Real Time LiDAR Point Cloud Compression And Transmission For Intelligent Transportation System

Bhaskar Anand,Vivek Barsaiyan,Mrinal Senapati and P. Rajalakshmi
Department of Electrical Engineering,
Indian Institute of Technology
Hyderabad, India
email:{ee18resch11006,ee18mtech11028,ee18mtech11025,raji}@iith.ac.in

***Abstract*—Real-time data transmission is one of the challenging tasks for LiDAR (Light Detection and Ranging) based applications. These applications are becoming more popular in the field of Surveying and Intelligent Transportation System (ITS). The size of Point cloud data (pcd files) generated by LiDAR is generally quite large. In this paper, a real-time Point cloud transmission over Wi-Fi is suggested. Also in order to avoid the transmission of huge data, an Octree-based Point cloud Compression technique is used. This Compression technique is provided by Point cloud Library (PCL). We encode the Point clouds into text files of much lower size as compared to pcd file size. Instead of sending the Point cloud data we send the text files which is further decoded on the receiver side. The preliminary experimental results show that this method has the potential to be used for an exchange of information (3-D view or Point cloud) between two vehicles for Intelligent Transportation System.**

***Index Terms*—LiDAR, Point cloud, Octree Based Compression, Point cloud Library (PCL), Intelligent Transportation System (ITS)**

I. INTRODUCTION

LiDAR (Light Detection and Ranging) sensors are used to measure distance and depth data of surrounding objects by illuminating them with laser light pulses and receiving the reflected pulses. Based on the time difference between received and transmitted laser lights and speed of the laser light, it creates a 3-D representation of objects. LiDAR's ability to capture 3-D information makes it very useful in many fields like Tele-operation, Tele-rehabilitation, Intelligent Transportation System for smart cities etc..

In the field of Tele-operation, a robot is operated remotely. The 2-D image does not provide the depth information so sending 3-D Point clouds may ease the operation to a great extent. Similarly, for efficient traffic management, it is necessary to acquire and transmit data from different sensors such as cameras, LiDAR,Process exited normally RADAR etc. The sensor data can be transmitted among vehicles as well as between vehicles and Road Side Unit(RSU). This transfer of data can enable the sharing of recent updated traffic and road condition information. Point clouds consist of huge data sets describing three-dimensional points associated with additional information such as distance, color etc. and therefore occupy a significant amount of memory resources. In all such applications transmitting LiDAR data is a challenging task because of its size. Transmitting such huge amount of data may not be always feasible. In such cases, it is necessary to compress the Point cloud. Octree compression is one of the most widely used methods for Point cloud compression. We have used this method for compression. In this paper, we have suggested an experimental set up by which LiDAR data is compressed to text files which are of much smaller size and then text files are transmitted. On the receiver side, these text files are decompressed back to pcd files. The continuously incoming pcd files are merged and can be visualized. We have done a feasibility analysis of the set up for transmission over WiFi.

This paper is divided into five sections. The first section gives an introduction to LiDAR, its application and the need for compression and transmission techniques. The second part is a brief review of the related work done in this field. In the third section, we describe the Octree-based object representation, compression technique, set-up and also suggested method is explained. Results and analysis are shown in the fourth section. The fifth section gives the conclusion and future work.

II. RELATED WORK

For Tele-operation application, real-time Point cloud data need to be transmitted from a remote workstation to the teleoperator to obtain information regarding the environment around the workstation. In this scenario, Point cloud data compression become necessary to get real-time data without much time delay. In [1] authors have suggested a Point cloud compression method based on H.264/AVC(Advanced Video Coding). In their proposed algorithm, the Point cloud is first converted to disparity image. The disparity image is then encoded using H.264/AVC and the encoded data is transmitted through the network in packets. On the receiver side encoded packets are decoded to get the disparity image which in turn is converted back to Point cloud.

[2] has used Octree-based Point cloud compression technique to send Point cloud data for telerehabilitation system application. A dynamic compression scheme has been suggested by the author. In this method, the difference between two successive frames is determined and based on this, transmission of the next frame is decided. If the difference is too small, the next frame will not be transmitted.

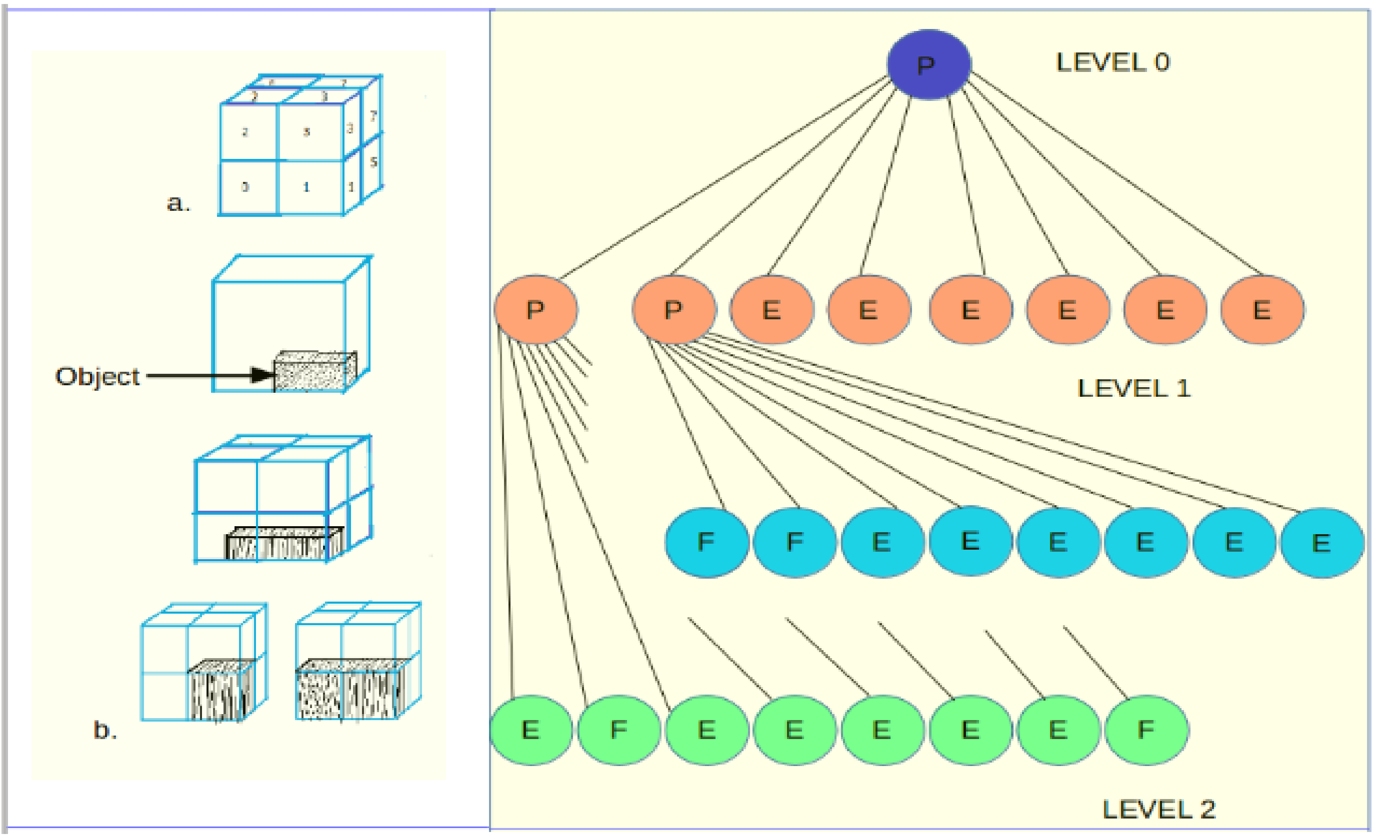

Fig. 1. Object representation using Octree structure('P'='Partially Filled','E'='Empty','F'='Completely Filled') [6]

Kammerl et al. [3] have exploited the temporal and spatial redundancy in the Point cloud for compression. Here also two consecutive Point clouds are compared and their structural difference is encoded. Here the author presents a lossy compression approach for Point cloud streams. Their proposed compression technique can handle Point cloud streams of arbitrary and varying size, point order and point density. By encoding Point clouds structural differences, they successively decode the Point clouds at the decoder side.

In [4] the authors have suggested a compression method in which points are hierarchically clustered. In this method, points are clustered so as to generate Point cloud with a lesser level of details (LOD). The process is repeated in an iterative manner to form a hierarchy of LOD.

[5] gives a method of compression using plane extraction and color segmentation. Here Delaunay triangulation is used to represent points from the plane of the scene.

III. Octree Compression and Transmission

A. Octree Compression

Before going into details of octree based Point cloud compression technique, a small introduction on Octree is needed here.

An Octree is a cube-shaped tree data structure in which surrounding space is divided into eight smaller cubic regions (child node). Each child node then recursively divides the smaller cubic region until it locates each cloud point by serialized Octree pattern. Each cube in Octree has the same spatial and time complexity. Any object in the space can be represented using this Octree-based encoding technique. The root node is positioned at level-0 and all other nodes positioned below the root node. An encoded Object 'O' (in figure 1) is thus represented by a family of ordered pairs O(n)=(S,D(n)),where S is a finite set of properties and D is the set of disjoint object elements which totally fill the space at resolution label 'n'.The properties used here can be 'E','P' or 'F' indicating the cube is empty from the object, partially filled with the object or is completely filled with object [6].

The position of a child node takes values from the child number set 0,1,2.......,k-1 where $k = 2^N$.Each node is completely identified by a node address string which is taken from child number set. The root positioned at level-0 is represented by an empty string. The node address of a child is the child number prefixed by the address string of the parent node. By this way, we can identify each node and also traversal order to reach that node in Octree Structure [6].

Now we briefly discuss the Octree-based compression technique given by [7] [8]. From a given Point cloud P, an Octree structure O can be created with a maximum number of Levels L and all the points will be put in the cubic cells in a sorted manner. Each point will be then quantized for compression. With L level Octree structure precision will be L bits per

Parameters	IITH data (32768 points)	IITH data (65536 points)	IITH data (131072 points)	KITTI data (64 channel)	Ouster data (64 channel)
original pcd size(KB)	342	748	1401	1962	2950
compressed text file size (KB)	94	267	488	392	349
compression per frame (%)	27.48	35.69	34.80	19.97	11.83
compression ratio per frame	3.63	2.80	2.87	5.0	5.45

TABLE I
COMPRESSION RESULTS

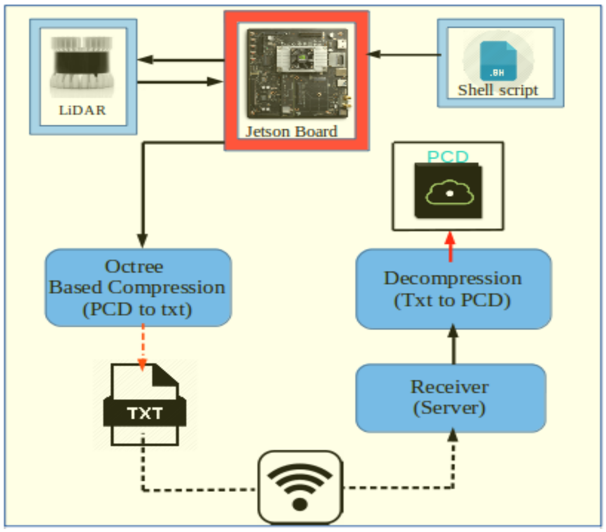

Fig. 2. Set Up for LiDAR data Transmission

coordinate direction. To encode the Octree we have to move to top-down and breadth-first fashion. Starting from the root node, each non-empty child node is encoded. Each cubic cell is represented by a byte where each bit representing a child cell. The decoder can faithfully recover the Octree by traversing the same way as the encoder.

B. Method and Setup

Here we propose an Octree-based Point cloud data compression and transmission set up for Intelligent Transportation System. In this method, a LiDAR can be mounted on a car which captures the surrounding environment as 3-D Point cloud data. In heavy traffic conditions, we have a large number of vehicles in a queue one after the other. In such a scenario, the front view of a vehicle is blocked by the vehicles ahead of it. The Point cloud captured by the front car can be transmitted to the car behind it as well as to Road Side Unit(RSU). This transmission will be of great help in sharing the latest updated traffic condition and road condition among all vehicles(see figure 3).

The problem faced in real-time transmission of the Point cloud is its size. Point cloud data is of large size (usually several Megabytes). So we compress the data to encode it into text files and then transmit it from the sender vehicle. For compression, we use Octree-based compression technique provided by PCL(Point cloud Library). At the vehicle on the receiver side, this text file is decompressed to pcd file (Point cloud data), which gives a 3-D view of the surrounding. By compressing the pcd files, we are also able to increase the transmission rate and decrease the latency during transmission.

We captured the Point cloud data using Ouster LiDAR (OS-1). NVIDIA Jetson TX2 board is used to save and transmit the data. After each reboot, multiple shell scripts run on the board. Using one of the scripts the board starts collecting data from the LiDAR after booting. The second script automatically connects to the Wi-fi created by client side (local server). Once the connection is established transmission of uncompressed Point clouds or compressed text files start. Test cases were run for both static and dynamic condition. Here static refers to the case in which sensor (transmitter side client) and local server (receiver side client) are both static. Dynamic is the case when the sensor is moving but the receiver side client is static. The speed of the vehicle carrying sensor was approximately 30 Km/Hour. A schematic diagram of set up is shown in figure 2.

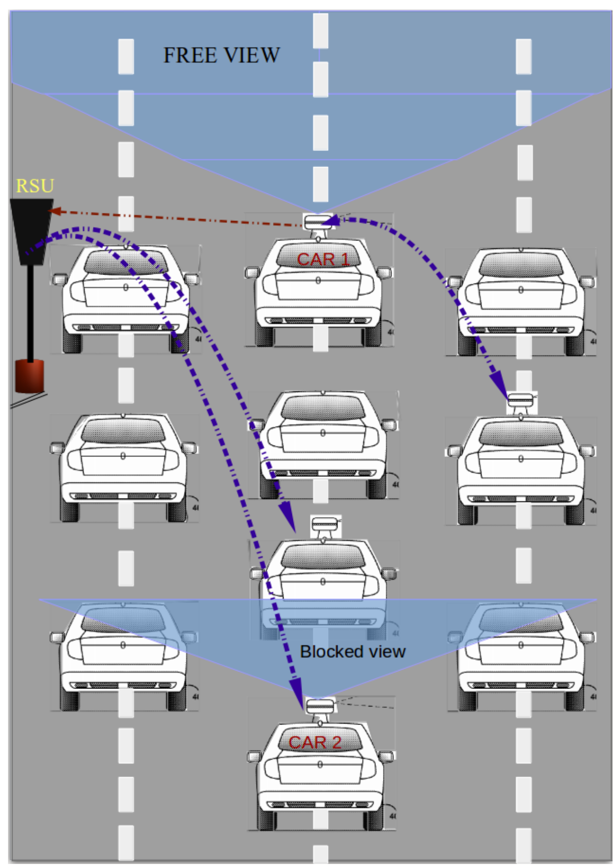

Fig. 3. Sharing of Compressed LiDAR data between vehicles directly or via a Road Side Unit (RSU)

	Latency (ms)		BandWidth(MBps)	
	PCD	TXT	PCD	TXT
Test case1	1.80	1.50	1.95	2.18
Test case2	1.72	1.51	1.46	2.24
Test case3	1.52	1.13	1.45	2.59
Average	1.68	1.38	1.62	2.33

TABLE II
TRANSMISSION RESULTS FOR STATIC CASE

	Latency (ms)		BandWidth(MBps)	
	PCD	TXT	PCD	TXT
Test case1	1.92	1.26	1.61	1.51
Test case2	1.39	1.87	2.81	2.47
Test case3	1.57	1.57	1.86	1.59
Average	1.62	1.56	2.09	1.85

TABLE III
TRANSMISSION RESULTS FOR DYNAMIC CASE

IV. RESULTS AND ANALYSIS

We performed the compression for KITTI dataset [9] and Ouster data (available online) which was captured with 64 channel LiDAR. The same was also done for LiDAR data captures at IIT Hyderabad (IITH) campus with 16-channel LiDAR. The Point clouds were captured at IITH by using three different LiDAR modes (512*10,1024*10,2048*10) so as to have 32768 or 65536 or 131072 points respectively. The compression results are summarized in table I. The KITTI and Ouster data used are taken in more standard conditions and are having a reasonably good number of non-zero points. IITH data captured is having a large number of zero points (those points having x, y and z components as zero). The percentage of non zero points out of a total number of points in IITH data are found to be too low (11-16%).

We have also done latency and bandwidth analysis of both compressed and uncompressed data captured at IITH. 'Qperf' is the monitoring tool used for this analysis. It gives detailed statistics of the network. The test cases were run for static and dynamic cases. The transmission analysis results are summarized in tables II and III. The data was continuously transmitted. Each test case refers to the latency and bandwidth calculated and averaged over a duration of 60 seconds. We can see from the results that latency decreases for the transmission of a text file as compared to pcd files (uncompressed). Comparison between the original Point cloud and its decompressed version for 16-channel LiDAR captured at IITH campus is shown in figure 4.

V. CONCLUSION AND FUTURE WORK

This paper shows the analysis of a set up for Point cloud compression and transmission for ITS. The preliminary results show a minimum 65% decrease in the data size, as we transmit encoded text files instead of pcd files. With respect to transmission, latency is found to be decreasing. The set up can be quite useful in sharing the Point clouds captured by a particular vehicle, among other vehicles. Thus the set up can add to the concept of connected vehicles which is an essential requirement for ITS. In the future, we also plan to analyze transmission using other networks like 3G and 4G. Also, a thorough analysis is to be done for the case where both the transmitter side and the receiver side is moving, which is generally the case in the vehicle transportation environment. We can also use this method at RSU's. RSU can also have the LiDAR which can capture the traffic congestion at the junction. This LiDAR data can be shared with the vehicles after compression.

ACKNOWLEDGMENT

This work was supported by the project M2Smart: Smart Cities for Emerging Countries based on Sensing, Network and Big Data Analysis of Multimodal Regional Transport System, JST/JICA SATREPS, Japan.

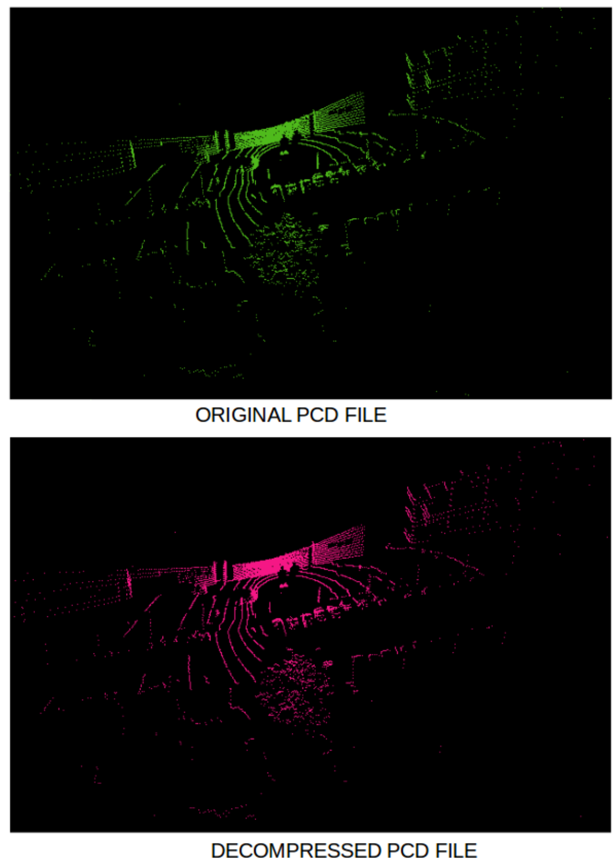

Fig. 4. Original and Decompressed Point clouds taken at IITH

REFERENCES

[1] Jang, G. R., Shin, Y. D., Park, J. H., & Baeg, M. H. (2014, October). Real-time point-cloud data transmission for teleoperation using H. 264/AVC. In Safety, Security, and Rescue Robotics (SSRR), 2014 IEEE International Symposium on (pp. 1-6). IEEE.

[2] Chen, Yilin. "Real-time Point cloud data transmission in Kinect Point cloud data based telerehabilitation system." PhD diss., 2015.

[3] Kammerl, J., Blodow, N., Rusu, R. B., Gedikli, S., Beetz, M., & Steinbach, E. (2012, May). Real-time compression of Point cloud streams. In Robotics and Automation (ICRA), 2012 IEEE International Conference on (pp. 778-785). IEEE

[4] Fan, Y., Huang, Y., & Peng, J. (2013, October). Point cloud compression based on hierarchical point clustering. In Signal and Information Processing Association Annual Summit and Conference (APSIPA), 2013 Asia-Pacific (pp. 1-7). IEEE.

[5] Morell, V., Orts, S., Cazorla, M., & Garcia-Rodriguez, J. (2014). Geometric 3D point cloud compression. Pattern Recognition Letters, 50, 55-62.

[6] Meagher, D. (1982). Geometric modeling using octree encoding. Computer graphics and image processing, 19(2), 129-147.

[7] Botsch, M., Wiratanaya, A. and Kobbelt, L., 2002, July. Efficient high quality rendering of point sampled geometry. In Proceedings of the 13th Eurographics workshop on Rendering (pp. 53-64). Eurographics Association.

[8] Peng, J. and Kuo, C.C.J., 2005. Geometry-guided progressive lossless 3D mesh coding with octree (OT) decomposition. ACM Transactions on Graphics (TOG), 24(3), pp.609-616.

[9] @ARTICLEGeiger2013IJRR, author = Andreas Geiger and Philip Lenz and Christoph Stiller and Raquel Urtasun,title = Vision meets Robotics: The KITTI Dataset, journal = International Journal of Robotics Research (IJRR),year = 2013