LIDAR Data Compression Challenges and Difficulties

Mohamed M. Abdelwahab
Faculty of Engineering, ECE Dept.
Zagazig University, Zagazig, Egypt
E-mail: wahabmaksoud1976@gmail.com

Walid S. El-Deeb
Faculty of Engineering, ECE Dept.
Zagazig University, Zagazig, Egypt
E-mail: wseldeeb@ucalgary.ca.com

Aliaa. A. A. Youssif
College of Computing and Inf. Technology
AASTMT, Cairo, Egypt
E-mail: aliaay@yahoo.com

***Abstract*—Airborne light detection and ranging (LIDAR) system uses in many field such as military, agriculture, geology and transportation. LIDAR data acquisition consumes several gigabytes from the storage capacity of the system. So, it's necessary to reduce the on board storage capacity as well as its transmission rate. Compression of LIDAR data means reducing the amount of data required to represent the digital data acquired by LIDAR system. This is done by removing data redundancies to achieve compression. In this paper we explain LIDAR system theory and component, compare between lossy and lossless compression techniques then we discuss LIDAR data compression techniques in order to choose a suitable one for LIDAR system that gives high compression ratio with suitable processing time. In most LIDAR data compression techniques classification of LIDAR data is an important issue as a preprocessing step to simplify the data. Using more than one compression algorithms increase the compression ratio but increase the processing time as well.**

Keywords-LIDAR; compression algorithms; lossless; lossy; 3D point cloud

I. INTRODUCTION

LIDAR is an active optical sensor used to collect 3D information from an object surface by using laser pulses **[1]**. These technology starts at 1960 and LIDAR terrain mapping starts 1970.

A pulse of light is transmitted at a sampling rate greater than 150 kHz and the reflected pulse is detected with the record of the travelling and return time. Time interval can be measured with a precision of 67 ps (corresponding to 1 cm range precision) [1]. By the constant speed of light, the range can be calculated with very high resolution rather than other technologies. Airborne LIDAR platforms are used to produce digital elevation models (DEM) for large areas [2].

LIDAR system consists of: high-precision clock, Global Positioning System (GPS), inertial navigation Measurement Unit (IMU), data storage, management systems and GPS ground station [3].

LIDAR system has the advantage of: higher density, very high Accuracy, little time for collecting data processing and acquisition, it doesn't need a lot of ground control station and the collected data are in digital format which simplify processing.

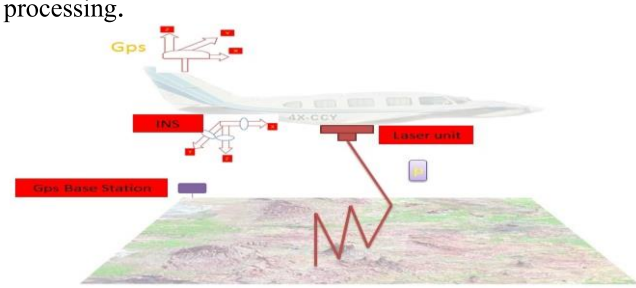

Figure 1. Schematic diagram of airborne LIDAR performing line scanning.

The basic idea of LIDAR system, shown in Figure 1, is to measure the time that the laser pulse takes to strike an object and return to the sensor, determining the distance using the travel time, record the laser angle, and then compute where the reflecting object (e.g., ground, tree, car, etc.) is located in three dimensions. Additional technologies were added to achieve high accuracy, such as GPS, Inertial Measuring Units IMU and Inertial Navigation Systems INS [4].

Aground station with known position was used to correct the GPS position which was recorded by the plane. Newer systems were more compact, lighter, have higher angular precision, and can process multiple laser returns in the air as shown in Figure 2. (i.e., a second laser shot is emitted before returns from the previous laser shot are received) [5], this increases the collected data by 30% and increases the ability to track target in three dimension. The data collected by LIDAR system had the following Types:

- Points: LIDAR data almost be in point format which stored in LAS file, LIDAR data contains not only x, y, z points but also the return intensity points.
- Digital Elevation Models (DEMs): it was derived from the Point data to represent bare earth and used to create contours.
- Contours: Contours were commonly available in vector formats (e.g., .shp, .dxf) and are most frequently derived from a pre-constructed DEM or TIN model

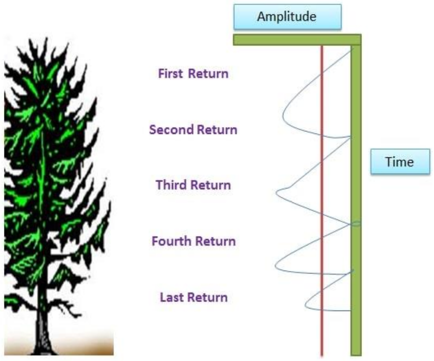

Figure 2. Multiple return pulses detection.

LIDAR system appears in many application: creating and updating the flood plain mapping for specific area, forest and tree Studies, coastal change mapping, 3d city modeling, volumetric calculation and military application (3d battle field visualization-preventing and combating crime).

II. DATA COMPRESSION

Compression: means reducing the size of data to save space when storing and save time when transmitting. This is done by removing redundancies from the input data [6]. The generation of information is formed by a probabilistic process that can be measured in a manner with the intuition [7].

Entropy:

$$H(S) = \sum p(s) \log_2 (1/p(s)) \tag{1}$$

where:

S: is a set of possible states, and

P(s): the probability of state.

Self-information of a message represented by the number of bits which contained in the message is defined by:

$$I(s) = \log_2 (1/p(s)) \tag{2}$$

NOTE: message with high probability contains less information [8].

A model of a general compression system block diagram is illustrated in Figure 3. It consists of an encoder capable of transforming the input data called (x) into a data set called (d). This data set is then transmitted through the channel to the decoder where a reconstruction (x^-) of the original data was produced [6].

III. COMPRESSION ALGORITHMS

Lossless compression is called error free compression. It reconstructs the data exactly as the original one from the compressed data set with loss about (5-25) % of the original data with very good quality [6]. Most Lossless compression techniques seldom achieve higher compression ratio.

Lossy compression reconstruct an approximation to the original data with large data loss about (50-90) % of the original data set [6]. This results in significant reduction of the file size with quality loss, can achieve much higher compression ratio but at the cost of introducing artifacts and distortions.

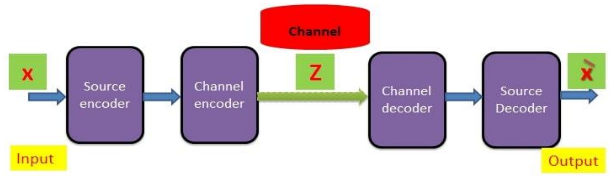

Figure 3. Block diagram of general compression model.

A. *Lossless Compression Algorithms*

1) Huffman coding

It is an optimal prefix code that generated from the probability of a set of information. Starts with a forest of trees, one for each message, repeat until only single tree remains, and select two trees with lowest weight root and combine them. In Huffman coding, the average number of bits per pixel can be reduced by assigning binary codes of different bit length to various image intensities. Hence, short code words are assigned to the intensity levels of high probability of occurrence and longer code words to less frequent intensity levels.

2) Arithmetic coding

The coding process begins by defining an arbitrary range of values. For convenience this range will be defined as (0, 1) [8]. The working range is then subdivided into *n* subintervals, where *n* is the total number of symbols in the input stream, such that each symbol is assigned one subinterval. The width of each subinterval is dependent on the probability of the symbol that is mapped to it. Therefore, if the probability of symbol *A* is twice that of symbol *B* then the width of the subinterval assigned to symbol *A* should also be twice that of symbol *B*. To calculate the width of each subinterval, the following formula is used:

$$W_I = P_i \times W_{range} \tag{3}$$

where:

W_I: is the width of the subinterval for the i^{th} symbol, P_i: is the probability of that symbol,

W_{range}: is the width of the current working range.

After subsequent encodings of n number of symbols with the same series of steps, we choose any number within the working range and it will be represented by the arithmetic encoded data.

The compressed code was represented by a single fractional number instead of a series of discrete symbols or codes. In order to recover the original ensemble, the decoder must know the model of the source used by the encoder and a single number within the interval determined by the encoder.

TABLE I. HUFFMAN'S ALGORITHM PSEUDO CODE

Number	Procedure
01	Input: LIDAR Data
02	Output: compressed data
03	Begin
04	Create a descending probability column of intensity.
05	Sum the node of minimum probability to perform intermediate node.
06	Repeat until all branches of intensity levels are added and the probability=1
07	Assign "0" to the top branch, assign"1"to the bottom branch
08	09:repeat until all tree leaves are reached

09	Create the code word of each intensity levels from root to specific path.
10	End.

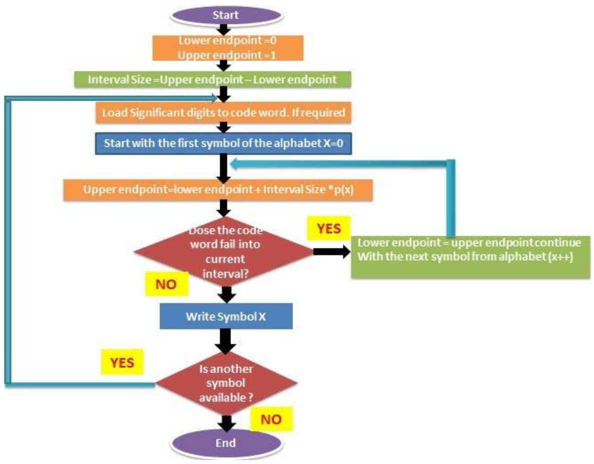

Figure 4. Arithmetic coding flow chart.

3) Lempel Ziv-Welch (LZW) code

LZW is represented as another category of lossless compression techniques known as dictionary coder in which the coding techniques relay upon the observation that there are correlations between parts of data [8]. The technique starts by building a dictionary of pervious seen strings and code groups of characters of varying length. Given a position in a file and look to the preceding part to find the longest match to the string. Then give a code referred to the match. LZW depends on sliding window with fixed distance back from the current position. The basic idea is to replace those repetitions by (shorter) references to a "dictionary" containing the original.

B. Lossy Compression Techniques

Lossy encoding is based on the concept of compromising the accuracy of the reconstructed image in exchange for increased compression. A lossy method should be designed such that the balance between the information loss and the compression ratio can be achieved [9].

1) Transform coding

In this technique, the original image is divided into sub-images of size, *n* x *n* for independent processing [10]. The resultant sub-images are then transformed into frequency domain using reversible, linear transformation. The transformed sub-images are then quantized.

Figure 6 illustrates the coding procedure of the transform encoding [10]. In the transformation operation the inter pixel redundancy can be reduced by quantizing the frequency coefficients.

The encoding process terminates by coding (normally by using a variable length code) the quantized coefficients. In general both the level of compression and computational complexity increase as the sub-image size increases, the most popular sub-image sizes are 8 X 8 and 16 X 16.

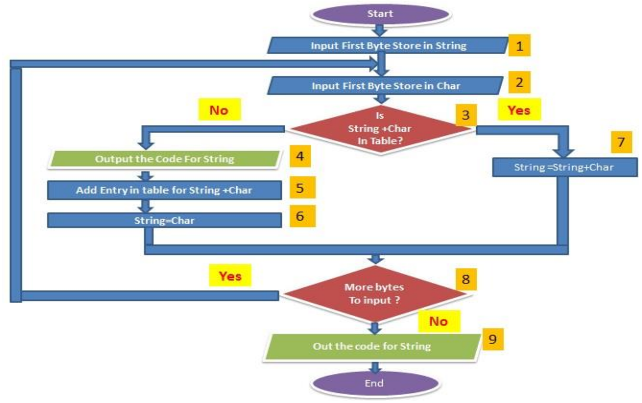

Figure 5. Lempel Ziv-Welch *(*LZW*)* flow chart.

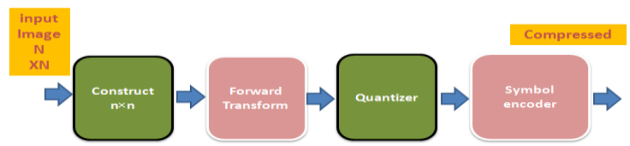

Figure 6. Functional block diagram of the transform encoding.

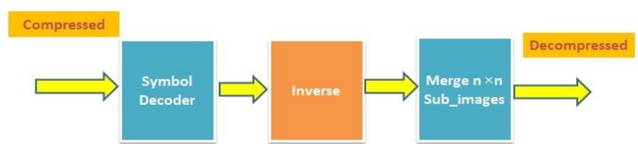

Figure 7. Functional block diagram of the transform decoding

Figure 7 illustrates the decoding procedure of the transformation decoding [8]. The compressed image data of a sub-image ($n \times n$) is decoded by a suitable decoder (the inverse of the used decoder). The decoded image data is then inverse transformed using one of the common used inverse transformations such as IDCT or IDWT and then the transformed *n* x *n* sub-images are merged to form the decompressed image.

2) Fractal coding

Fractal compression is considered as a lossy compression method for digital images, based on fractals. The method is best suited for textures and natural images, relying on the fact that parts of an image often resemble other parts of the same image [10]. Fractal image coding has the advantage of higher compression ratio, but is a lossy compression scheme. The fractal image compression technique partitions the original image into two different block sizes called the range and domain block. The best match between range and domain block is known as the transformation mapping. The image is reconstructed using iterative functions and inverse transforms [8].

- Fractal coding will be poor quality due to low PSNR Values [8].
- Fractal applications are mainly focused on two aspects [10]. Applying fractal dimension and signature as features in image segmentation, texture analysis and Iterated Function System (IFS) into image compression.

In Fractal Block Coding (FBC). An image is regularly segmented into sets of 8x8 blocks, called range blocks, and then for each range block, finding a contractive mapping and domain block. FBC is a lossy coding technique, and similar to vector quantization.

IV. LIDAR Data Compression Techniques

A. Hybrid Lossless Compression Method Based on Improved LZW and Huffman

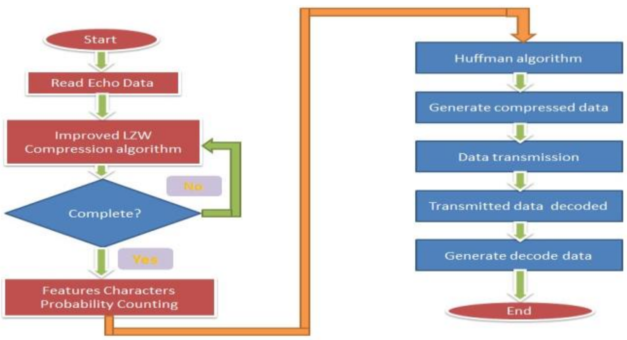

Figure 8. Data compression and decompression with hybrid LZW and Huffman.

It is a hybrid lossless compression, which based on the improved LZW (Lempel-Ziv-Welch), which combines Huffman coding. The compression process starts with receiving the reflected pulses, compressed by the improved LZW algorithm, state the probability of the feature characters, coding these data with Huffman code and transmit to the ground.

The decompression is done by reconstructing the data with Huffman and LZW decoding algorithms. The compression ratio of this approach is more than twice of the LZW compression algorithm with small increase of compression time [11].

In LZW algorithm most of the time is consumed in searching and matching to the dictionary so if we find an efficient approach to reduce this time it will results in low processing time. Reducing the characters when establishing the dictionary by removing unnecessary data. This reduces the searching time. Also using another stage of compression by Huffman coding improves the compression ratio.

B. Compressing LIDAR Data by Arithmetic Coding with Distance-Based Predictor

Data (3D information) from LIDAR system will be compressed by using a predictor of incoming point from the previous information. The compression is done by using the arithmetic algorithms. The predictor is used to discover the relation between two adjacent point's. Data preprocessing were performed by extracting the coordinate and the color data then, splitting LIDAR data into chunks. Distance-based predictor was used to reduce entropy as the point coordinate in chunk file range from 0 to 60000. Encode the coordinate values directly is not a wise choice [12]. Because the number of coordinate symbols is 60000, the entropy encoder will not work well.

The principle of coding algorithm is to encode the first point coordinate directly and encode the interval value of x, y, z instead of the coordinate value for the forthcoming points. The compression ratio of mobile vehicle LIDAR data is 16.1% and compression ratio of airborne LIDAR data is 26.27% [8]. The average compression speed is 563 million per second. And the average decompression speed is 449 million per second [12]. This method gives continuous compression even if we lost the point because of the usage of the distance predictor. But the compression ratio is still not good enough. Also, the processing time of compression and decompressing is not short.

C. LIDAR Data Compression Based on Geometry

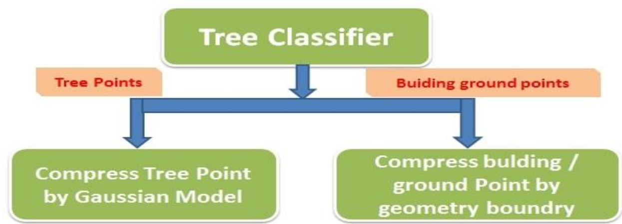

Figure 9. High level process flow of the geometry based compression system.

The compression of LIDAR data is done by classifying the data depending on its geometric property, which gives not only a good compression ratio but also a classification of the data. The compression process consists of the generation of the depth image for 3D tree and building/ground data respectively, compress them by JPEG2000 [9].

At first, LIDAR data is classified into tree and non-tree points; different compression techniques depends on its geometry are applied to different type of data [13]. This method improves the compression ratio with low processing time

TABLE II. Encoding a Target Points Procedure

Number	Procedure
01	Project 3D points into a 2D depth image.
02	Apply height-based clustering to cluster buildings.
03	Use least square fitting to fit each cluster of pixels into a plane.
04	Find the boundary pixels for each building cluster.
05	Simplify the boundary by line fitting.
06	Compress the boundary pixel location and plane parameters by arithmetic encoder.

D. Compressive-Sensing of 3D Point Clouds

We analyze 3D point cloud data compression using compressive-sensing. This is done by examine different error sources which affect the system. Excluding edge points from error calculation gives us better criteria to decide the best compression ratio in the system [14]. The maximum error from compressive sensing is found at the edge point of the cloud point of LIDAR data. So, excluding the edge point from error calculation improves the compression ratio.

E. LIDAR Data Compression with Quad Tree Structure Preprocessing

Data collected by LIDAR system is very large which made difficulties in storage. However, if it is small; the interpolation method is not suitable for surface generation. The compression algorithm depends on quad tree structure for fast preprocessing which simplify the LIDAR data set [15]. The compression system is begun by building a point region quad-tree and compressing this structure with a threshold [15].

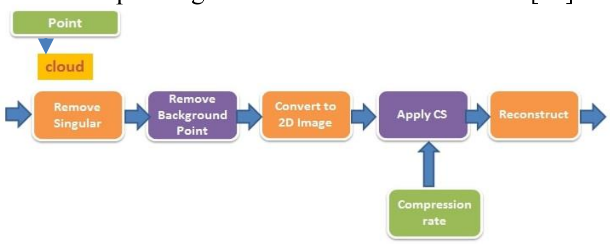

Figure 10. Framework for compressive-sensing of 3D point clouds.

TABLE III. PSEUDO CODE LIDAR DATA COMPRESSION WITH QUAD TREE STRUCTURE PREPROCESSING

Procedure data compression
Input: LIDAR data
Output: compressed quad-tree (Q).
Begin
Create a normal quad tree
Insert all the leaf nodes of quad tree (Q) into a linear list, L.
While L is not empty do begin
Pick a node m from L
Find the edge neighbors of m
Calculate the height difference with the neighbor of m
if the abs (height difference)<threshold, then
Average the location coordinate and height value between m and the neighbor m
Store the new node to L
Discard this node and merged the neighbor of m
Else pass the node
End while
Return quad-tree

In this system an accurate collection of boundary points are introduced with adjustable threshold that gives the opportunity to extract the correct shape of the object.

F. Lossy Compression Technique of LIDAR Waveform

The recorded echo pulses from LIDAR system always called waveform data. From these reflected pulses we can analyze the properties of the surface based on its material signature. Another method of solving the problem of storing and classifying the data is to use a numerical method including DWT and Kohonen's Self-Organizing Map [17].

Using the Discrete Wavelet Transform has two advantages: first, it is an efficient tool to compress waveform data, and second, the wavelet coefficients describe the shape of the echo pulse, and, therefore, they can also be used for classification.

The steps of a transform-based lossy compression technique are the following:

- Preprocessing of the original data. (e.g., partitioning An image into smaller blocks).
- Transformation (e.g., using discrete cosine Transformation, DWT, etc.).
- Quantization of the coefficients.
- Lossless compression of the coefficients is used: RLE encoding then Huffman coding.

Because wavelet transform is a good compressive tool, also the wavelet coefficient can describe the shape of echo pulses [4]. The compression algorithm is based on DWT and the coefficient is used for classification. The compression method consists of using DWT as a preprocessing step then uses one of the lossless compression algorithms such as Huffman and run length encoding. With this method 18.7% compression rate has been achieved (i.e., the data was compressed to less than one-fifth of its original size) [18]. Compression ratio about 80%. Using this lossy compression technique results in high compression ratio but decreasing the quality of the reconstructed data.

G. Speed up Lossless LIDAR Data Compression

Huge amount of data could be acquired by LIDAR system, which caused a problem in processing time. Most of researches concentrate on compression ratio but this technique stress on improving the time efficiency. Because the point cloud of LIDAR can be coded independently, so it can be processed simultaneously. Multithreading scheme is used (i.e. multi core central processor and direct memory access).The overall compression time has been reduced over 70% [19].

These methods prove that the use of multi-core processor and DAM can speed up the LIDAR data compression technique. The compression time is decreased.

The compression algorithms will be summarized as follows: [20].

Encoding the reflected points by using predictive coding, Use variable length code to encode the error in prediction and finally the arithmetic code is used.

V. CONCLUSION

Compressing LIDAR data became an urgent process because of increasing the amount of collected data from very high resolution LIDAR system. Some of the previous techniques reduced the collected data by removing the unnecessary data caused high compression ratio but this affecting the quality of the reconstructed data. Another technique applied two compression algorithms LZW then Huffman, this gave high compression ratio but the processing time was increased. Using lossy compression technique will provide high compression ratio up to 80% but with low quality of the reconstructed data (or images). Sometimes the LIDAR system may loss location in real time scanning; this caused a problem but using distance-based predictor will overcome this problem and provide continuous compression but insufficient CR. Classifying the input data simplify processing. Excluding the edge point from error calculation could improve the CR. The difficulty and challenges is to find a suitable compression technique that gives high compression ratio to save the storage capacity but this will take large processing time due to the large amount of input data .So our challenge is to design a compression technique to deal with this large data to provide high CR with small processing time. This is done by classifying the input data as a preprocessing step to simplify dealing with this large data .Then using two compression algorithms one lossless and the other is lossy will increase CR. Then using multithreading scheme will speed up processing.

REFERENCES

[1] Robert Burtch. "LIDAR Principle and Applications". IMAGIN Conference, Traverse City, MI. 2002.

[2] Ahmed Kotb, Safaa Hassan. "A Comparative Study Among Various Algorithms for Lossless Airborne LIDAR Data Compression" National Authority for Remote Sensing and Space Sciences Cairo, Egypt, 14th(icenco), 2018.

[3] Guy Eblelloch, Carnegie."Introduction to Data Compression". Mellon University. Jan. 2013.

[4] (NOAA) Coastal Services Center "An Introduction to Lidar Technology, Data, and Applications." National Oceanic and Atmospheric Administration, November 2012.

[5] ZhaoLijiana, B, LaiZulonga LiYingchengb XueYanlib LiaoMingb, "Application And Analyses Of Airborne LIDAR Technology In Topographic Survey Of Tidal Flat And Coastal Zone, The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences. Vol. XXXVII. Part B3b. Beijing 2008.

[6] Emmanul C., Ifeacher Barre, W.Jervis. "Digital Signal Processing a Practical Approach". Addison-Wesley Publishing Company, Third Edition.2010.

[7] Jamie Carter, Keil Schmid, Kirk Waters, Lindy Betzhold. "An Introduction to LIDAR Technology, Data, and Applications". National Oceanic and Atmospheric Administration (NOAA), 2017.

[8] Rafal C. Gonzalez, Ritchard E. Woods. "Digital Image Processing". Third Edition, November 2012.

[9] YiKun Zhang, Xiao Li, DengXin Hua, Hao Chen, HaiYan Jin Xi'an. "A LIDAR Data Compression Method Based on Improved LZW and Huffman Algorithm". 2010 International Conference on Electronics and Information Engineering (ICEIE 2010).

[10] Mohamed M Abd Elwahab, Satellite Image Compression, Faculty of Engineering, Cairo University, February 2013.

[11] Xian Xiong Liu, Yue Wang. "A Scan Line Based Data Compression Approach for Point Cloud: Effective and Lossless" .Fourth International work shop on eo and RSA, 2017.

[12] Xiaoling Li, Wenjun Zeng, Ye Duan. "Geometry Based Airborne LIDAR Data Compression". icme 2013.

[13] Vahid Behravan, Gurjeet Singh. "A Framework for Compressive-Sensing of 3D Point Clouds". 12th International Conference on Computational Intelligence and Security, 2016.

[14] A.V. Levenets, I.V. Bogachev, En UnChye. "Telemetry Data Compression Algorithms Based On Operation of Displaying onto Geometric Surfaces". International Siberian Conference on Control and Communications (SIBCON), 2017.

[15] Ruoyu Du, Hyo Jong Lee. "A Novel Compression Algorithm for LIDAR Data". 2012 5th International Congress on Image and Signal Processing (CISP 2012).

[16] ChenxiTu, Eijiro Takeuchi, Chiyomi Miyajima, Kazuya Takeda. "Continuous. Point Cloud Data Compression Using SLAM Based Prediction". IEEE Intelligent Vehicles Symposium (IV) June 11-14, 2017, Redondo Beach, CA, USA, 2017.

[17] C. Toth, S. Laky, P. Zaletnyik, D. Grejner-Brzezinska, "Compressing And Classifying LIDAR Waveform Data". Budapest University of Technology and Economics, Muegyetem RKP.3, Budapest, H-1111, Hungary, 2012.

[18] Domen Mongus, Denis ˇSpeliˇc, Borut ˇZalik. "Multithreaded Approach for Lossless LIDAR Data Compression". Smetanova Ul. 17, SI 2000 Maribor, Slovenia.2012.

[19] Joe Liadsky. "Introduction to LIDAR". NPS LIDAR Workshop May 24, 2007.

[20] Guodong Wang, Xiaojian Liu. "Parallel Image Processing Platform Based on Multi-core DSP".ICIS, 2017.