

Terrestrial Laser Scanning Data Compression Using JPEG-2000

Grzegorz Józków¹ 

Received: 20 March 2017 / Accepted: 29 August 2017 / Published online: 26 September 2017
© The Author(s) 2017. This article is an open access publication

Abstract The size of remotely sensed data is constantly growing, which causes problems related to data transfer and archiving. The most common solution of this problem is data compression. This work investigates the feasibility of sensor-level TLS data compression using JPEG-2000 lossless and lossy schemes. TLS data are treated as a multi-band image or a set of images consisting of range, intensity, and optionally RGB bands. The investigation focuses on performance assessment in terms of storage space saving, data distortion due to lossy compression, and compression and decompression speed. Experiments executed on real TLS data show that the proposed JPEG-2000 strategy for both lossy and lossless schemes allows to save more storage space than other compression methods. The lossy scheme resulted in the strongest compression and only had insignificant impact on data distortion. Finally, JPEG-2000 compression was identified as the fastest method. It is even suitable for real-time compression at sensor-level.

Keywords Terrestrial laser scanning · Compression · JPEG-2000 · Performance analysis

Zusammenfassung *Kompression von terrestrischen Laserscanningdaten auf Basis von JPEG-2000* Das Volumen von Fernerkundungsdatensätzen wächst kontinuierlich, was unterschiedliche Probleme verursacht, z.B. bei der Datenübertragung oder der Archivierung. Die am meisten benutzte Abhilfe dieser Probleme besteht in der Anwendung von Methoden zur Datenkompression. Diese Arbeit untersucht

die Kompression von TLS-Daten auf der Sensor-Ebene mit Hilfe des JPEG-2000 Standards, wobei sowohl verlustfreie als auch verlustbehaftete Verfahren angewendet werden. Die TLS-Daten werden als ein Bild oder als ein Satz von Bildern behandelt, die die Entfernung, Intensität oder optional auch RGB-Bänder enthalten. Diese Untersuchung konzentriert sich auf die Bewertung in Bezug auf Speicherplatzersparnis und Fehler, die durch die verlustbehaftete Kompression erreicht werden. Die Kompressions- und die Dekompressionsgeschwindigkeit werden ebenfalls evaluiert. Die auf realen TLS-Daten durchgeführten Experimente zeigten, dass die vorgeschlagene JPEG-2000-Strategie für die verlustbehaftete und die verlustfreie Kompression ermöglicht, mehr Speicherplatz als andere Kompressionsmethoden zu sparen. Die verlustbehaftete Kompression mit den stärksten Kompressionsraten hatte nur eine unwesentliche Auswirkung auf die Qualität der Daten. Zusammenfassend wurde die JPEG-2000-Kompression als die schnellste Methode identifiziert. Sie ist sogar für eine Echtzeitkompression auf Sensor-Ebene geeignet.

1 Introduction

Current sensor development in LiDAR (Light Detection and Ranging) technology is mostly focused on two aspects: creating multi-wavelength laser scanners and introducing new types of sensors, namely low-power single-photon and Geiger mode array sensors (Toth and Józków 2016). Both paths of development will result in a significantly larger amount of data and acquired in shorter time than in the case of pulsed single wavelength high-power laser scanners used in current practice. The problems arising with respect to the increasing data volume can be solved by real time or near real time data compression at the level of the sensor, prior to the data recording on the scanner disk.

✉ Grzegorz Józków
grzegorz.jozkow@igig.up.wroc.pl

¹ Institute of Geodesy and Geoinformatics, Wrocław University of Environmental and Life Sciences, Grunwaldzka 53, 50-357 Wrocław, Poland

The problem of compressing laser scanning data is discussed in the literature with respect to several aspects. In the case of airborne LiDAR data, first efforts were put in the compression of point clouds (Mongus and Žalik 2011; Iseburg 2013) created by combining LiDAR and navigation sensor data. These methods were designed for efficient compression of airborne LiDAR point clouds; however, they are frequently used to compress TLS (Terrestrial Laser Scanning) point clouds or even point clouds created from dense image matching. Nevertheless, the background for these methods was similar to image compression—the increase of the compression rate is obtained by exploiting the correlation between neighbouring data samples. However, in an irregularly distributed 3D point cloud it is more difficult to define an appropriate neighbourhood. This problem can be solved by switching from the point cloud to sensor raw data, because laser scanners collect data in a very regular manner. This strategy was applied to the compression of airborne LiDAR full waveform data (Józków et al. 2015a) and showed that compression efficiency increases if the data decorrelation is performed in more dimensions (Józków et al. 2015b). Moreover, data regularity allows to use typical image compression tools such as the JPEG-2000 standard (Taubman and Marcellin 2002) that eliminate problems with standard development and software implementations.

This work discusses the problem of applying JPEG-2000 compression to TLS data at their place of origin—the sensor. Since terrestrial laser scanners very often offer colour information from built-in RGB cameras, the joint compression of TLS data and RGB intensities was also investigated. This paper is structured as follows: Sect. 2 discusses the issue of TLS data regularization to make them suitable for image compression tools; Sect. 3 describes the executed experiments in terms of applying the JPEG-2000 standard to TLS data, assessment of compression performance, and test data; Sect. 4 shows and discusses the obtained results; and Sect. 5 concludes the investigation.

2 TLS Data as an Image

2.1 Image Formation

The principle of measurements for most terrestrial laser scanners is identical to the principles that are used in total station measurements. Both instruments use spherical coordinates: horizontal and vertical angles, and range (α , θ , and R in Fig. 1) that are subsequently transformed to more practical Cartesian coordinates XYZ. The mechanical construction of such scanners determines the manner in which measurements are executed. Typical terrestrial laser scanners have two rotation axes (Fig. 1). The first one is the vertical axis of the scanner and the second one is horizontal axis of scanner head (mir-

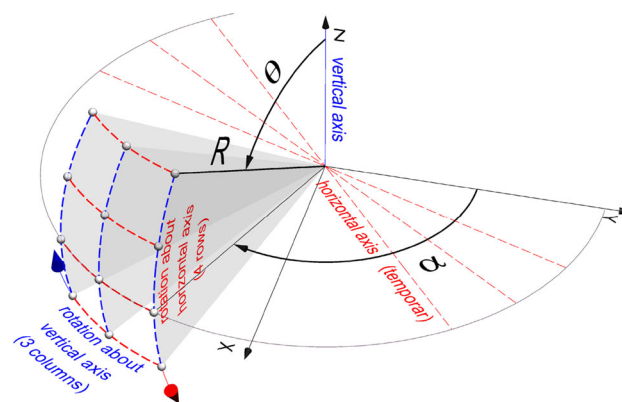


Fig. 1 Principle of TLS data acquisition. Rotations of scanning mechanism about horizontal and vertical axes result in arranging laser shots in rows and columns, respectively

ror). The second axis is not fixed during data acquisition but changes its direction together with the rotation about the vertical axis. The combination of both movements causes that scanning points are distributed in a regular pattern: the rotation about the vertical axis creates scanning columns, while the rotation about the second axis arranges points in scanning rows. Note that here, data regularity means the arrangement of scanning points in rows and columns even if their spatial distribution is irregular.

The arrangement of points in rows and columns allows to see TLS data as an image where measured ranges are pixel values (range image). Vertical and horizontal angles do not need to be stored for every point, because they can be restored knowing the scanner kinematics (e.g. rotation speed, pulse or sampling rate, etc.), calibration parameters (see, e.g. Glennie 2012) and angular offsets for the pixel that is placed in the first row and the first column of created image. These parameters are low-size metadata and, thus, do not require compression. Cartesian coordinates XYZ of points can be determined knowing row and column indices in the image, pixel values (ranges), and metadata. Storing the metadata for an entire scan and ranges for all points is more storage space efficient than keeping three Cartesian coordinates for all points. Obviously, laser shots without return (zero value pixels in the range image) should be kept to preserve image structure, but images with many neighbouring pixels having the same value can usually be well compressed.

2.2 Image Bands and Metadata

The number of created image bands depends on the number of measured attributes that are unique for scanning points. Beside range and intensity, some scanners measure other parameters, such as Doppler velocity in frequency-modulated continuous wave laser scanners (Massaro et al. 2014). Emitted laser shots without any return can get zero

range and intensity values meaning that points are not created for such image pixels. On the other hand, in pulsed laser scanners, single laser shots may result in multiple returns. If these returns are created directly by the scanner, additional bands for range, intensity, and other parameters need to be created similarly to multi-band images (image cubes) acquired by multispectral cameras. The lack of a certain return will cause zero values of the appropriate pixels of image cubes. LiDAR image cubes created by multi-return systems can be compressed using methods that are appropriate for multi-band image compression. If the sensor stores waveform data, the compression methods for airborne LiDAR waveform cube proposed by Józko^o et al. (2015a, b) can be adopted. In this case, two dimensions of the cube will be created according to the method described in Sect. 2.1 and the third dimension will depend on the maximal number of waveform samples. In the case of terrestrial laser scanners that use a few laser diodes emitting different wavelengths, image cubes can be created separately for the data collected by each wavelength. This will result in more image cubes. However, diodes may emit pulses with different frequencies, resulting in a different number of rows and columns in each cube and, consequently, preventing the creation of a single cube containing more bands.

Since the vast majority of terrestrial laser scanners has a built-in camera, or can be equipped with an external RGB camera, to collect images needed for point cloud colouring, the image data can also be compressed as the auxiliary data. There are two possible approaches for compressing such data using image compression methods. The first option is the independent compression of all collected images. The second approach requires point cloud colouring executed during data collection. In this approach, three (RGB) additional bands are created according to the same procedure as described in Sect. 2.1. The latter method can be used in the case of built-in cameras, because known and stable relative orientation between camera and laser sensor allows to perform colour projection without the need of additional image processing.

The metadata are not unique for each point. Beside the data mentioned above, required to assign spherical coordinates to image pixels, i.e. the parameters of scanner kinematics, and angular offsets for the pixel that is placed in the first row and the first column of the created image, the metadata needed for the proposed compression scheme contain two other groups of parameters. The first group contains the information about image bands that is not included in the compressed images, e.g. parameters that allow to convert ranges from pixel intensities to metrical values. The second group contains typical TLS metadata, such as the number of scans or their transformation parameters to an external coordinate system. All metadata can be arranged in an ASCII file, e.g. in xml format, that is significantly smaller than the compressed TLS data.

2.3 Bit Depth

Beside the image dimension, also its bit depth has a significant impact on the storage space required by the uncompressed image. The above-mentioned RGB images used for point cloud colouring typically have a depth equal to 24 bits (8 bits per band), though sensors usually use lower depth. The LAS standard (ASPRS 2008), assumes 16 bits to store each of three or four colour attributes for every scanning point. Similarly, intensity values are usually detected by the sensor with a lower bit depth than the 16 bits used in the LAS format. In the case of range images, the bit depth should be sufficient to store the range with higher precision than the accuracy of range measurements. For example, the distance measurement accuracy for Leica ScanStation C10 scanner is 4 mm (at 1–50 m range). This means that all ranges up to about 500 m (Leica ScanStation C10 range is 300 m) can be stored with 4 mm precision using only 17 bits. Storing these ranges with better precision requires more bits. The minimal bit depth necessary to store ranges can be selected depending on the scanner parameters (ranging accuracy) or even by the scanning purpose in which only a certain range of measured ranges need to be stored:

$$b = \left\lceil \log_2 \left(\frac{R_{\max} - R_{\min}}{\varepsilon} \right) \right\rceil \quad (1)$$

where b is the approximated minimal bit depth, R_{\max} is the maximum observed range, R_{\min} is the minimum observed range, and ε is the range precision.

Obviously, ranges can be stored in the image as 32 bit floating point numbers (similarly as terrain heights in a digital terrain model stored as geotiff) without the need of encoding them into integers, especially if such image is subsequently compressed. However, more bits than the required minimum may limit compression performance in the case of the lossless JPEG-2000 scheme. Another rationale to use integers is that A/D converters in laser scanners also use integers; thus, integers are better-suited for sensor-level compression. From the implementation point of view, the bit depth used to encode ranges into integer numbers should be a multiple of 8 (single byte), thus a bit depth equal to 24 seems to be optimal to encode TLS ranges. 24 bits are sufficient to store ranges up to about 16.777 km with a precision of 1 mm.

3 Methodology

3.1 JPEG-2000

Images created from TLS data can be compressed using a variety of image compression methods. A previous investigation on waveform data compression (Józko^o et al. 2015a)

showed that JPEG-2000 compression has better performance than other tested methods; thus, it was used also in this study. The JPEG-2000 standard provides a wide range of compression features that broaden its applications. However, the implementation of a complex standard with all its functions is challenging. In addition, some of the technologies included in the JPEG-2000 standard use intellectual property rights (IPR). However, the JPEG-2000 committee worked diligently to ensure that Part 1 of the standard could be implemented on a royalty- and fee-free basis (Rabbani et al. 2009). Technologies that were excluded from Part 1 because of their complexity or IPR issues have been included in Part 2 of the JPEG-2000 standard (Rabbani et al. 2009).

Among the most interesting features of the JPEG-2000 standard, three should be mentioned in the context to this investigation. The first one is the ability to perform lossy or lossless compression schemes depending on the requirements (Skodras et al. 2001). The second one is the support for high dynamic range images (Xu et al. 2005) or even floating point data (Gamito and Dias 2004). The third one is the multicomponent transform that allows for efficient compression of volumetric data, i.e. image cubes (Lepley et al. 2009).

In general, the lossy and lossless JPEG-2000 compression schemes differ at two stages. The first difference occurs during data decorrelation where the wavelet transform using CDF 9/7 or CDF 5/3 wavelet (Cohen et al. 1992) is used for the lossy or lossless scheme, respectively. The main difference occurs in the next stage—quantization encoding. In the case of the lossy scheme, quantization steps larger than 1 result in a partial reduction of the size of the data and cause information loss. A quantization step equal to 1 does not result in data distortion and reduction; thus, it is used during lossless compression.

JPEG-2000 can be used to compress images of any bit depth, including images with pixel values given as floating point numbers. However, the use of non-integer numbers causes small data distortion due to round-off errors during wavelet transform even if the lossless compression is applied. For that reason, in this investigation ranges were encoded into integer numbers before JPEG-2000 compression.

The last of the mentioned JPEG-2000 features (multicomponent transform) is used for data decorrelation between image bands that can benefit in a more efficient compression of volumetric data, e.g. hyperspectral images. Note that RGB images can also be treated as volumetric data containing three bands, but in the typical JPEG-2000 compression of RGB images, the transform between bands is performed only to change colour space. Each of the transformed colour components (bands) is handled separately during the compression.

All three features mentioned above are incorporated in the commercial Kakadu Software (www.kakadusoftware.com/). This software in the form of compiled by developer binaries

(Copyright is owned by NewSouth Innovations Pty Limited, commercial arm of the UNSW Australia in Sydney) was used in this investigation. Although multicomponent transform is included in Part 2 of the JPEG-2000 standard, the used implementation does not involve 3rd party IPR technologies.

A practical implementation of JPEG-2000 compression at the level of the sensor assumes that images are created in the scanner memory in real time (during scanning), compressed after the end of the scan, then saved as compressed files on the scanner disk or the memory card. For example, after receiving an echo, the range and intensity are determined and stored as pixels in separate images. The lack of a return causes adding zero values to appropriate images. Pixels are added to the image column by column, because the column direction matches the scanning order (Sect. 2.1). Obviously, the size of the scanner memory may be insufficient to keep all pixels from the whole scan, but the compression and writing of files on the disk can be executed progressively after collecting data for a certain number of columns. Compressed files containing data of parts of the scan can be subsequently merged in the JPEG-2000 domain (Rosenbaum and Taubman 2003) without the need of decompression. This implementation can be scaled to multi-return and multi-wavelength systems resulting in a larger number of images created in the memory. Images having identical number of rows and columns can be arranged in a cube and compressed as the volumetric data.

3.2 Algorithm

The algorithm of TLS data compression with the JPEG-2000 standard presented in this work consists of two major steps. The first one is TLS data preprocessing to create the images that are subsequently compressed in the second step.

Since the format of raw files produced by scanners used in this investigation is unknown, the collected data had to be converted first. Scanner manufacturers provide the software that allows to convert these files into point clouds in several formats. Among the available formats, a simple ASCII PTS format was used as the most convenient one to convert point clouds into images. This format keeps each point as a separate row containing an XYZ triplet, intensity, and RGB triplet. Cartesian XYZ coordinates were transformed to spherical coordinates necessary for this investigation. The obtained horizontal and vertical angles were used to determine the total number of rows and columns in the created images as well as the row and column indices for each scanning point. Although test data were collected by pulsed scanners, polar coordinates showed that the test data contain only single returns.

Five image bands were created for each data set. The first created band contained ranges where metric values (deci-

Table 1 Parameters of image cubes with respect to experiment variant

Variant (number of cubes)	Bands composing cubes	Band depth (bits)
1	(5) Range, intensity, R, G, B	24
2	(1) Range (4) Intensity, R, G, B	24 12
3	(1) Range (1) Intensity (3) R, G, B	24 12 8

mal numbers) were encoded into unsigned integers using linear 24 bit quantization. The next band contained intensities. Because in PTS files intensities were given as signed 12 bit integers, they were changed to unsigned 12 bit integers by adding a constant value of 2048. Finally, three auxiliary bands containing RGB components were created. Values of RGB intensities were provided for each band as 8 bit unsigned integers and, thus, did not require any conversion.

The second step of the compression experiments was executed in three variants, depending on the number of compressed image cubes and used bands. The rationale of applying compression on more than one image cube is the different physical meaning of the created bands. Particularly, range is different from intensity bands; note that RGB values are also intensities but obtained by passive sensors and for different wavelengths. Also the depth of the created range, intensity, R, G, and B bands is different and equal to 24, 12, 8, 8, and 8 bits, respectively. In the first variant of the experiment, a single cube containing all five bands was created. In the second variant, two cubes containing 1 (range), and 4 bands (intensity, R, G, and B), respectively, were created. In the third variant, three cubes were created containing 1 (range), 1 (intensity), and 3 (R, G, and B) bands, respectively. In the variants where image cubes were composed of bands having different depths, bands of lower depths were scaled to match the maximal bit depth of all bands in this cube. This scaling caused no loss of information, though files subject to compression temporarily occupied more space on the disk. Image cubes were saved on the disk as binary files encoded according to band sequential (BSQ) scheme where each pixel was stored as an unsigned integer. Synthetic information about tested variants, including the number of compressed cubes, bands, and their bit depth is given in Table 1. In addition, compression in each variant was executed using lossy and lossless JPEG-2000 schemes. The results of compression in all variants were stored as JPX files.

3.3 Compression Performance

The performance of compression can be analysed with respect to several aspects:

- Level of data (file size) reduction,
- Degree of data distortion if lossy compression is used,
- Compression and decompression speed.

Seen from the perspective of the main goal of compression, the reduction of file size, the primary measure of compression performance should describe the gain of storage space. Usually, relative parameters describing the ratio of the uncompressed to the compressed file size are used. However, such parameters may be misleading if there are various file formats used to store the uncompressed data. Consequently, it may cause the size of uncompressed files containing the same data to differ. For example, the size of E57 (Huber 2011) and PTS files containing identical TLS point cloud differ significantly. Even the most popular LAS format is not optimal in terms of storage space minimization. In many cases, LiDAR point clouds do not contain information about Point Source ID, or Scan Angle Rank (see ASPRS 2008), but LAS format reserves storage space for such parameters for every point. For that reason, a parameter describing the level of data reduction that is independent of the file format is preferred. In the case of image compression, such a parameter is the bits per pixel (BPP) ratio or bits per pixel per band (BPPPB) ratio used for single- and multi-band images, respectively (Serra-Sagristà and Aulí-Llinàs 2008). The parameter BPP ratio describes the average number of bits used to store a single pixel of the image. A lower value of this parameter means a better compression performance. Analogically to image data where pixels are the elementary elements, in this work the bits per point (BPP) ratio was introduced to assess compression performance, because a scanning point can be treated as the elementary element of TLS data. The BPP ratio can also be used to compare storage space efficiency of different file formats including compressed and uncompressed formats. In this investigation, the BPP ratio was calculated for JPEG-2000 compressed TLS data. The compression performance was assessed by comparing the obtained values with BPP ratios calculated for the same data, but stored in the following file formats:

- RAW data files stored by the scanner; note that this format differs between manufacturers and/or scanners and

is usually unknown to the public; however, it is likely to be an uncompressed binary format,

- PTX ASCII format developed by Leica (2015); this format was used only to store TLS data containing RGB components,
- LAS v1.2 binary format (ASPRS 2008); note that the smallest Point Data Record Formats 0 and 2 (see ASPRS 2008) were used to store TLS data without and with RGB values, respectively,
- PCD binary format used by the Point Cloud Library (PCL 2017),
- E57 binary format developed for TLS data exchange (Huber 2011),
- BIN binary format used in the CloudCompare software (Girardeau-Montaut 2011),
- ZIP losslessly compressed files created from PCD files using the 7-zip Software (Pavlov 2016) with the highest possible compression level selected; note that ZIP is a general purpose compression method that can be used for any data stored as a file,
- LAZ losslessly compressed files using LASzip method (Isenburg 2013); LASzip compression was applied to created LAS files; note that this compression method was developed to compress LAS files containing airborne LiDAR data.

The size of files containing point clouds stored in some formats is sensitive to the precision of point attributes. This applies especially to point clouds stored in ASCII formats or losslessly compressed files. Because ranging errors for typical TLS scanners are usually not smaller than 1 mm, there is no practical need to store Cartesian coordinates with a precision higher than 0.1 mm. For that reason, Cartesian coordinates in all file formats listed above as well as range data used to create binary BSQ files compressed using the lossless JPEG-2000 scheme were stored with a precision equal to 0.1 mm. Similarly to coordinates, the precision of intensity values influences the size of files if intensities are expressed as decimal numbers. Some formats use exclusively decimal notation for intensities (e.g. PTX uses 0–1 range), others use exclusively integer notation (e.g. LAS), and other ones use decimal or integer values (e.g. PCD, E57, BIN) depending on data source. As the data used in this study have 12 bit intensity depth, a sufficient precision of intensity values given as decimal numbers in the range from 0 to 1 is equal to about

0.0001. In this study, decimal notation of intensities was used in PTX and PCD formats.

Since lossy compression methods, especially JPEG-2000, offer very high compression ratios, the parameter that describes the amount of compression should be given together with the parameter that shows data loss, or more precisely, data distortion after decompression (reconstruction). There are two common parameters independent of the data type that are used to describe data distortion due to lossy compression: signal to noise ratio (SNR) and peak signal to noise ratio (PSNR). Seen from the perspective of TLS data compression, these parameters may be less relevant since it is difficult to transfer certain SNR or PSNR values to a certain distortion of the range and intensity. If lossy compression is applied to a physical signal, the distortion that is below the signal noise is negligible (Józków et al. 2015b). Similarly, the distortion of other observation (e.g. range) can be neglected if it is lower than the measurement error. Because ranging, intensity, and RGB errors differ between terrestrial laser scanners (some of them are not even provided by the manufacturers), typical values according to the knowledge of the author were used in this work (Table 2). The distortion of TLS data due to lossy compression was treated as irrelevant if none of the thresholds was exceeded. Two thresholds were defined for each type of observation. The first threshold is related to the maximal absolute residual (MAR) between the given and the value reconstructed after compression for all pixels of TLS image, which was set to the value of a typical error. The second threshold is related to the mean absolute error (MAE) calculated from all residuals and it was set to a value equal to 10% of typical error. Exact values of the used thresholds are shown in Table 2. It should be emphasized that the assumption about errors being negligible are very rigorous. Some applications may accept much higher distortion or might not use intensity or RGB components.

The lossy compression method allows the user to decide about the compression strength. This is achieved by specifying compression levels (e.g. low, medium, high, etc.), compression factors (e.g. values between 1 and 100), or more useful parameter such as the expected BPPPB ratio used in this work. However, data distortion with respect to the used compression strength varies between data sets, and is difficult to predict. For that reason, lossy JPEG-2000 compression was executed with several compression strengths

Table 2 Typical TLS data parameters and thresholds of acceptable distortion

Data	Typical TLS data		Threshold for acceptable distortion	
	Range of values	Measurement error	MAR	MAE
Range	1–300m	2 mm	2 mm	0.2 mm
Intensity	12 bits	3 DN	3 DN	0.3 DN
R, G, B	8 bits (each band)	2 DN	2 DN	0.2 DN

to estimate the value of expected BPPPB ratio that fulfils the threshold requirements given above. Note that the expected BPPPB ratio means the average value for an entire image, thus it can be specified by the user as a decimal number.

Another aspect of compression performance is compression and decompression speed. In the application of real time or near real time sensor-level data compression, high compression speed is desired. One of the goals of this work is to achieve compression speed that allows to compress TLS data progressively during scanning. Note that decompression speed has a lower priority since sensor-level compression of TLS data is executed mainly for transferring and archiving purposes. The evaluation of JPEG-2000 performance in terms of TLS data compression and decompression speeds was executed by comparing the speed of JPEG-2000 compression to the speed of two methods for TLS (point cloud) data compression; ZIP and LASzip that were used also to evaluate compression strength. The speed parameter introduced here for this evaluation was the average time necessary to compress or decompress one scanning point.

The design of this experiment requires identical conditions for each test in terms of the assigned processing power and any slowing-down effects. As for all methods the compression was executed using binaries compiled by developers, it was impossible to exclude the part of the time related to disk operations necessary for loading input files into memory and saving compressed or decompressed files on the disk. To minimize this time, a PC with fast solid-state drive (SSD) was used in the experiments, and all files were read from and written to this drive. In addition, all tests were repeated

5 times to eliminate any slowing down effects related to the operating system. The time was measured automatically as the time in which executed binaries were operating.

3.4 Test Data

Numerical experiments were executed on six data sets collected by three terrestrial Leica ScanStation laser scanners: 2, C10, and P20 (Table 3). Test data were collected for various scenarios with different scanning resolutions, fields of view, and measured ranges. Intensity images (one of the compressed bands) created from test data are shown in Fig. 2 while parameters of scans are shown in Table 4. The variety of the data allows to evaluate the impact of their variability on JPEG-2000 compression performance.

4 Results and Discussion

4.1 Data Distortion Due to Lossy Compression

The first experiments aimed at determining the value of the user input parameter (expected BPPPB ratio) that gives the strongest compression, but keeps the data distortion under the specified thresholds (Table 1). Figures 3, 4, and 5 show the obtained distortion (MAR and MAE) for range, intensity, and RGB values with respect to the expected BPPPB ratio, for three variants of data arrangement in 1, 2, and 3 cubes, respectively. Horizontal black lines in Figs. 3, 4, and 5 show the selected threshold values.

Table 3 Parameters of scanners used to collect test data

Scanner	Leica ScanStation 2 (Leica 2007)	Leica ScanStation C10 (Leica 2011)	Leica ScanStation P20 (Leica 2013)
Type	Pulsed	Pulsed	Pulsed
Wavelength	532 nm (green)	532 nm (green)	658 (red) 808 (infrared)
Scanning mechanism	Rotating mirror protected by housing and two glass windows	Vertically rotating mirror on horizontally rotating base	Vertically rotating mirror on horizontally rotating base
Maximal FOV	360° (horizontal) 270° (vertical)	360° (horizontal) 270° (vertical)	360° (horizontal) 270° (vertical)
Maximal scan rate	50,000 points/s	50,000 points/s	1,000,000 points/s
Range	300 m @ 90% albedo 134 m @ 18% albedo	300 m @ 90% albedo 134 m @ 18% albedo	120 m @ 18% albedo
Distance accuracy	4 mm	4 mm	≤1 mm
3D position accuracy	6 mm @ 1–50 m	6 mm @ 1–50 m	3 mm @ 50 m 6 mm @ 100 m
Built-in camera	24° × 24° FOV 1024 × 1024 pixels	17° × 17° FOV 1920 × 1920 pixels	17° × 17° FOV 1920 × 1920 pixels

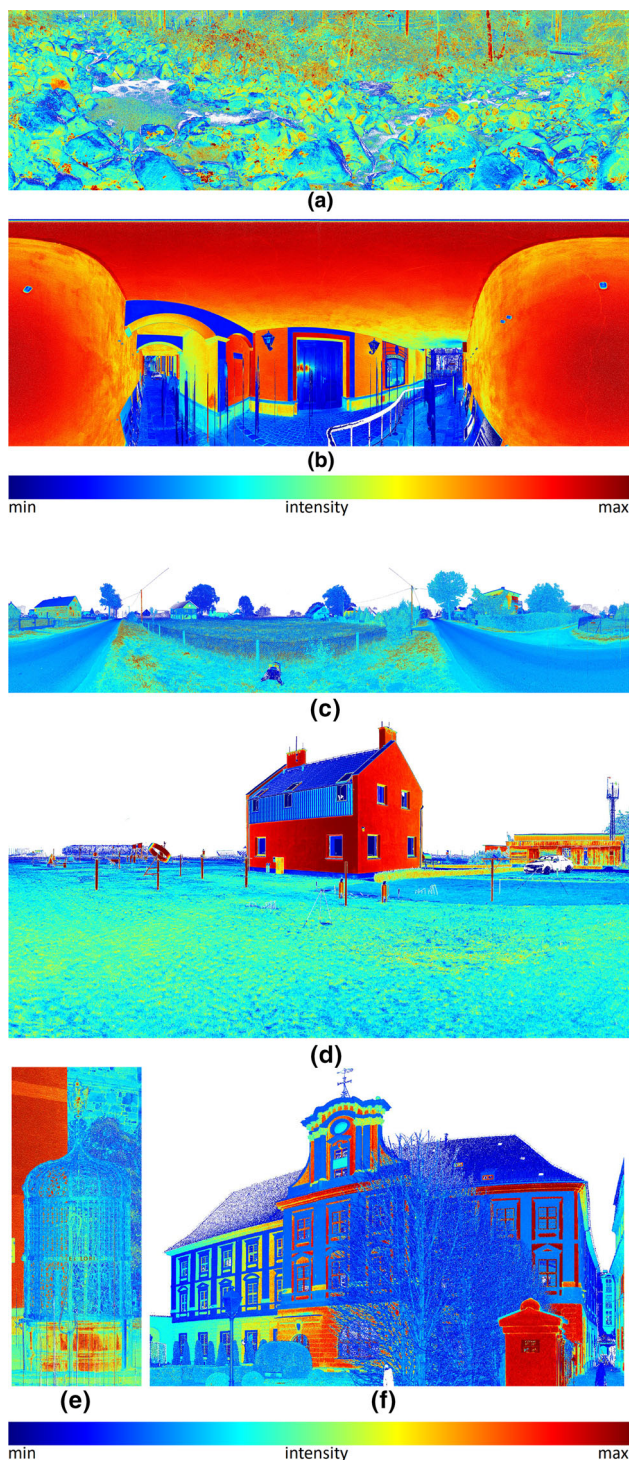


Fig. 2 Intensity images created from test data: **a** forest with mountain river bed, **b** passage under the building connecting two streets, **c** village, **d** building and lawn, **e** top of the well, **f** facade. Colour map is stretched between 1st and 99th intensity percentiles for each data set separately. White colour means lack of return for a certain laser pulse

An analysis of Figs. 3, 4, and 5 shows that range data can be compressed in a lossy way without significant distortion if compressed separately for intensity and RGB components

(variants 2 or 3 of data arrangement). In this case, thresholds for both MAR and MAE were not exceeded if a compression ratio of 4–8 BPPPB was used, depending on the test data (Figs. 4a or 5a). In the case of compressing all components in a single cube (variant 1 of data arrangement), the acceptable range distortion could be obtained at a much lower data reduction, because ratios of 8–13 BPPPB had to be used to keep the distortion under the assumed thresholds (Fig. 3a).

In the case of intensity data, the lowest compression ratio with respect to data distortion was obtained if all data components were compressed simultaneously in a single cube. Thresholds were not exceeded if ratios of 5–8 BPPPB were used (Fig. 3b). Slightly higher ratios were necessary in the case of compressing intensity and RGB components in one cube (Fig. 4b), and the highest BPPPB ratios if all components were compressed separately (Fig. 5b). However, the number of components (variant of data arrangement) has a low impact on intensity distortion, because in variant 3, a distortion comparable to the one in variant 1 could be obtained just by increasing the expected BPPPB ratio by about 1.5 bits. Similar statements can also be formulated for the RGB components. The lowest distortion was obtained if all components were compressed together (Fig. 3c), but acceptable distortion was obtained in two other variants by increasing BPPPB ratio just by 1 bit.

As in lossy compression of multi-band images the same compression strength (expected BPPPB ratio) must be selected by the user for all bands, it is impossible to compress TLS data in variant 1 using 13, 9, and 4 BPPPB ratios for range, intensity, and RGB components, respectively. To keep the acceptable distortion for all components, the highest 13 BPPPB ratio should be selected. This means that after compression according to variant 1, 65 bits are required to store all components of a single pixel. Significantly less bits are required in variants 2 and 3. The variant 2 requires about 40 bits (8 bits for single-band range cube and 8 bits for each component of four-bands intensity-RGB cube). The number of bits necessary to store a single pixel is lowest in variant 3. It requires only about 32 bits (8, 9, and 5 bits to store range, intensity, and RGB components, respectively). This variant was used in subsequent experiments. Note that a given value of 32 bits is the maximal for all test data sets, because for data sets where many laser shots do not have returns, e.g. data set (d) Building, lossy compressed range and intensity components keep an acceptable distortion at lower BPPPB ratios. This can be beneficial if multiple returns from single laser shots are observed. Additional cubes containing consecutive returns will have more pixels having zero value, but can be compressed with lower BPPPB ratio, so that the BPP ratio will stay low.

The analysis of the spatial distribution of range distortion (Fig. 6) obtained in variant 3 shows that JPEG-2000

Table 4 Parameters of test data

Test data	Leica ScanStation scanner	Approximated number of		Points to pixels ratio	Approx. FOV (°)		Point angular separation (")	Ranges (m)			
		Points	Created pixels		H	V		Min	Max	Median	Std.
(a) Forest	C10	79.7M	83.3M	0.96	207	53	41	0.2	146.8	6.5	8.5
(b) Passage	C10	15.1M	15.4M	0.98	360	132	200	0.1	276.5	1.4	3.5
(c) Village	C10	23.0M	31.8M	0.72	360	73	103	1.7	200.1	5.7	17.0
(d) Building	P20	10.3M	15.3M	0.67	98	50	64	3.3	124.3	8.0	12.7
(e) Well	2	5.2M	5.3M	0.98	6	15	15	0.4	274.5	22.9	12.8
(f) Facade	C10	1.1M	1.7M	0.66	65	44	148	0.5	139.4	34.6	17.5

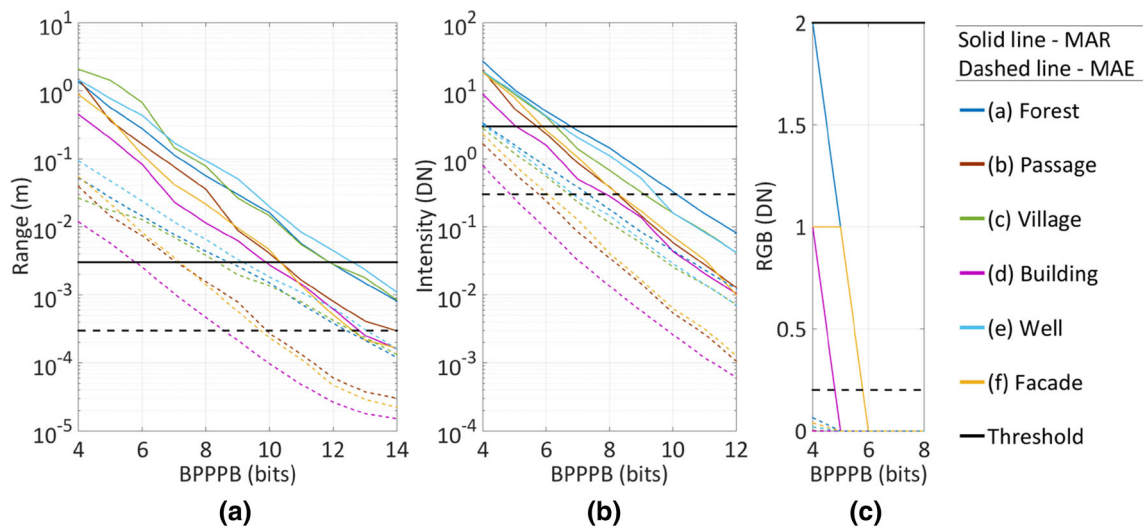


Fig. 3 Distortion of **a** range, **b** intensity, and **c** RGB caused by lossy compression. Compression executed with various expected BPPPB ratios for data arranged in one cube (variant 1). Distortion was mea-

sured as Maximal Absolute Residual (MAR) and Mean Absolute Error (MAE) between original and decompressed data. Colours indicate 6 data sets while black lines show the thresholds of acceptable distortion

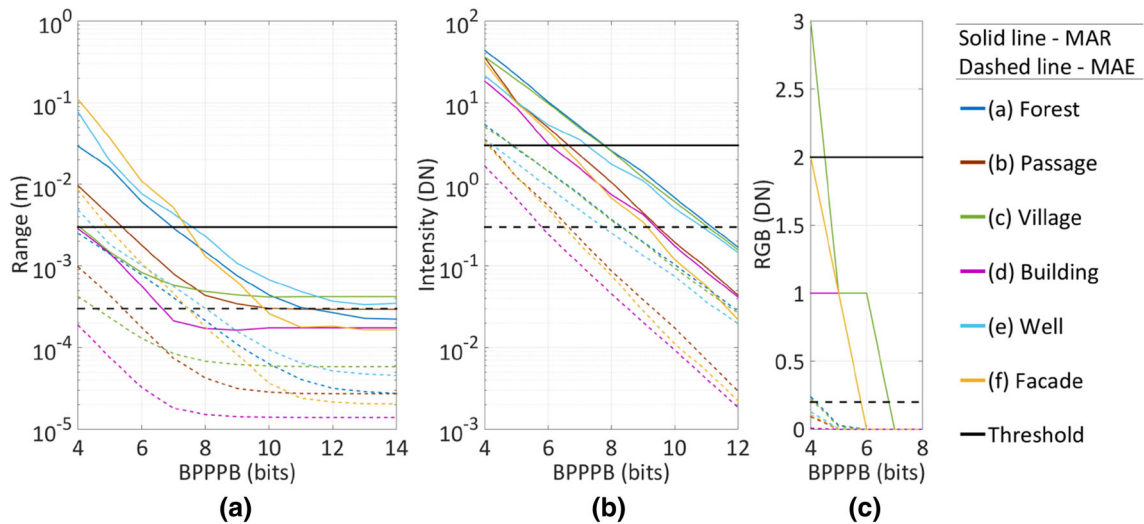


Fig. 4 Distortion of **a** range, **b** intensity, and **c** RGB caused by lossy compression. Compression executed with various expected BPPPB ratios for data arranged in two cubes (variant 2). Distortion was mea-

sured as Maximal Absolute Residual (MAR) and Mean Absolute Error (MAE) between original and decompressed data. Colours indicate 6 data sets while black lines show the thresholds of acceptable distortion

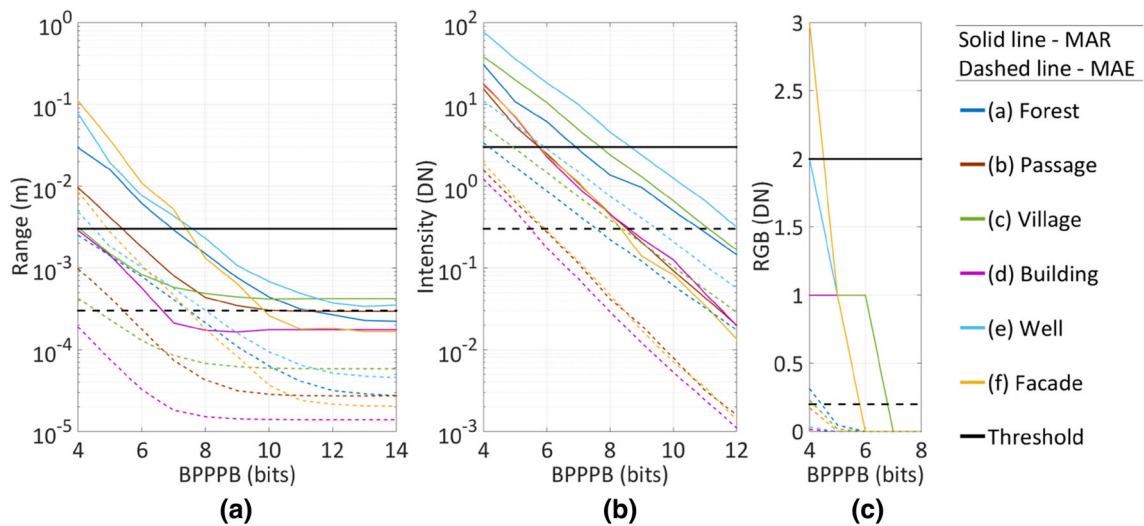


Fig. 5 Distortion of **a** range, **b** intensity, and **c** RGB caused by lossy compression. Compression executed with various expected BPPPB ratios for data arranged in three cubes (variant 3). Distortion was mea-

sured as Maximal Absolute Residual (MAR) and Mean Absolute Error (MAE) between original and decompressed data. Colours indicate 6 data sets while black lines show the thresholds of acceptable distortion

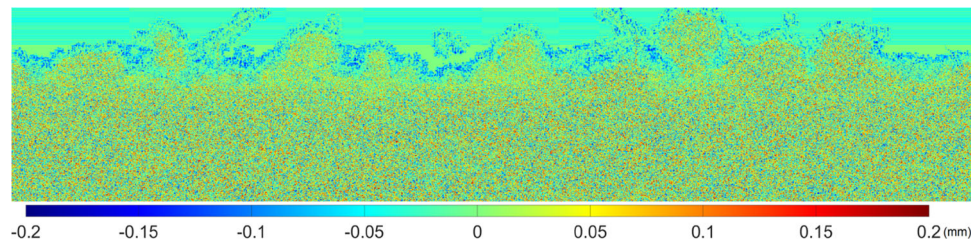


Fig. 6 Range residuals between original and reconstructed from compressed images data. Residuals obtained for (c) Village data set compressed in a lossy manner according to variant 3 with an expected

BPPPB ratio equal to 8. The range of obtained residuals is between -0.40 and 0.41 mm. In the figure, residuals lower or larger than -0.2 or 0.2 mm were rounded to -0.2 and 0.2 mm, respectively

lossy compression does not cause compression artefacts on image areas covered by points. However, errors on the edge between area containing and not containing points (range equal to 0) are visible in Fig. 6. Any distortion that occurs on areas without points is easy to detect and remove, because the corresponding range is shorter than 1 mm and, thus, cannot be measured by terrestrial laser scanners. Note that the obtained residuals (Fig. 6) contain both compression distortion and round-off errors caused by encoding ranges into integers prior to compression and decoding them into floating point numbers after decompression.

4.2 Compression Strength

The efficiency of the TLS data compression strategy proposed here was evaluated by comparing the average BPP ratio obtained for JPEG-2000 based compression (both: lossless and lossy schemes) and other TLS data formats, including compressed files. The average BPP ratio for each data format was calculated from about 134M of points arranged in six test

data sets. Since RGB components for TLS data can be treated as auxiliary data, results were provided in two variants: with and without RGB data. The average BPP ratio is shown in Fig. 7. Clearly, both: lossless and lossy JPEG-2000 schemes provided the lowest BPP ratios among all tested compression formats. The average number of bits necessary to store single scanning point without RGB components is equal to about 22 and 18 bits for lossless and lossy JPEG-2000 schemes, respectively (Fig. 7). These values grow to about 35 bits if TLS data contain also RGB information (Fig. 7). Lossily compressed JPEG-2000 files containing TLS data without RGB components need only about 60% of the storage space that is necessary to keep the same data in LAZ files (LASzip). Note that LASzip compression was developed for efficient compression of airborne LiDAR point clouds. However, it is frequently used for lossless compression of TLS point clouds. Because the variability of the information within a single image (scan) has an impact on the strength of lossless compression, the obtained average BPP ratio was different for each of the 6 test data sets. It was observed that LASzip

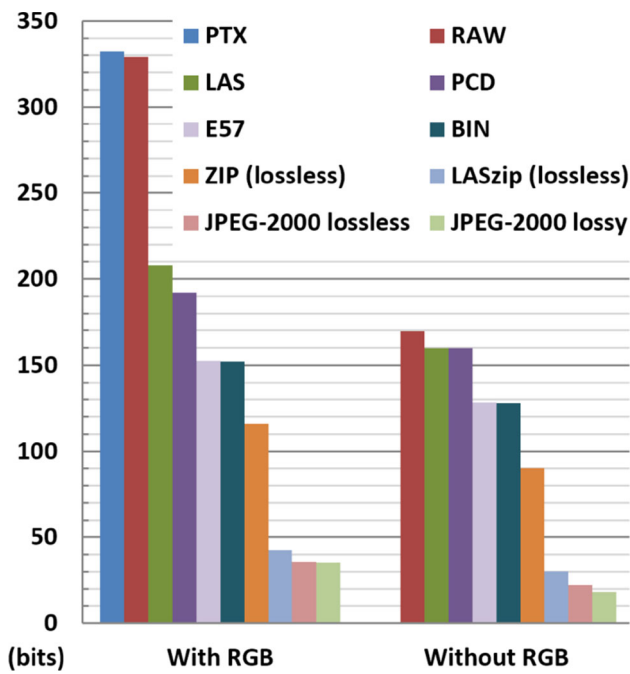


Fig. 7 Average number of bits necessary to store data for single TLS point in various formats. Average calculated from about 134M of points arranged in six test data sets

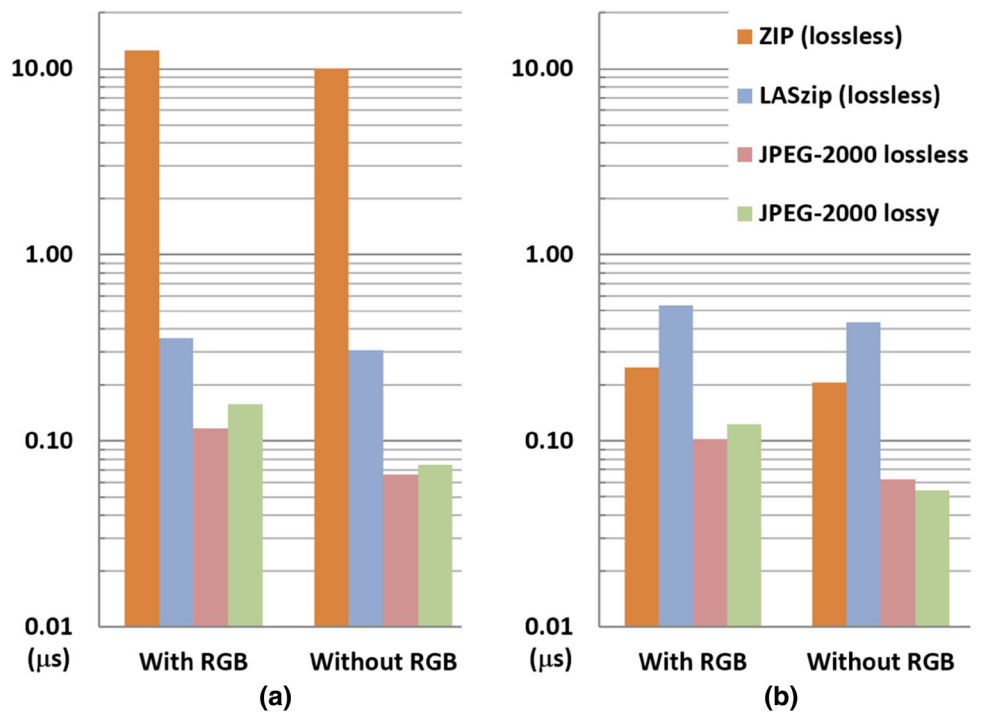
and JPEG-2000 compression generally were strongest for data set (b) Passage, and the weakest for data sets (f) Facade, and (c) Village. This corresponds to range variability where data sets having lower standard deviation of ranges (Table 4) were compressed more efficiently. Obviously, the variability

of intensity and RGB components also influenced average BPP ratio.

4.3 Compression Speed

The last analysed aspects of compression performance were the compression and decompression speed. Similarly as for the compression strength experiment, the analysis was executed for TLS data with and without RGB components. The average time needed to compress or decompress one point using various methods is shown in Fig. 8. The average for each data format was calculated from about 134M of points arranged in 6 test data sets based on the five repeated tests. Differences between repeated tests were lower than 4% and, thus, considered as insignificant. Our results show that in terms of both compression and decompression speed, JPEG-2000 is a few times faster than the LASzip method, though the compression according to the lossy scheme is slightly slower than for the lossless scheme. Note the logarithmic scale in Fig. 8. ZIP compression was the slowest and was about 100 times slower than JPEG-2000. Moreover, an analysis of the speed for individual test data sets showed that the average time needed to compress or decompress one scanning point is not dependent on the total number of points in the data set. Excellent speed of JPEG-2000 compression can be shown by calculating the number of points compressed during one second. Even the slowest lossy JPEG-2000 compression of TLS data containing RGB components allowed to compress more than 6 million points in one second. It should be emphasized that JPEG-2000 compression was executed in variant 3

Fig. 8 Average time needed to **a** compress, and **b** decompress data for a single TLS point. Average calculated for four various formats from about 134M of points arranged in six test data sets



where range, intensity, and RGB data were kept in separate files; therefore, disk operations could slow down this method more than ZIP or LASzip methods. Although the computing power of a laser scanner is lower than for a desktop computer, the obtained compression speed allows to state that JPEG-2000 compression can be used during data acquisition in real or near real time.

5 Conclusions

This work investigated the feasibility of sensor-level TLS data compression using JPEG-2000 lossless and lossy schemes. The TLS data were treated as an image or images consisting of range, intensity, and, optionally, RGB bands (components). It was found that creating a multi-band image is not beneficial due to the different dynamic range of its components. The investigation focused on performance assessment in terms of strength of storage space reduction, data distortion due to lossy compression, and compression and decompression speed. The first measure of the performance showed that JPEG-2000 compressed TLS data require significantly less storage space than the same data stored in other, including compressed, formats. Experiments executed on various TLS data sets showed that in average, only about 22 and 18 bits are required to store range and intensity for a single scanning point that is compressed using JPEG-2000 lossless and lossy schemes, respectively. These values are about 13 and 17 bits larger if RGB components from the camera are added to each scanning point. A very strong storage space reduction in lossy scheme was achieved with insignificant data distortion that was below the level of measurement error for typical terrestrial laser scanners. Finally, JPEG-2000 was the fastest among the tested compression and decompression methods. It allowed to compress more than million scanning points containing RGB components during one second. This proves that JPEG-2000 is fast enough to be applied in real time or near real time TLS data compression at the sensor-level. Although experiments were executed on data collected by pulsed laser scanners, the described compression methodology is also suitable for continuous wave phase scanners. It is not relevant whether scanning points are created by separate laser pulses or by sampling a continuous wave—the principle of arranging points in the image for both types of scanners is identical.

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- ASPRS, 2008: LAS Specification, Version 1.2. http://www.asprs.org/wp-content/uploads/2010/12/asprs_las_format_v12.pdf. Accessed 14 Feb 2017
- Cohen A, Daubechies I, Feauveau J-C (1992) Biorthogonal bases of compactly supported wavelets. *Commun Pure Appl Math* 45(5):485–560
- Gamito MN, Dias MS (2004) Lossless coding of floating point data with JPEG 2000 Part 10. In: *Proc SPIE 5558. Applications of digital image processing*, vol XXVII, pp 276–287
- Girardeau-Montaut D (2011) CloudCompare forum—asking for *.bin file for test. <http://www.danielgm.net/cc/forum/viewtopic.php?t=50>. Accessed 14 Feb 2017
- Glennie C (2012) Calibration and kinematic analysis of the Velodyne HDL-64E S2 Lidar sensor. *Photogramm Eng Rem Sens* 78(4):339–347
- Huber D (2011) The ASTM E57 file format for 3D imaging data exchange. In: *Proc SPIE 7864. Three-dimensional imaging, interaction, and measurement*, vol 7864, pp 78640A
- Isenburg M (2013) LASzip. *Photogramm Eng Rem Sens* 79(2):209–217
- Józków G, Toth C, Quirk M, Grejner-Brzezinska D (2015a) Compression strategies for LiDAR waveform cube. *ISPRS J Photogramm Rem Sens* 99:1–13
- Józków G, Toth C, Quirk M, Grejner-Brzezinska D (2015) Study on sensor level LiDAR waveform data compression using JPEG-2000 standard multi-component transform. *PFG Photogramm Fernerkund Geoinf* 3:201–213
- Leica (2007) Leica ScanStation 2 datasheet. https://hds.leica-geosystems.com/downloads/123/hds/hds/ScanStation/brochures-datasheet/Leica_ScanStation%202_datasheet_us.pdf. Accessed 17 Jun 2017
- Leica (2011) Leica ScanStation C10 datasheet. https://hds.leica-geosystems.com/downloads/123/hds/hds/ScanStation%20C10/brochures-datasheet/Leica_ScanStation_C10_DS_en.pdf. Accessed 17 Jun 2017
- Leica (2013) Leica ScanStation P20 datasheet. http://w3.leica-geosystems.com/downloads/123/hds/hds/ScanStation_P20/brochures-datasheet/Leica_ScanStation_P20_DAT_en.pdf. Accessed 17 Jun 2017
- Leica (2015) Cyclone pointcloud export format. Description of ASCII.ptx format. <http://w3.leica-geosystems.com/kb/?guid=5532D590-114C-43CD-A55F-FE79E5937CB2>. Accessed 14 Feb 2017
- Lepley M, Houchin JS, Kasner J, Marcellin M (2009) JPEG 2000 extensions (Part 2). In: Schelkens P, Skodras A, Ebrahimi T (eds) *The JPEG 2000 suite*. Wiley, New York
- Massaro RD, Anderson JE, Nelson JD, Edwards JD (2014) A comparative study between frequency-modulated continuous wave LADAR and linear LiDAR. *Int Arch Photogramm Rem Sens Spat Inf Sci* 40(1):233–239
- Mongus D, Žalik B (2011) Efficient method for lossless LiDAR data compression. *Int J Rem Sens* 32(9):2507–2518
- Pavlov I (2016) 7-zip. <http://www.7-zip.org>. Accessed 14 Feb 2017
- PCL (2017) Documentation. Point cloud library (PCL). http://pointclouds.org/documentation/tutorials/pcd_file_format.php. Accessed 14 Feb 2017
- Rabbani M, Joshi RL, Jones PW (2009) JPEG 2000 core coding system (Part 1). In: Schelkens P, Skodras A, Ebrahimi T (eds) *The JPEG 2000 suite*. Wiley, New York
- Rosenbaum R, Taubman DS (2003) Merging images in the JPEG 2000 domain. *Proc Vis Imaging Image Process (VIIP)* 2003:249–252
- Serra-Sagrístà J, Aulí-Llinàs F (2008) Remote sensing data compression. In: Graña M, Duro RJ (eds) *Computational intelligence for remote sensing*. Springer, Berlin, pp 27–61

- Skodras A, Christopoulos C, Ebrahimi T (2001) The JPEG 2000 still image compression standard. *IEEE Signal Process Mag* 18(5):36–58
- Taubman D, Marcellin M (2002) *JPEG2000: image compression fundamentals, standards and practice*. Springer, Berlin
- Toth C, Józków G (2016) Remote sensing platforms and sensors: a survey. *ISPRS J Photogramm Rem Sens* 115:22–36
- Xu R, Pattanaik SN, Hughes CE (2005) High-dynamic-range still-image encoding in JPEG 2000. *IEEE Comput Gr Appl* 25(6):57–64