the quality of the spatial regions with high importance scores.

C. CONTROL PROCEDURE

The extend scheme is executed using Algorithm 1. For the compression format $p_{i,x}$ ($x = h$ or l), it uses the octree as described above and a smaller voxel size for virtual sensor devices with higher importance scores. The voxel size $p_{i,x}$ is given by Eq. (4).

D. BANDWIDTH-ALLOCATION EXAMPLE

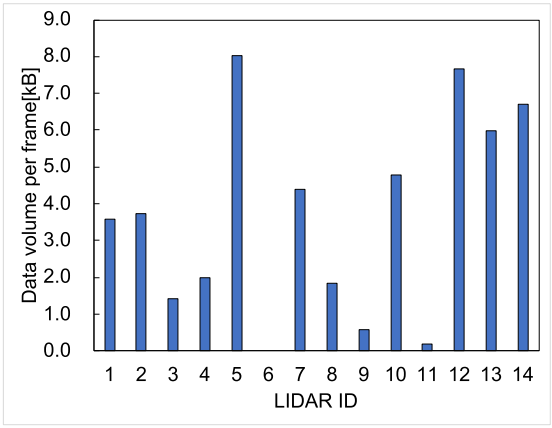
This section presents an example of bandwidth allocation of the extended scheme. We used the Ko-PER intersection dataset and octree-based compression, as we did for the proposed scheme, described in Section III-D.

1) IMPORTANCE ESTIMATION

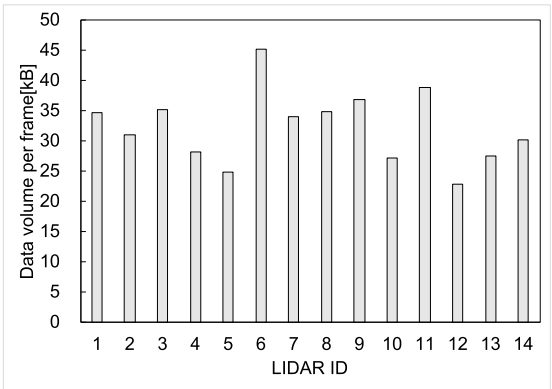
In our numerical study, we similarly assumed that the bounding boxes given by the dataset are spatially important because the extended scheme takes into account the regions where objects are likely to exist as important regions. The point cloud of a sensor device was then divided into high-importance regions inside the bounding boxes and low-importance regions outside the bounding boxes. We counted the number of points inside the bounding boxes for each virtual sensor device and defined the total number of points as the importance score of high-importance regions. In the extended scheme, the importance score of the low-importance spatial region of sensors devices was set to 0 because there are no bounding boxes in low-importance spatial regions, and the importance score of the high-importance spatial regions of sensor devices equals the importance score of those in the proposed scheme, as shown in Figure 4b. Therefore, by splitting the point-cloud data of a sensor device into high- and low-importance spatial regions, the high-importance spatial regions of sensor devices of low importance are scored higher than the low-importance spatial regions of LIDAR sensors of high importance.

2) RESULTS

This section presents the data volume after compression of each sensor device to discuss how bandwidth was allocated to each region of the devices. Figures 7a and 7b show the data volumes of the high- and low-importance regions of the sensor devices before compression with the extended scheme. Figures 8a and 8b also show the data volumes of the high- and low-importance regions of the sensor devices after compression. For comparison, we chose the same parameters for the high- and low-importance regions of the sensor devices; $p_{\max}$ was $1/32$, $p_{\min}$ was 5.0×10^{-3}, and k was $1/8$. Note that the compression ratio was determined from Eq. (4). As shown in Fig. 8b, the data volumes of the low-importance regions of the sensor devices decreased compared with those in Figure 7b, while in the high-importance regions of the sensor devices, compression was carried out in accordance with the number of points in the bounding boxes in Figure 4b. For instance, the data volumes after compression of the high-importance regions of sensor devices 1, 2, 5, 7, 10, 13, and 14 remained large because the numbers of points in the bounding boxes of the regions of these devices were large. On the other hand, the data volumes of the sensor devices with the low-importance regions decreased because the number of points in the bounding boxes were 0. Thus, in the extended scheme, the data of the regions of sensor devices with higher importance data were compressed with a higher compression ratio, which means a larger bandwidth is allocated to the regions of sensor devices with higher-importance data.



(a) High-importance region



(b) Low-importance region

FIGURE 7. Data volume before compression.

V. PERFORMANCE EVALUATION

First, we describe the performance evaluation metrics, which were the point-to-plane peak signal-to-noise ratio (PSNR) and classification accuracy. Then, we compare the proposed and extended schemes with the uniform compression and random sampling schemes.

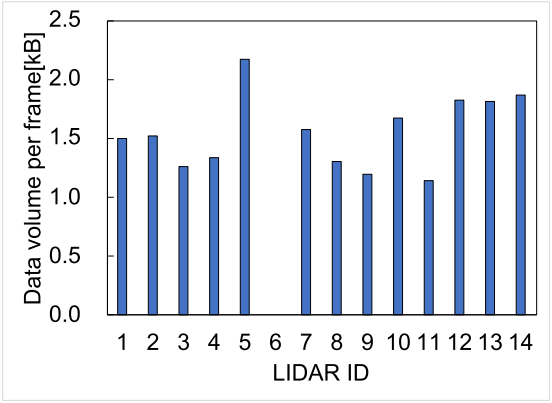
A. PSNR

The PSNR [28] of a point cloud was used as an objective quality metric. Previous studies on evaluating point-cloud compression [29]–[31] used the PSNR. The PSNR is defined as

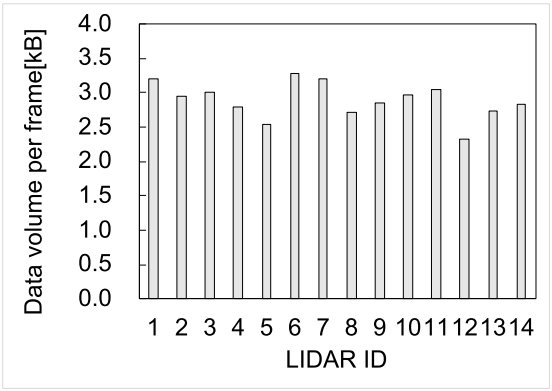
$$\mathrm{PSNR}(A, B) = 10 \log_{10} \frac{\mathrm{MAX}^2}{\mathrm{MSE}} [\mathrm{dB}], \tag{8}$$

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

MAX is conventionally defined as the length of the diagonal or the maximum of three sides of the bounding box of point cloud A. According to Tian *et al.*, one disadvantage of this definition is that, given the same amount of error for each point cloud, a spatially larger point cloud would produce a higher PSNR than a spatially smaller point cloud [28]. They proposed the maximum distance of the nearest-neighbor points to point cloud A, which selects MAX on the basis of the intrinsic resolution of point cloud A. They also defined MAX



(a) High-importance region



(b) Low-importance region

FIGURE 8. Data volume after compression.

as the maximum distances of nearest-neighbor points to point cloud A. They used the point-to-plane metrics for calculating MSE. The point-to-plane MSE is defined as

$$\mathrm{MSE}(A, B) = \frac{1}{N_A} \sum_{i=1}^{N_A} (\vec{E_{i,j}} \cdot \vec{N_i})^2, \tag{9}$$

where N_A, $\vec{E_{i,j}}$, and $\vec{N_i}$ represent the number of points in point cloud A, error vector between a_i and b_i, and normal vector of a_i, respectively. a_i and b_i are the point belonging to point cloud A and the nearest-neighbor point of a_i that belongs to point cloud B, respectively. The error from the surface of the structure is represented by taking the inner product of $\vec{E_{i,j}}$ and $\vec{N_i}$. Thus, the maximum of MSE (A,B) and MSE (B,A) is taken as MSE. In this study, the normal vector was calculated using an algorithm from the PCL. This algorithm estimates the normal vector from the nearest-neighbor points to the observation point. We set the number of nearest-neighbor points to 10.

Another definition of MSE is point-to-point MSE. The point-to-point definition calculates the MSE from the distance between points. Figure 9 shows the differences between the two metrics. The point-to-plane metric captures the surface features of the structure more effectively than the point-to-point one does. We thus used the point-to-plane MSE.

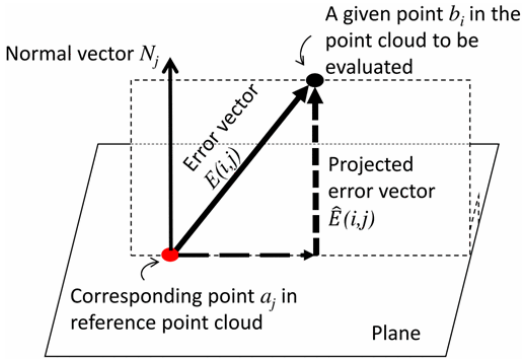


FIGURE 9. Point-to-point MSE vs. point-to-plane MSE [28].

B. CLASSIFICATION ACCURACY

The PSNR is not a practical measure of the quality of a point cloud. Therefore, we used accuracy of object classification as a more practical metric. Moreover, there are generally three methods of point-cloud evaluation; shape-classification object detection and point-cloud segmentation [32]. Shape classification classifies point clouds into certain categories, e.g., cars, bicycles, and pedestrians. Object detection detects objects as bounding boxes from the point cloud in a certain scene. Semantic segmentation divides a point cloud into several subsets according to the meaning in the scene. We performed shape classification in our study. The bounding boxes do not need to be detected because the information of bounding boxes is contained in the Ko-PER dataset.

There are two quality metrics for classification: overall accuracy (OA) and mean class accuracy (mAcc) [32]. OA is obtained by dividing the number of correct instances by the total number of instances. mAcc represents the mean accuracy for shape classes, which is derived from the average accuracy of each class. Table 1 shows an example, where OA is 0.95(= 95/100) and mAcc is 0.50(= (95/95 + 0/5)/2). Thus, OA and mAcc are not proportional; both needed to be used for classification evaluation.

We used Random Forest as a machine learning method for classification [33], [34]. We extracted all sets of points in each bounding box from the Ko-PER dataset using the label information. For machine learning, we used four features of each set of points: the total number of points and the maximum width, maximum height, and maximum length between two points in the bounding box. For machine-learning training, 1211 frames of data before compression were used after bonding boxes with 0 points or 0 width, height, and length were removed. We used the class ID as the label. We only

TABLE 1. Example of classification.

	Cars	Pedestrians	Total
Dataset	95	5	100
Correct instances	95	0	95

considered cars and pedestrians. For the machine-learning test, we used the top 100 frames with the largest number of points in a bounding box from Sequence1a of the Ko-PER dataset. We used the Random Forest of scikit-learn with the default setting and set random_state and n_jobs to 1 and −1 [35].

C. BENCHMARK SCHEMES

We compared the proposed and extended schemes with two benchmarks: uniform and sampling. The uniform scheme uses octree-based compression. However, different from the proposed scheme, the uniform scheme uses the same voxel size for all sensor devices without taking into account the importance of the point cloud. We set p_i to 1.0×10^{-5}, 1.0×10^{-4}, 1.0×10^{-3}, 1.0×10^{-2}, 1.0×10^{-1}, 0.25, 0.5, 0.75, 1.0, 2.0, 3.0, 4.0, 5.0, and 6.0. The sampling scheme performs random sampling for points with a given sampling rate. The sampling rate was set to 0.01, 0.05, 0.10, 0.15, 0.20, 0.25, 0.30, 0.35, 0.40, 0.45, 0.50, 0.55, 0.60, 0.65, 0.70, 0.75, 0.80, 0.85, 0.90, 0.95, and 0.99. Note that, for the sampling scheme, the results were averaged over ten trials by changing the random seed in each trial.

We note that other compression methods mentioned in Section II such as Draco and MPEG PCC are other options for the compression method used in the proposed scheme instead of octree-based compression rather than benchmark schemes for the proposed scheme. Table 2 compares octree-based compression, Draco, and MPEG PCC from three aspects: controllable, efficient, and specialized for stream. The advantage of octree-based compression is that it allows us to control the compression ratio by setting only a few parameters. It was reported that Draco performs compression more efficiently than octree-based compression in terms of data reduction and computation time [3]. Although octree-based compression and Draco are applicable for stream data of point cloud, they process the data frame by frame. MPEG PCC deals with stream data of point cloud as time-series data and considers the difference and the similarity between frames over time to improve the efficiency of the compression [12], [13].

TABLE 2. Comparison among compression methods for point cloud.

Method	Controllable	Efficient	Specialized for stream
Octree-based	✓		
Draco		✓	
MPEG PCC		✓	✓

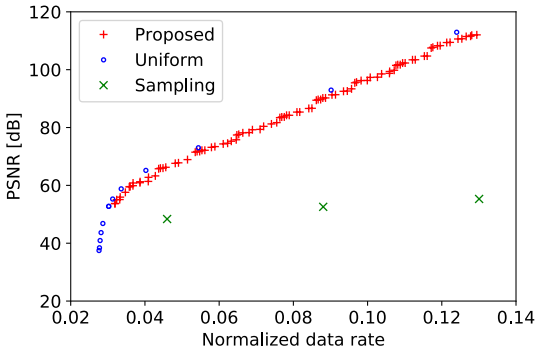
D. RESULTS

1) PSNR

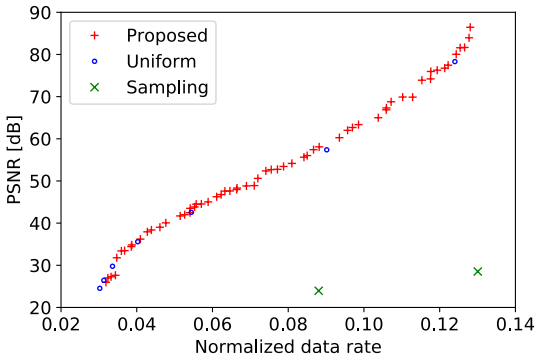
This section discusses the results of the performance evaluation using the PSNR as the quality metric. We first selected the top 100 frames with the largest number of points in a bounding box from Sequence1a and calculated the average PSNRs. The proposed scheme used octree-based compression, and it compressed by changing the voxel size p_i. $p_{\max}$ and $p_{\min}$ were set to the combinations listed in Table 3, and k was set to $2^n (n = -3, -2, -1, \ldots, 3)$.

TABLE 3. Parameter combinations used in proposed scheme ($p_{\max}$ and $p_{\min}$).

$p$max	$p_{\min}$
1	1.0×10^{-1}, 5.0×10^{-2}, 1.0×10^{-2}
1/2	5.0×10^{-2}, 1.0×10^{-2}, 5.0×10^{-3}
1/4	1.0×10^{-2}, 5.0×10^{-3}
1/8	1.0×10^{-2}, 5.0×10^{-3}
1/16	5.0×10^{-3}, 1.0×10^{-3}
1/32	1.0×10^{-3}, 5.0×10^{-4}
1/64	1.0×10^{-3}, 5.0×10^{-4}
1/128	5.0×10^{-4}, 1.0×10^{-4}
1/256	1.0×10^{-4}, 5.0×10^{-5}
1/512	1.0×10^{-4}, 5.0×10^{-5}



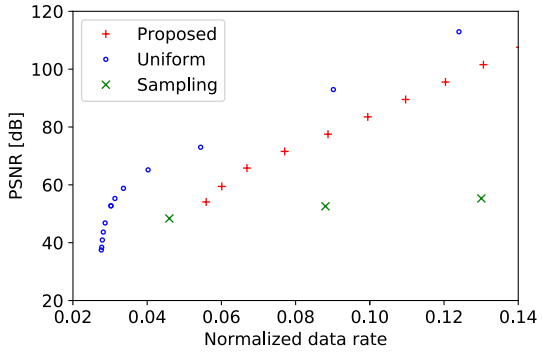
(a) Overall region



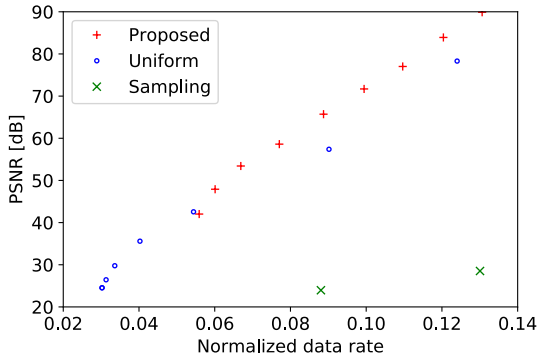
(b) High-importance region

FIGURE 10. PSNR vs. normalized data rate in proposed scheme.

Figure 10 shows the PSNR performance against the normalized data rate in the proposed scheme. Figures 10a and 10b show the PSNRs in the overall region and high-importance regions determined with the method presented in Section III-D3. As shown in Fig. 10a, the PSNR in the overall region of the proposed scheme was a little lower than that of the uniform scheme. However, the PSNR in the high-importance region of the proposed scheme was higher than that of the uniform scheme, as shown in Fig. 10b. The sampling scheme did not perform well compared with the uniform and proposed schemes, both of which used octree-based compression. Note that the PSNRs in Figs. 10a

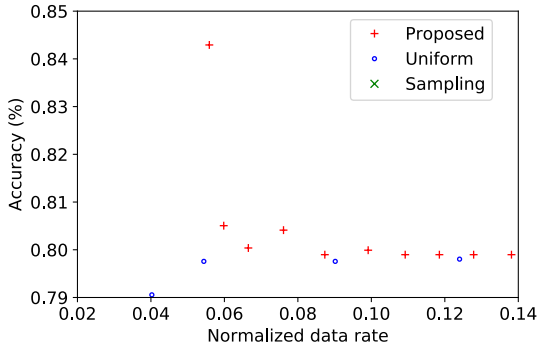


(a) Overall region

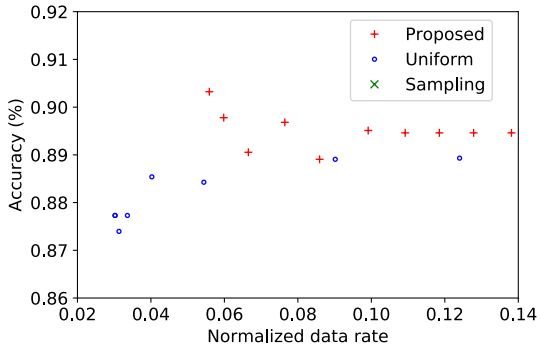


(b) High-importance region

FIGURE 11. PSNR vs. normalized data rate in extended scheme.



(a) Overall accuracy



(b) Mean class accuracy

FIGURE 12. Evaluation by classification.

and 10b are not comparable with each other; PSNRs for different images are generally not comparable. Obviously, the proposed scheme performed well by improving the PSNR in the high-importance region by appropriately allocating bandwidth to multiple sensor devices.

Figure 11 shows the PSNRs of the extended scheme. As we did for Fig. 10, if a combination of parameters achieved a better PSNR in the high-importance region at a lower normalized data rate than another combination, we removed the plot of the latter because the former is obviously more effective. Figures 11a and 11b show the PSNRs in the overall and high-important regions, respectively. The observation from Fig. 11 is basically the same as from Fig. 10; using the (extended) proposed scheme, PSNRs in the overall and high-importance regions decreased and increased, respectively, compared with the uniform scheme. However, the extended scheme improved the PSNR more than the proposed scheme did in the high-importance region compared with the uniform scheme.

2) CLASSIFICATION ACCURACY

This section presents accuracies of the classification obtained with the method described in Section V-B. Figures 12a and 12b show the OA and mAcc, respectively, which were described in Section V-B.

Figure 12 shows the classification performance against the normalized data rate. If a combination of parameters achieved a better result at a lower normalized data rate than another combination, we removed the plot of the latter because the former is obviously more effective. As shown in Fig. 12a, the OA of the extended scheme was higher than those of the uniform and sampling schemes. As shown in Fig. 12b, in terms of mAcc, the proposed scheme performed better than the uniform and sampling schemes.

Thus, the proposed scheme improved classification accuracy for the high-importance region by appropriately allocating bandwidth to regions of multiple sensor devices.

VI. CONCLUSION

We proposed a bandwidth-allocation scheme for multiple LIDAR sensor devices to improve the quality of received point-cloud data while satisfying the constraint of the bandwidth capacity of the communication channel from the sensor devices to the edge server. The proposed scheme sets the compression ratio higher for sensor devices with higher-importance point-cloud data. We also described an extension of the proposed scheme for the case in which there are two virtual sensors for each physical sensor device; it divides up the point-cloud data into high- and low-importance regions and assigns different compression ratios in accordance with the importance. A numerical study showed that the proposed scheme allocates bandwidth to multiple sen-

sor devices, as expected, and performed better than the benchmark schemes in terms of the quality of the received point-cloud data. We evaluated the extended scheme's object classification and found that it improved classification accuracy relative to that of a uniform scheme and sampling scheme.

For future work, the proposed scheme should be examined with other compression methods mentioned in Section II such as Draco and MPEG PCC instead of octree-based compression.

REFERENCES

[1] 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.

[2] 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.

[3] K. Sato, R. Shinkuma, T. Sato, E. Oki, T. Iwai, D. Kanetomo, and K. Satoda, "Prioritized transmission control of point cloud data obtained by LIDAR devices," *IEEE Access*, vol. 8, pp. 113779–113789, 2020.

[4] R. Dominguez, E. Onieva, J. Alonso, J. Villagra, and C. Gonzalez, "LIDAR based perception solution for autonomous vehicles," in *Proc. 11th Int. Conf. Intell. Syst. Design Appl.*, Nov. 2011, pp. 790–795.

[5] 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.

[6] 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, 2009.

[7] J. Peng and C. C. J. Kuo, "Octree-based progressive geometry encoder," in *Proc. 4th Internet Multimedia Manage. Syst.*, vol. 5242, Nov. 2003, pp. 301–311.

[8] Y. Huang, J. Peng, C.-C. J. Kuo, and M. Gopi, "Octree-based progressive geometry coding of point clouds," *SPBG*, vol. 6, pp. 103–110, 2006.

[9] 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.

[10] O. Devillers and P.-M. Gandoin, "Geometric compression for interactive transmission," in *Proc. Visualizat. (VIS)*, 2000, pp. 319–326.

[11] Google. (2017). *Draco: 3D Data Compression*. [Online]. Available: https://github.com/google/draco

[12] 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.

[13] E. Alexiou, I. Viola, T. M. Borges, T. A. Fonseca, R. L. de Queiroz, and T. Ebrahimi, "A comprehensive study of the rate-distortion performance in MPEG point cloud compression," *APSIPA Trans. Signal Inf. Process.*, vol. 8, pp. 1–27, Nov. 2019.

[14] H. Houshiar and A. Nüchter, "3D point cloud compression using conventional image compression for efficient data transmission," in *Proc. 25th Int. Conf. Inf., Commun. Autom. Technol. (ICAT)*, Oct. 2015, pp. 1–8.

[15] 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, pp. 99–106, Sep. 2017.

[16] Y. Xiao, C. L. P. Chen, and Y. Wang, "Fair bandwidth allocation for multi-class of adaptive multimedia services in wireless/mobile networks," in *Proc. IEEE VTS 53rd Veh. Technol. Conf.*, vol. 3, Spring 2001, pp. 2081–2085.

[17] R. Shinkuma, M. Okada, and S. Komaki, "Adaptive transmission scheme for Web prefetching in wireless environment," *IEICE Trans. Electron.*, vol. 85, no. 3, pp. 485–491, 2002.

[18] R. Shinkuma, M. Okada, and S. Komaki, "A new resource management scheme for multiple video transmission in wireless environment," *IEICE Trans. Commun.*, vol. 85, no. 10, pp. 2153–2160, 2002.

[19] N. Nasser and H. Hassanein, "Prioritized multi-class adaptive framework for multimedia wireless networks," in *Proc. IEEE Int. Conf. Commun.*, vol. 7, Jun. 2004, pp. 4295–4300.

[20] H. Ekstrom, "QoS control in the 3GPP evolved packet system," *IEEE Commun. Mag.*, vol. 47, no. 2, pp. 76–83, Feb. 2009.

[21] M. M. Rahman, C. Despins, and S. Affes, "Service differentiation in software defined virtual heterogeneous wireless networks," in *Proc. IEEE Int. Conf. Ubiquitous Wireless Broadband (ICUWB)*, Oct. 2015, pp. 1–5.

[22] X. Zhou and C.-Z. Xu, "Harmonic proportional bandwidth allocation and scheduling for service differentiation on streaming servers," *IEEE Trans. Parallel Distrib. Syst.*, vol. 15, no. 9, pp. 835–848, Sep. 2004.

[23] R. Schnabel and R. Klein, "Octree-based point-cloud compression," in *Proc. Eurographics Symp. Point-Based Graph.*, vol. 6, Jul. 2006, pp. 111–120.

[24] 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

[25] 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.

[26] R. B. Rusu and S. Cousins, "3D is here: Point cloud library (PCL)," in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2011, pp. 1–4.

[27] Y. Liu and R. Zhong, "Buildings and terrain of urban area point cloud segmentation based on PCL," in *Proc. IOP Conf., Earth Environ. Sci.*, vol. 17, no. 1. Bristol, U.K.: IOP, 2014, Art. no. 012238.

[28] 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.

[29] 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.

[30] 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.

[31] 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.

[32] Y. Guo, H. Wang, Q. Hu, H. Liu, L. Liu, and M. Bennamoun, "Deep learning for 3D point clouds: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, early access, Jun. 29, 2020, doi: 10.1109/TPAMI.2020.3005434.

[33] A. Liaw and M. Wiener, "Classification and regression by randomforest," *R News*, vol. 2, no. 3, pp. 18–22, 2002.

[34] G. Louppe, "Understanding random forests: From theory to practice," 2014, *arXiv:1407.7502*. [Online]. Available: http://arxiv.org/abs/1407.7502

[35] *SciKit-Learn 0.23.2 Random Forests*. Accessed: Apr. 20, 2021. [Online]. Available: https://scikit-learn.org/stable/modules/linear_model.html#lasso

RYO OTSU received the B.E. degree in electrical and electronic engineering from Kyoto University, in 2019, and the master's degree from the Graduate School of Informatics, Kyoto University, in 2021. His research interest includes 3-D image sensor 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. He worked as an Assistant Professor with the Graduate School of Informatics, Kyoto University, from 2003 to 2011, and as an Associate Professor, from 2011 to 2021. He was a Visiting Scholar with the Wireless Information Network Laboratory, Rutgers University, from Fall 2008 to Fall 2009. In 2021, he joined the Faculty of Engineering, Shibaura Institute of Technology, as a Professor. His main research interest includes cooperation in heterogeneous networks. He received the Young Researchers' Award from the IEICE, in 2006, 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. He is currently a fellow of the IEICE.

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, Japan. 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. From 2012 to 2015, he was a Research Fellow with the Japan Society for the Promotion of Science.

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, Yokohama, Japan, in 1991, 1993, and 1999, respectively. He was with Nippon Telegraph and Telephone Corporation (NTT) Laboratories, Tokyo, from 1993 to 2008, and The University of Electro-Communications, Tokyo, from 2008 to 2017. From 2000 to 2001, he was a Visiting Scholar with Polytechnic University, Brooklyn, New York. In 2017, he joined Kyoto University, Japan, where he is currently a Professor. His research interests include routing, switching, protocols, optimization, and traffic engineering in communication and information networks. He is a fellow of IEICE.