

Comparison of Algorithms for Lossless LiDAR Data Compression

Domen MONGUS, Bojan RUPNIK and Borut ŽALIK

The GI_Forum Program Committee accepted this paper as reviewed full paper.

Abstract

Contemporary Airborne Light Detection and Ranging (LiDAR) systems are capable to rapidly gather the data from large geographical areas with high precision and great density. As a result, obtained datasets can contain several tens of millions of points, making LiDAR data compression an important issue. In this paper, three domain-specific compression algorithms are compared against a general-purpose algorithm. Selected testing LiDAR datasets are derived from the practice to challenge common data compression issues. In this way, influences of the terrain type, point density, and number of contained points on the compression efficiency are studied.

1 Introduction

In the past decade, Light Detection and Ranging (LiDAR) has become one of the prime remote sensing technologies (LIU 2008). LiDAR systems are active sensor systems that use a short wavelength laser light to rapidly obtain information about the distant objects with high precision and great density. The range is obtained with measuring time delay between transmission of the laser pulse and detection of its reflection (MANUE 2008).

Different types of LiDAR systems exist. For spatial data acquisition, airborne LiDAR systems are used most frequently. In this case, the LiDAR data scanner is mounted on an aircraft, from where it records the Earth's surface, as shown in Figure 1. Several supplementary sensor systems are used to georeference the data. Inertial measurement unit (IMU) is used to establish an angular orientation of the sensor system by measuring the roll, pitch and heading of the aircraft (MANUE 2008). Additional scan angle measurement defines an angular orientation of each emitted laser pulse that allows mapping the range measurement into 3D point coordinates. Finally, global positioning system (GPS) is used to define the position of the LiDAR scanner that is used to project the point coordinates into a local (e.g. Gauss-Krüger coordinate system), or a global geographic coordinate system (e.g. Universal Transverse Mercator coordinate system). Additionally, many systems include a digital camera to capture photographic imagery of the terrain that is being scanned to obtain points' colours.

Today, airborne LiDAR systems are capable of executing over 150.000 measurements per second, where achieved density exceeds 10 points per square meter (MANUE 2008). Additionally, they are able to distinguish between different reflections of a single emitted

laser pulse that allows them to penetrate through vegetation coverage and record the terrain under it. However, this results in a huge datasets that may contain several tens of millions of points. They are usually stored in LAS files, an open standard binary file format proposed by the American Society for Photogrammetry & Remote Sensing (ASPRS 2009). The main aim of the LAS format is to assure the standard exchange of captured LiDAR data. Although the LAS file format prescribes different point record types, at least 160 bits per point are used. Consequentially, LAS files may consume more than a gigabyte per square kilometre. In practice, this presents considerable problems. Expensive storage, difficult distribution to the users and time consuming exchange over the internet are just some of the reasons why LAS data compression has recently become an important issue.

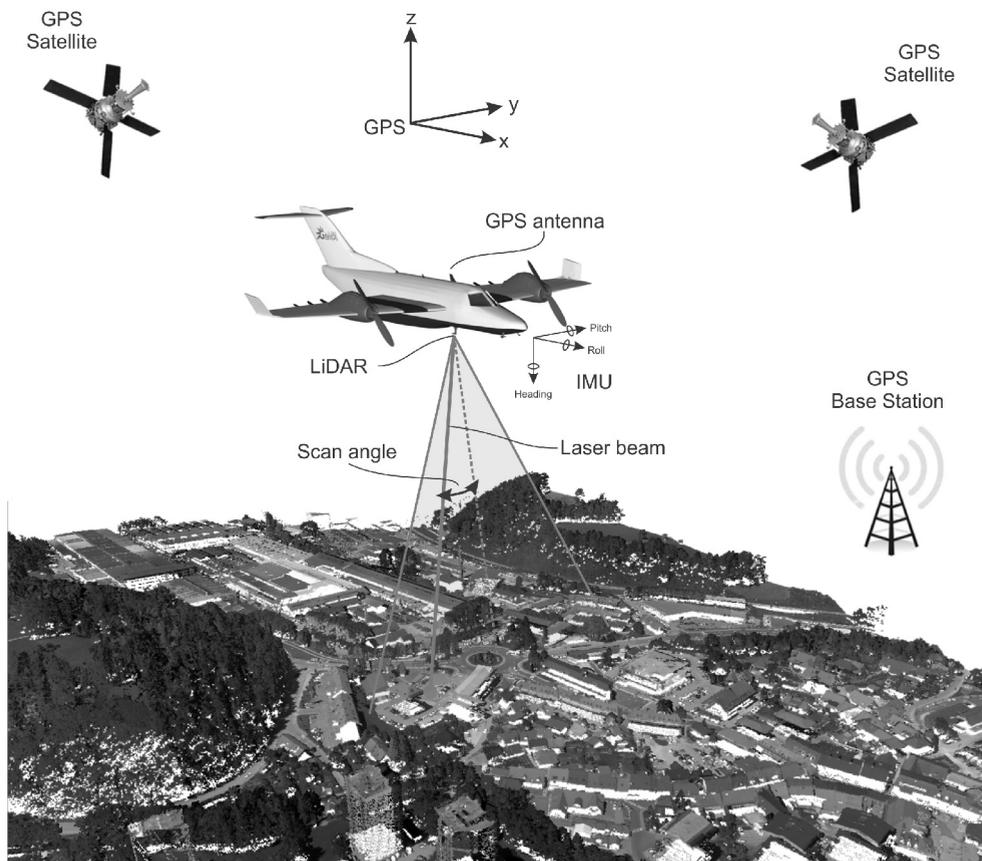


Fig. 1: Airborne LiDAR data acquisition

In this paper, a comparison between a general-purpose compression algorithm and three domain-specific compression algorithms is presented. Approaches for geometrical (LiDAR) data compression are explained in Section 2. In Section 3, the results of testing algorithms are presented, while Section 4 concludes the paper.

2 Algorithms for LiDAR data compression

Data compression is among the oldest disciplines in computer science, and is highly related with the information theory and the pioneering Shannon's works in late 1940s and 1950s (SHANNON 1948; SHANNON 1951). A huge number of various compression algorithms have been developed since then (SALOMON 2006). The most general classification of data compression methods distinguishes between lossless and lossy methods. After decompression, lossy methods cannot retrieve the original data. Because of this, they are usually applied on multimedia data (e.g. digital images, audio, and video). The loss of information is practically not detected due to the imperfection of human senses. On the other hand, many applications cannot afford the losses in the data compression process (e.g. text, medical data, engineering, and scientific data), and therefore, lossless algorithms have to be applied. The lossless compression methods are based either on dictionary-based approaches (e.g. LZW proposed in WELCH 1984) or on statistic methods (e.g. Huffman coding (HUFFMAN 1952), arithmetic coding (RISSANEN & LANGDON 1979)). In practice, these methods are frequently combined in various packages (such as PKZIP, RAR, etc. widely used today). However, these methods are general-purposed and do not employ application-specific knowledge about the data. For compression of specific data, the characteristic patterns can be exploited to predict the composition of the forthcoming data, which can be compressed with considerably greater efficiency. Such methods have been developed, for example, for triangular meshes (ISENBURG et.al. 2005), voxel data (KLAJNŠEK & ŽALIK 2005), and XML files (LUOMA & TEUHOLA, 2007). Airborne LiDAR data are gathered according to the flight plan and regular movement of the laser beam (MANUE 2008). Therefore, it can be expected that an efficient prediction model can be developed. In continuation, a brief description of the only publically available algorithm among the tested ones is given (details can be found in MONGUS & ŽALIK (2011)).

In Figure 2, a schema for LiDAR data compression is shown, consisting from three consecutive steps:

1. Points are encoded with a predictive coding scheme.
2. Prediction errors are coded with the variable-length-coding (VLC).
3. VLC values are compressed with the arithmetic coder (AC) and stored in the output file.

In the predictive coding model, the history of points is analysed to predict the position and associated scalar values of the next point in the data stream. Since the same predictions can be obtained during the decoding process, only prediction errors need to be stored. Thus, accurate predictions lead to a high reduction of absolute values that can be stored more efficiently. However, different prediction rules are used due to different nature of the attributes. The constant prediction rule presumes that the next attribute value is the same, as the current one. Thus, it is efficient for coding attributes that rarely changes (e.g. scan direction flag, user data, point source ID, and number of returns). On the other hand, some of LiDAR point attributes are never the same for two successive points (e.g. GPS time of recording). In this case, linear interpolation of previous values is used to estimate the prediction for the next point attribute value. Thus, linear predictive coding is highly accurate when values change for a constant quantity. Unfortunately, LiDAR points are usually not uniformly distributed and more complex prediction rules must be used to achieve adequate accuracy. MONGUS & ŽALIK (2011) presented an accurate prediction rule

for estimating the points' positions that exploits geometric correlation arisen from LiDAR data scanning.

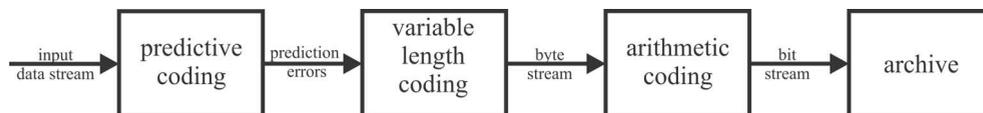


Fig. 2: A schema for lossless LiDAR data compression

Errors in the prediction can be stored with fewer bytes and therefore, they can be represented with a variable-length code (VLC). Four description bits are assigned to each error value. The description bits define the sign and the length (in bytes) of each value. The error values (they are usually small) are therefore stored with the minimal number of bits. The stream of error values bits and the associated description bits are finally compressed by AC.

3 Results

The efficiency of three domain-specific compression algorithms (LASCOMPRESSION 2011; LASZIP 2011; LiDAR COMPRESSOR 2011) and a general-purpose open-source compression algorithm (7-ZIP 2011) have been evaluated. The testing datasets were carefully selected and consisted of LAS files of various sizes, point density, and representing most characteristic terrain types. Each terrain type is presenting different problems for compression algorithms.

Table 1: Testing datasets for comparison of compression algorithms

File	File Size (bytes)	Terrain Type	Number of Points	Density (points/m ²)	Bits per point
1	952.000.229	Flat	34.000.000	0,925	224
2	440.056.349	Flat	15.716.290	0,881	224
3	104.088.465	Flat	3.717.437	0,890	224
4	812.916.053	Hill	29.032.708	5,317	224
5	386.589.873	Hill	13.806.773	4,236	224
6	65.734.205	Hill	2.347.642	6,890	224
7	429.180.001	Watered	15.327.849	16,540	224
8	988.974.481	Watered	35.320.509	0,679	224
9	375.167.329	Watered	13.398.825	14,110	224
10	104.639.368	Urban	3.737.102	0,413	224
11	537.088.209	Urban	20.657.230	30,296	208
12	179.510.973	Urban	6.411.098	0,900	224

Flat terrain present the most basic problem, which is very promising for compression, since a high level of redundancy is contained (e.g. the points are almost aligned with the grid and

the heights are similar). Because compression algorithms reduce storage requirements by removing the redundancy within the data, high level of efficiency is expected in this case. On the other hand, hilly terrains (Figure 3) are less favourable for data compression mainly because the point density is inconsistent throughout the files. Point density is depended on the exposure of the slope to the LiDAR scanner, where the density on exposed slopes is significantly higher than in case of shaded slopes. Because of this, points are not regularly distributed and their heights distinctively differ. Less accurate prediction schemas are expected in this case.



Fig. 3: Hilly terrains present greater problems for compression because of inconsistencies of point density in the data set

A similar problem is noticed in the case of urban areas (Figure 4), since buildings disrupt the even height arrangement of points. In addition, height differences between nearby points may vary because of a large number of different sized objects (e.g. cars, buildings). Because each object causes a discontinuity in point distribution, the prediction schemas are challenged. Another type of difficulties for compression algorithms may cause datasets containing weak reflections (Figure 5) as a consequence of scanning watered areas (e.g. rivers, lakes). In this case, the reflections are not properly detected, causing large gaps in point distribution presenting further difficulties to efficiently predict the positions of the upcoming points.



Fig. 4: Urban areas cause abrupt differences in point distributions, as well as great variations of point height, making the prediction less effective

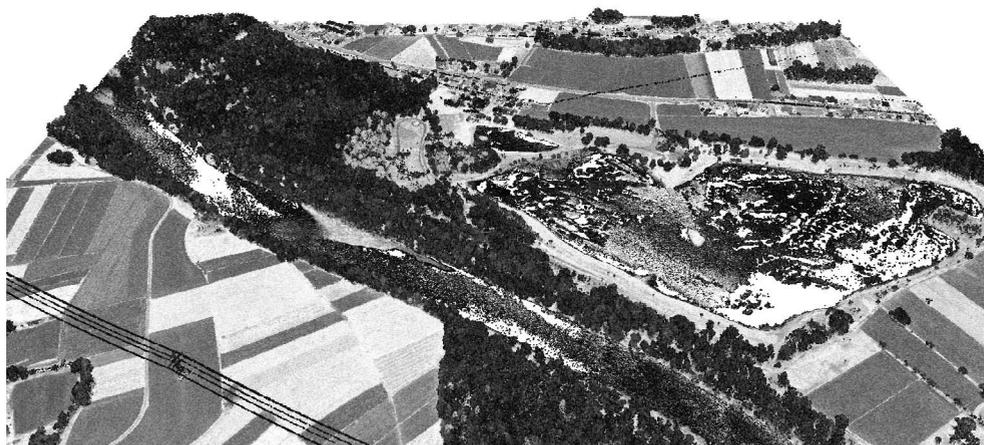


Fig. 5: Watered areas cause discontinuities in the point sets, reducing the efficiency of the prediction

In Table 2, the results achieved by compression algorithms over the testing dataset are presented.

Table 2: Testing datasets for comparison of compression algorithms

File	LAS Compression		LASZip		LiDAR Compressor		7-Zip	
	Compression Ratio (%)	Bits per point						
1	10,8	24	11,1	25	27,9	63	21,0	47
2	9,6	22	10,0	22	26,7	60	20,3	45
3	9,3	21	9,51	21	26,0	58	20,0	45
4	15,9	36	17,3	39	32,5	73	26,1	59
5	14,2	32	16,0	36	31,5	71	25,0	56
6	15,7	35	17,2	39	16,3	37	25,0	56
7	16,5	37	18,0	40	32,7	73	24,1	54
8	16,0	36	15,7	35	32,9	74	26,5	59
9	16,2	36	17,3	39	32,1	72	23,5	53
10	16,2	36	14,6	33	35,8	80	38,2	86
11	18,0	38	22,8	47	21,1	44	26,1	54
12	11,8	26	12,0	27	30,0	67	22,0	49
Avg.	14,2	31,6	15,1	33,6	28,8	64,3	24,8	55,3

As seen in Table 2, average compression ratios achieved by domain-specific algorithms highly differ. Although *LiDAR Compressor* is domain-specific algorithm, it is still outperformed by a general-purpose 7-Zip algorithm. Even more, comparable domain-specific *LASCompression* algorithm is capable to further increase a compression ratio by a factor two, while it is being closely followed by *LASZip*. When comparing bits-per-point (BPP), a reduction of nearly 200 BPP can be expected in case of *LASCompression*. Figure 6 shows the variations of bits per point achieved by different algorithms.

Although performance of 7-Zip only slightly depends on the terrain type, the effect on domain-specific algorithms is much more noticeable. *LAS Compression* excels at flat terrain types, where prediction schema is much more accurate. The compression ratio achieved is on average around 10%. *LASZip* is slightly less effective, except in cases where the points are stored in lesser density – density presenting the average number of points per square meter. *LAS Compression* is not significantly affected by different point densities. Lizardtech's *LiDAR Compressor* and *7-Zip* produce poorer results.

Compression ratios decrease when compressing other terrain types as mentioned earlier. *LASCompression* still prevails, compressing more problematic terrain types on average to around 16% and reducing the bits per point to around 35. Again *LASZip* follows closely with *LiDAR Compressor* and *7-Zip* not reaching efficient compression ratios.

4 Conclusion

In this paper we compared different algorithms for compressing LiDAR data. There are efficient algorithms available (*LASCompression*, *LASZip*) that can greatly reduce the spatial requirements of LiDAR data. The results of *LASCompression* and *LASZip* indicate that the

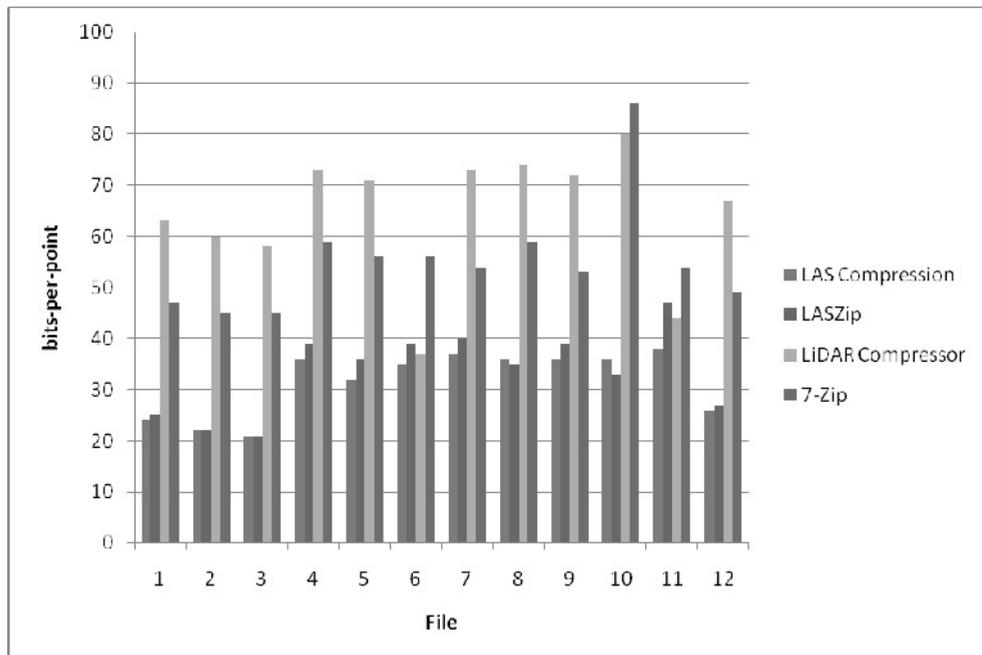


Fig. 6: Comparison of bits per point needed for storing one point in compressed files

efficiency of their compression is in tight correlation with different terrain types and also somewhat with density of points.

Flat terrains provide a better opportunity for effective compression, while urban areas and riversides present distributions of points that are managed with greater difficulty. Furthermore, *LASZip* performs better with a lower point density, while *LASCompression* performs similarly effective regardless of density. By its disposition the general-purpose compression algorithm *7-zip* does not rely on any geometric factors for compression and is thus basically independent of terrain types or point density.

LiDAR Compressor, though claimed to be intended for LAS file compression, does not seem to take advantage of geometric coherence of LiDAR data. The results of its compression are not comparable to other algorithms, as it is outperformed even by the general-purpose compression algorithm.

Acknowledgement

This work was in part supported by the Slovenian research agency under the grant L2-3650.

The authors would like to thank the company GeoIn, d.o.o. (www.geoin.si), for providing large amount of LAS files and for performing extensive test of the *LAS Compression* application.

References

- 7-ZIP (2011), 7-Zip Home. <http://www.7-zip.org/> (accessed at 10.01.2011).
- ASPRS (2009), LAS specifications.
<http://www.asprs.org/society/committees/standards/> (accessed at 10.01.2011).
- HUFFMAN, D. A. (1952), A Method for the Construction of Minimum-Redundancy Codes. *Proc. I.R.E.*, 40 (8 September 1952): 1098-1101.
- ISENBURG, M., LINDSTROM, P. & SNOEYINK, J. (2005), Lossless compression of predicted floating-point geometry. *Computer-Aided Design*, 37 (8): 869-877.
- KLAJNŠEK, G. & ŽALIK, B. (2005), Progressive lossless compression of volumetric data using small memory load. *Computerized Medical Imaging and Graphics*, 29 (4): 305-312.
- LASCOMPRESSION (2011), LASCompression Home.
<http://gemma.uni-mb.si/lascompression/> (accessed at 10.01.2011).
- LASZIP (2011), ASlib, LASzip, and LAStools: converting, filtering, viewing, processing, and compressing LIDAR data in LAS format.
<http://www.cs.unc.edu/~isenburg/lastools/> (accessed at 10.01.2011).
- LiDAR COMPRESSOR (2011), Lizardtech's LiDAR Compressor at.
<http://www.lizardtech.com/products/lidar/> (accessed at 10.01.2011).
- LIU, X. (2008), Airborne LiDAR for DEM generation: some critical issues. *Progress in Physical Geography*, 32 (1): 31-49.
- LUOMA, O. & TEUHOLA, J. (2007), Predictive modeling in XML compression. *Proc. 2nd International Conference on Digital Information Management*. Lyon, France, 28-31 October 2007, pp. 565 - 570.
- MANUE, D. F. (2008), Aerial mapping and surveying. In: DEWBERRY, S. O. & RAUENZAHN, L. N. (Eds.), *Land development hand-book*. 3th edition. McGraw-Hill Professional, pp. 877-910.
- MONGUS, D. & ŽALIK, B. (2011), Efficient method for lossless LIDAR data compression. *International journal of remote sensing* (in press).
- RISSANEN, J. J. & LANGDON, G. G. (1979), Arithmetic coding. *IBM Journal of Research and Development*, 23 (2): 149-162.
- SALOMON, D. (2006), *Data compression: The complete reference*, 4th edition. Springer, 1092 p.
- SHANNON, C. E. (1948), *A Mathematical Theory of Communication*. *Bell System Technical Journal*, 27: 379-423.
- SHANNON, C. E. (1951), *Prediction and Entropy of Printed English*. *Bell System Technical Journal*, 30: 50-64.
- WELCH, T. A. (1984), A Technique for High-Performance Data Compression. *Computer*, 17 (6): 8-19.