# COMPRESSING AND CLASSIFYING LIDAR WAVEFORM DATA

C. Toth [a,*], S. Laky [a,b], P. Zaletnyik [a,b], D. Grejner-Brzezinska [a]

[a] The Center for Mapping, The Ohio State University
470 Hitchcock Hall, 2070 Neil Avenue, Columbus, OH 43210
toth@cfm.ohio-state.edu, dorota@cfm.ohio-state.edu
[b] HAS-BME Research Group for Physical Geodesy and Geodynamics
Budapest University of Technology and Economics, Muegyetem rkp. 3, Budapest, H-1111, Hungary
laky.sandor@freemail.hu, zaletnyikp@gmail.com

**Commission I, WG I/2**

**KEY WORDS:** LiDAR, Full-waveform, Compression, Classification

**ABSTRACT:**

Today's advanced LiDAR systems are able to record the entire laser echo pulse, provided that sufficient data storage is available on the airborne platform. The recorded echo pulses, frequently called waveform data or full-waveform, can then be used to analyze the properties of the reflecting surface, such as classifying objects based on their material signatures; for example, land classification. However, both the efficient storage of waveform data and the waveform-based classification still present formidable challenges. In this paper, solutions based on state-of-the-art numerical methods, including the Discrete Wavelet Transform and Kohonen's Self-Organizing Map, are proposed to carry out these tasks. 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 performance of the proposed method is evaluated using actual waveform data.

## 1. INTRODUCTION

Recent technological developments in LiDAR systems, most importantly the introduction of full-waveform LiDAR systems, make it feasible to record the intensity of the return pulse as a function of time, and thus to acquire additional data about the reflecting surface (Shan and Toth, 2009). From the shape of this return pulse, additional properties of the reflecting surface can be extracted, such as land information, i.e., the classification of the reflecting objects into categories, such as trees, other vegetation, roofs, pavement, etc.; see typical waveform shapes in Figure 1.

Though the general concern about recording the echo pulses is the need for massive storage space, there are other important aspects of the large volume of the recorded waveform data, including the required sensor data transfer speed and on-board storage capacity, the means and speed of transferring the waveform data to the end users, and finally the storage space and computing power needed to process the waveform data. Compressing the captured digital data at sensor level can benefit most of these processes and can make the use of full-waveform data feasible in practice.

There is a plethora of compression and classification methods, used in a large variety of applications. In this paper, the Discrete Wavelet Transform (DWT) is used as the base of the proposed compression algorithm. In Section 2, the selected compression algorithm is discussed in detail, and a comparison to existing widely used lossless compression methods is presented. For classification, besides the LiDAR-derived geometry, the wavelet coefficients are used. It is shown that this data can be efficiently used to separate man-made and natural objects, without the need of inverse wavelet transform (i.e., decompression), thus reducing the needed computation power and time. A non-supervised machine learning approach is selected for our classification (clustering); the algorithm, based on Kohonen's Self-Organizing Map, is discussed in Section 3.

The performance of the developed compression and classification algorithms is validated on a test area in Toronto (Ontario, Canada). A smaller suburban area (Scarborough) of approximately 50 by 65 meters, containing about 6,000 waveforms, and offering a great variety of land coverage, is used for the investigation. The data was acquired by an Optech ALTM 3100 full-waveform LiDAR system. The sampling interval of the digitizer is 1 nanosecond, and the captured signal is digitized by an 8-bit analog-to-digital (A/D) converter, giving a dynamic intensity range of 0-255. The digitizer can cover targets extending over a range of 66 meters (Optech, 2005).

## 2. COMPRESSION USING DWT

### 2.1 Proposed compression algorithm

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

1. Preprocessing of the signal or image to be compressed (e.g., partitioning an image into smaller blocks)
2. Transformation (e.g., discrete cosine transformation, DWT, etc.)
3. Quantization of the coefficients (dropping coefficients based on some criterion, such as the order of magnitude or other properties, and then storing the remaining coefficients in a more compact form)
4. Further lossless compression of the coefficients

In our investigation, a compression scheme following the above-mentioned steps is proposed. Other lossy compression schemes also exist, e.g., Compressive Sampling (CS) is one of

the emerging methods that incorporates the compression mechanism into the sensing mechanism itself. An initial study of applying the CS to full-waveform LiDAR data is reported in (Laky *et al.,* 2010). Our implementation of the proposed compression scheme is based on the GNU Octave (Eaton *et al.,* 2008) and the WaveLab toolbox (Buckheit and Donoho, 1995).

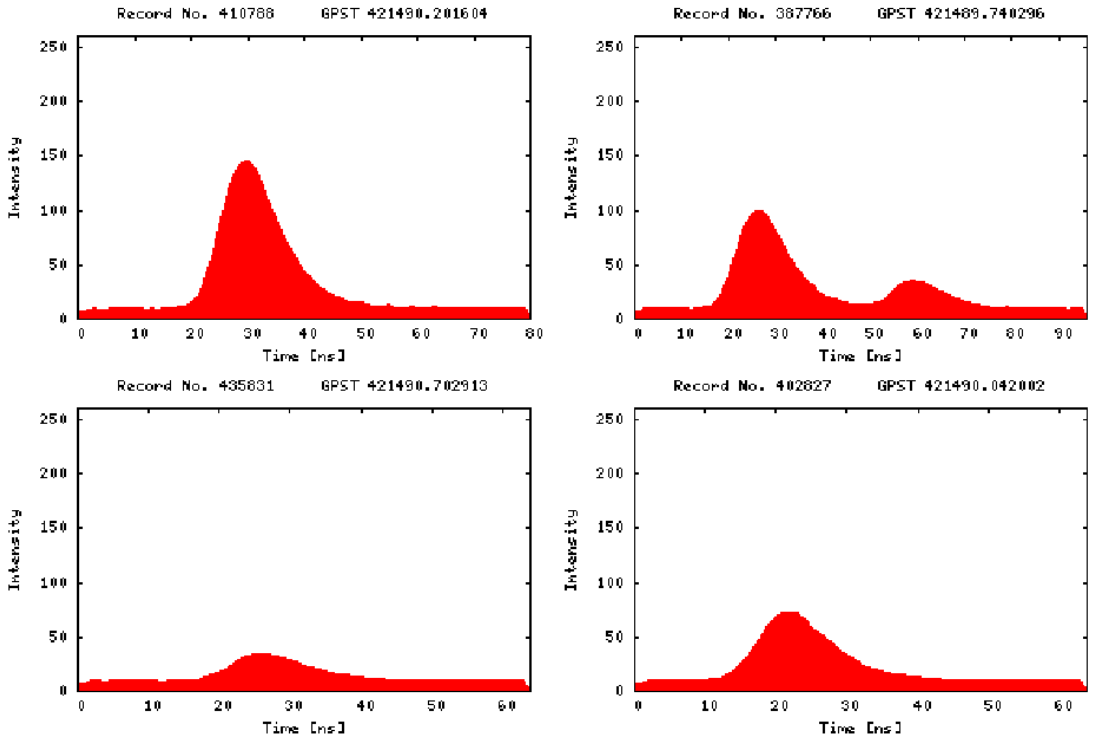


Figure 1. Typical waveforms of different surfaces (Courtesy of Optech Incorporated)

The details and the goals of the specific steps of the proposed compression scheme are the following:

1. *Preprocessing*. The low value data is discarded when the intensity of the return pulse falls below a given threshold. In our case, this threshold is set to 10 intensity units. This step removes the 0-10 unit range from the bottom of the waveform, essentially creating a seamless zero-padding of the data. Since a data record length of the power-of-two is needed by the implementation of the DWT in the WaveLab toolbox, the records are further zero-padded to meet this requirement, resulting in an increase of the data size compared to the original size. In addition, the unusually short waveforms are removed, as those waveforms are likely results of measurement errors.
2. *Discrete wavelet transformation* (Figure 2). As the result of preliminary investigations, the CDF (Cohen-Daubechies-Feauveau) wavelet family with parameters 3 and 9 was chosen for this task (Laky *et al.,* 2010). The earlier research aimed at finding the wavelet family that produces the smallest average reconstruction error around the compression ratio of 20%. Since the original waveform data is stored as unsigned 8-bit integers and transformation produces the resulting wavelet coefficients as 64-bit double-precision floating point numbers, there is an eight-time increase in the data storage space.
3. *Eliminating high-order wavelet coefficients.* Experimental results indicated that keeping only the first 25% of the wavelet coefficients allows for a reasonable small reconstruction error. Obviously, this step also reduces the needed storage space to one fourth.
4. *Eliminating outliers*. It has been observed that in some (rare) cases the intensity of the return pulse exceeds the maximum intensity of the D/A converters, in which case clipping occurs (the top of the return pulse is clipped). As the shape of these waveforms is different from the shape of the other (regular) waveforms, the resulting wavelet coefficients also have a different order of magnitude. Since the information content is low in these situations, the waveforms that have out-of-range wavelet coefficients are simply removed.
5. *Quantization of the wavelet coefficients*. To efficiently store the wavelet coefficients, quantization, i.e., mapping of 64-bit double-precision floating point values to unsigned integer values, is needed; in other words, the distribution of the coefficient values does not require the high resolution of the 64-bit number representation. During quantization, the zeroing out the low magnitude wavelet coefficients also occurs (thresholding). In our experiments, a simple linear quantizer is used, which only requires storing the minimum and maximum wavelet coefficient values and the quantized values. Thus, the dequantized values can be calculated by a simple linear equation. To allow for near real-time compression of the waveforms, the quantization parameters, i.e., the abovementioned minimum and

maximum values, are calculated for every 50 waveforms. Consequently, a temporary storage for the wavelet coefficients of maximum 50 waveforms is needed. Another important aspect of the quantizer is that it must have the capability to store exactly zero values; otherwise, these exactly zero elements would be later dequantized to the nearest level, adding considerable noise to the reconstructed waveform. The last parameter of the quantization is the granularity of the quantizer, i.e., the number of integer levels, usually set to a power of two, so the maximum range is utilized for a given number of bits.

6. *Run length encoding of the quantized coefficients*. After quantization, a high number of consecutive zeros remain in the quantized wavelet coefficients, thus run length encoding is used to replace these consecutive zeros with a short code.
7. *Creating a continuous stream of the compressed data*. This is needed to avoid storing complex data types (matrices) in the output file. Additional header data, such as length of the quantized wavelet coefficient vector and length of the original waveform, is appended to the quantized wavelet coefficients of every waveform, making operations, such as seeking in the compressed file and reconstruction of the waveform, possible. Note this step increases the storage need of the data.
8. *Huffman coding of the data stream*. Traditional representation of numerical integer data utilizes fixed-length codes. E.g., when using an 8-bit quantizer, all the 256 possible values are stored as a group of 8 binary digits. However, the frequency of these values are not even. Therefore, assigning shorter codes to more frequent values, and longer codes to less frequent values, lossless compression can be achieved, which is the basic idea of Huffman coding. Note the lookup table of these Huffman codes must also be stored.

Table 1 summarizes the specific steps of the proposed compression scheme. The configuration parameters of the method that can be changed depending on the need for better compression rate or better reconstruction (decompression) quality are also outlined in the table. Furthermore, the ratio of the storage size of the compressed data versus the storage size of the original data, the compression rate, is also shown in the last column.

| Step # | Description | Parameters | Compression rate |
|---|---|---|---|
| 1 | Preprocessing: decreasing the intensity values by a given amount (the threshold of the digitizer), removing unusually short waveforms, and zero-padding to the length of the next power-of-two (required by WaveLab) | Amount of decrease: 10 intensity units, minimum length: 16 nanoseconds | 100.9% |
| 2 | Discrete wavelet transform | Wavelet family: CDF/3/9 | 806.8% |
| 3 | Eliminate high-level wavelet coefficients | Coefficients to keep: first 25% | 201.7% |
| 4 | Eliminate outliers | Valid range of wavelet coefficients: -360 .. +650 | n/a |
| 5 | Quantization of wavelet coefficients | Quantization threshold: 5.0, granularity of quantizer: 8 bits (256 levels), quantization parameters stored for every 50 waveform | 22.2% |
| 6 | Run length encoding of the quantized coefficients | n/a | 18.5% |
| 7 | Creating continuous stream of the compressed data | n/a | 21.3% |
| 8 | Huffman coding of the data stream | n/a | 18.7% |

Table 1. Steps of the proposed compression scheme

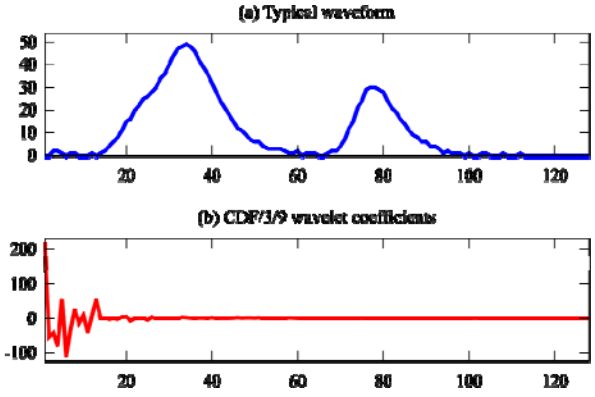


Figure 2. LiDAR waveform and its wavelet coefficients

### 2.2 Reconstruction quality

The reconstruction scheme consists of the inverse of every step in the compression scheme, applied in reverse order. The quality of the reconstruction is a function of four parameters:

1. The chosen number of wavelet coefficients to store
2. The number of quantization levels (bits)
3. The quantization threshold
4. How often we store the minimum and maximum values of the quantizer

In Table 2, the performance of the compression and the quality of the reconstruction are examined as a function of the quantization threshold; see step (5) of the proposed compression algorithm. All the other parameters remain as fixed values. First, by subtracting the decompressed waveforms from the preprocessed waveforms, an error signal is calculated;

waveforms deemed to be outliers in step (4) of Table 1 are excluded. Then, the standard deviation of the error and the maximum absolute error are calculated for all the ~6,000 waveforms. In Table 2, the minimum, mean and maximum of these values are listed, along with the compression rate achieved.

| Threshold | Minimum std. | Mean std. | Maximum std. | Minimum abs. max. | Mean abs. max. | Maximum abs. max. | Comp. rate |
|---|---|---|---|---|---|---|---|
| 1.0 | 0.30 | 0.70 | 1.41 | 0.85 | 2.56 | 7.30 | 22.3% |
| 3.0 | 0.34 | 0.73 | 1.42 | 0.85 | 2.62 | 7.35 | 19.7% |
| 5.0 | 0.37 | 0.82 | 1.47 | 0.94 | 2.81 | 7.35 | 18.7% |
| 7.0 | 0.40 | 0.95 | 1.87 | 0.94 | 3.13 | 7.35 | 17.8% |
| 9.0 | 0.40 | 1.13 | 2.50 | 1.07 | 3.54 | 7.73 | 17.1% |

Table 2. Reconstruction error statistics as a function of the quantization threshold

### 2.3 Comparison with common lossless compression schemes

The performance of the proposed lossy compression scheme is compared to widely-used lossless compression schemes. To test these compression methods, a raw binary data file has been created from the waveform data, appending the length of each waveform data as header information to every waveform. The resulting file then is compressed using three different programs: Info-ZIP, GNU GZIP and BZIP2. Info-ZIP, a ZIP implementation compatible with the well-known PKZIP program, and GNU GZIP both utilize the Lempel-Ziv coding (LZ77) and Huffman coding. The LZ77 coding replaces repeating sequences in the file that occur at different places with references to their first appearance (within a given window size). These two methods applied after each other is also known as the Deflate algorithm. The BZIP2 uses the Burrows-Wheeler transform, which changes the order of the bytes so that the data can be compressed more efficiently later, followed by the move-to-front transform, which further increases the compressibility, and finally utilizes Huffman coding. Table 3 summarizes the results obtained by these methods. All of these programs have options to choose between quick execution and higher compression rates. The tests were carried out using the two extreme settings (maximal speed or maximal compression).

| Program | Compression rate (maximum speed) | Compression rate (maximum compression) |
|---|---|---|
| Info-ZIP | 46.7% | 38.6% |
| GNU GZIP | 46.7% | 38.6% |
| BZIP2 | 26.2% | 23.9% |

Table 3. Performance of various widespread lossless compression schemes

Clearly, the performance of BZIP2 approaches the performance of our compression scheme. The significantly better performance of BZIP2 is due to using more advanced lossless transforms, compared to the competitors, before utilizing Huffman coding. Of course, this performance gain does not come without a price, as the computation requirement of the compression increases significantly; comparing BZIP2 and GZIP, there is a 3.5 times difference between the runtimes.

Generally speaking, lossless compression methods do not have the choice between reconstruction quality and compression performance, as they always give perfect reconstruction, and offer only a trade-off between computation requirement and compression performance. Using the proposed lossy compression method, a choice to improve the compression rate is always given at the cost of decreasing the reconstruction quality.

## 3. CLASSIFICATION USING WAVELET COEFFICIENTS

### 3.1 LiDAR classification methods

LiDAR measurements are widely used in digital elevation model generation and object extraction, though can also be used for land classification purposes. In particular, the full-waveform LiDAR technology provides the possibility of further analyzing the shape of the waveform and, thus, obtaining additional information about reflecting objects and their geometric and physical characteristics.

There are a large number of studies addressing LiDAR data classification. The first classification algorithms used only LiDAR-derived geometric characteristics of a point relative to its neighborhood (Maas, 1999; Vögtle and Steinle, 2003). Beyond the geometry, Tóvári and Vögtle (2004) also used the intensity values to classify objects. Others used not only LiDAR-derived data, but aerial imagery or an independent digital elevation model to improve the classification performance (Brattberg and Tolt, 2008, Charaniya *et al.,* 2004).

Since the introduction of full-waveform LiDAR systems, there have been other possibilities to improve the efficiency of the classification of surface objects, as the entire waveform can be analyzed to derive additional information. Numerous studies are based on decomposing the waveform into a sum of components or echoes to generate a denser and more accurate 3D point cloud (Mallet and Bretar, 2009), modeling the waveforms with Gaussian (Wagner *et al.,* 2006), Generalized Gaussian or Lognormal function (Chauve *et al.,* 2007). Ducic *et al.,* (2006) decomposed the waveform into Gaussian components, and used this not only to improve the accuracy of the peak detection, but also to discriminate between vegetation and non-vegetation points using the parameters of the Gaussian functions, such as the amplitude and the standard deviation (pulse width). Mallet

*et al.,* (2008) used the Generalized Gaussian model for classifying urban areas with one more parameter, the shape parameter of the Generalized Gaussian model, which allows simulating Gaussian, flattened or peaked pulses too. Both Ducic and Mallet used supervised classification methods.

In our earlier studies, we used a non-supervised classification method, the Kohonen's Self-Organizing Map (SOM) (Kohonen, 1990), based on various statistical parameters describing the shape of the waveform, such as the maximum value of the intensity (amplitude), standard deviation (pulse width), skewness and the kurtosis (Zaletnyik *et al.,* 2010). In this paper a new set of input parameters is proposed for the SOM-based classification.

### 3.2 Proposed classification algorithm

A SOM is formed of neurons located on a regular, usually one- or two-dimensional grid, and represents the result of a vector quantization and projection algorithm defining a nonlinear projection from the input space to a lower-dimensional output space. A typical application of SOM is in the analysis of complex vector data where the data elements may be related to each other even in a highly nonlinear fashion. The process, in which the SOM is formed, is an unsupervised learning process (Kohonen *et al.,* 1996). In the proposed algorithm, apart from the basic LiDAR dataset and the geometry derived from the dataset, the first 16 wavelet coefficients are used.

The SOM defines a mapping from the input data space $\mathbf{R}^n$ onto a regular two-dimensional array of neurons. Each neuron i in the grid has an associated $d$-dimensional code vector $\boldsymbol{m}_i = [m_{i1}\ m_{i2}\ \ldots\ m_{id}]$, where $d$ is equal to the dimension of the input vectors. The lattice type of the array can be defined as rectangular or hexagonal. An input vector $x \in \mathbf{R}^n$ is compared with each $m_i$ code vector and the best match is defined as a response to the input, and thus, is mapped onto this location. During the training the code vectors change so that they follow the probability density of the input data.

To implement the classification scheme, GNU Octave and the SOM_PAK software (Kohonen *et al.,* 1996) were used. The steps of the proposed algorithm are the following:

1. The first 16 wavelet coefficients of the preprocessed waveforms (see step (2) Table 1) are fed to the SOM. In our implementation, the size of SOM is 2×2, the nodes are rectangular, the neighborhood function is "bubble" type (SOM terminology for rectangular neighborhood function), and the learning rate function is linear. After this step, the points are categorized into 4 types, coded as 0-3 (see Figure 3b).
2. The number of return pulses is extracted from the basic LiDAR data.
3. The range of the first return pulse is extracted from the basic LiDAR data. Using these ranges, the local range differences are then calculated. That is the difference of a given range from the maximal range of the points in a given radius. This local range difference roughly corresponds to the local height difference, i.e., how much the given point protrudes from or extends beyond its surrounding area. Using this value, the points are classified into two categories: "surface" points with a local range difference less than 3 meters, and "protruding" points, (see Figure 3a).
4. Using the information collected in steps (1-3), a decision is made for every point (see Figure 5a).
    a. If the SOM class is 2, or the number of peaks is not one, then the classification of the point is tree[1].
    b. If the SOM class is 3, then the classification of the point is grass.
    c. If the SOM class is 0 or 1, and the point is a "protruding" point, then the classification of the point is roof.
    d. If the SOM class is 0 or 1, and the point is a "surface" point, then the classification of the point is pavement.
5. To enhance the classification, a mode filter is applied to the result[2]. Basically, for every point, the class that occurs most frequently in a given radius is applied to it. In our implementation, the radius to be considered is 1 meter (see Figure 5b).

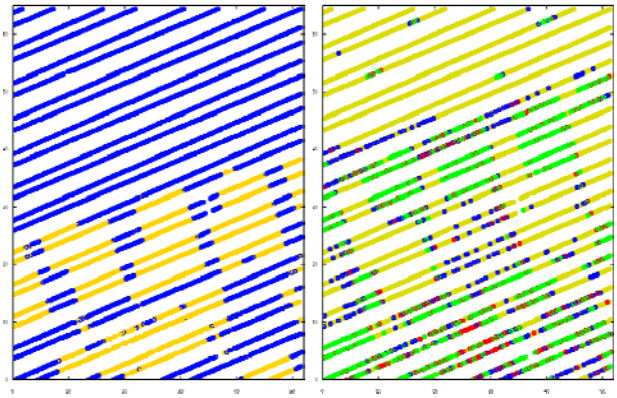

Figure 3. Classification (a) using local range differences and (b) using SOM

### 3.3 Evaluation of the proposed algorithm

For evaluation purposes, 700 randomly-picked points were manually classified on the test area (see Figure 4). To facilitate the manual classification, a Google Maps API-based application has been developed. The LiDAR data has been stored in a PostgreSQL database, and the server-side components have been written in Perl.

In Tables 4 and 5, the manual and the SOM-based classifications are compared before and after mode filtering. As the sum of the percentages in the diagonal shows, 88.9% of the points have been correctly classified after mode filtering; before mode filtering, this was 82.6%. After mode filtering, the worst case of the misclassification is 6.1% of the points, which have

[1] Most of the multiple-echo waveforms were backscattered from trees, only a few from the edges of the buildings. In this step, every multiple-echo waveform point is assumed to be tree. The edge points are sparsely located, and, therefore, in the following step, mode filtering, are eliminated.

[2] Mode (or modal) filters are used in thematic map post processing to reduce class speckle. The mode filter replaces isolated, single classified pixels with the mode class found within a small area around the point. It has been shown to increase classification accuracy and reduce noise (Bradley *et al.,* 1994).

been manually classified as tree, but classified as grass in the SOM approach. Note that this can also be due to changes in the canopy between the time of taking the image used for manual classification and the time of the LiDAR measurement.

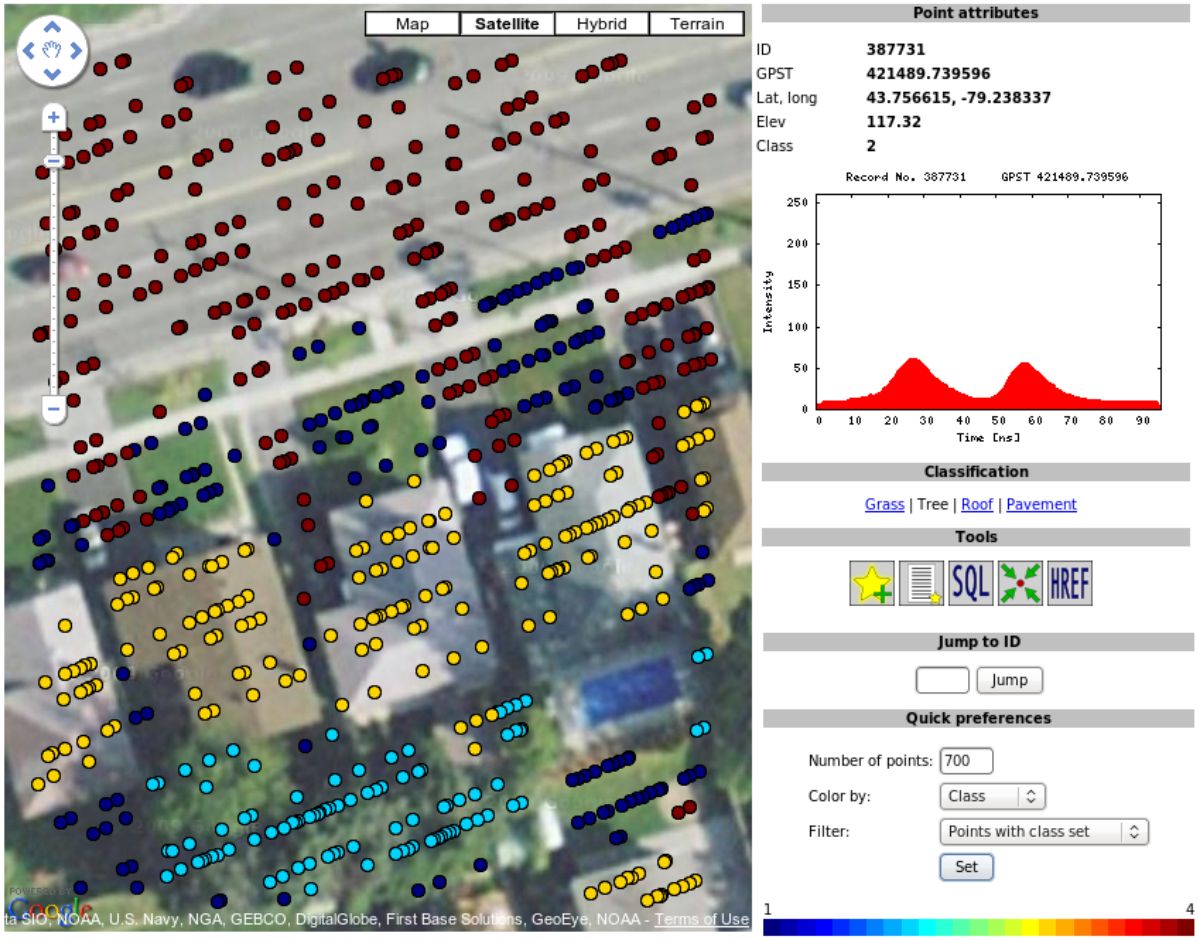


Figure 4. Google Maps-based application for manual classification

| | Grass | Tree | Roof | Pavement |
|---|---|---|---|---|
| Grass | 17.0% | 3.7% | 0.1% | 1.9% |
| Tree | 4.9% | 5.6% | 2.9% | 0.3% |
| Roof | 0.3% | 0.6% | 25.3% | 0.1% |
| Pavement | 0.3% | 2.3% | 0.1% | 34.7% |

Table 4. Comparison of the manual and the SOM-based classification before mode filtering. Rows: manual classification, columns: SOM-based classification. Numbers are the percentages of the manually-classified points in the respective categories

| | Grass | Tree | Roof | Pavement |
|---|---|---|---|---|
| Grass | 20.4% | 0.3% | 0.1% | 1.9% |
| Tree | 6.1% | 7.0% | 0.4% | 0.0% |
| Roof | 0.4% | 0.0% | 25.9% | 0.0% |
| Pavement | 0.6% | 1.3% | 0.0% | 35.6% |

Table 5. Comparison of the manual and the SOM-based classification after mode filtering. Rows: manual classification, columns: SOM-based classification. Numbers are the percentages of the manually-classified points in the respective categories

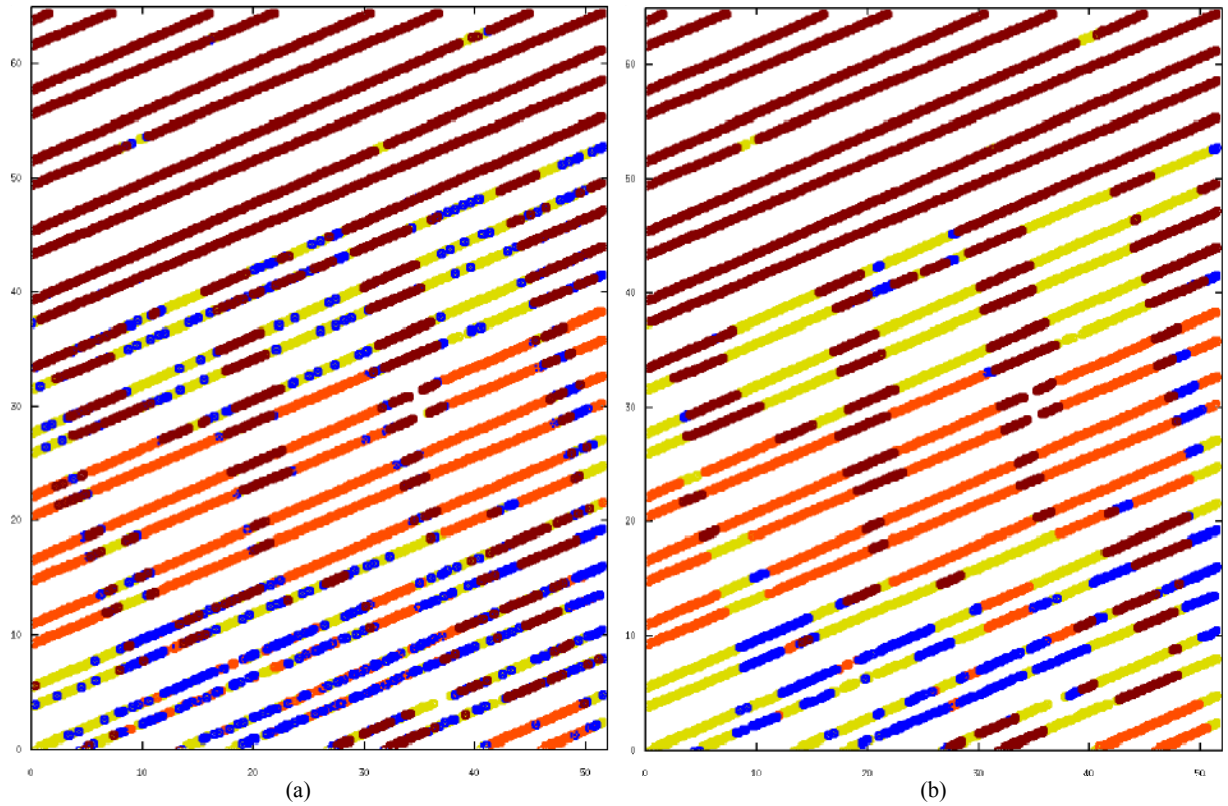


Figure 5. Classification results, (a) before and (b) after mode filter

## 4. SUMMARY

In this study, we have examined the compression and classification of full-waveform LiDAR data. A DWT-based compression algorithm has been developed and the resulting wavelet coefficients have also been used for classification purposes. Our compression method has been compared with commonly used lossless compression schemes. For classification, an unsupervised method (SOM) has been used that requires no training set, though for evaluation purposes randomly-picked points have been manually classified in the test area.

The proposed DWT-based compression scheme consists of the following steps: preprocessing, discrete wavelet transform, lossy compression (elimination of small wavelet coefficients, elimination of outliers, quantization) and lossless compression (run length encoding and Huffman coding). With this method an 18.7% compression rate has been achieved (i.e., the data was compressed to less than one-fifth of its original size) with only a 0.8 intensity value standard deviation of the reconstruction error (difference between the original and the reconstructed signal). Widespread lossless compression schemes, such as ZIP, GZIP and BZIP2, have also been applied to the test waveform data for comparison. With ZIP and GZIP a 39% compression rate can be achieved, while BZIP2 with 24% approaches the performance of our compression scheme. Using the proposed lossy compression method, the compression rate can be further improved at the cost of decreasing the reconstruction quality.

For classification, the first 16 wavelet coefficients have been used apart from the geometry derived from the basic LiDAR dataset. The chosen clustering algorithm is the Kohonen Self-Organizing Map, with 2×2 neurons in a rectangular lattice. To enhance the classification, a mode filter is applied to the result. In our investigation, the points are categorized into 4 types: tree, grass, roof, and pavement. Using 700 (12% of the full data set) manually classified reference points, the performance of the unsupervised classification has been validated at a success rate of 88.9%.

**Acknowledgements**

The authors thank Optech Incorporated for the data provided for this research. The third author wishes to thank to The Thomas Cholnoky Foundation supporting her visit at The Center for Mapping at The Ohio State University, during the time-period this work has been accomplished.

**References**:

Bradley A. W., Ow, C. F. Y. , Heathcott, M., Milne, D., McCaffrey, T. M., Ghitter, G., Franklin, S. E., 1994. Landsat MSS Classification of Fire Fuel Types in Wood Buffalo National Park, Northern Canada; *Global Ecology and Biogeography Letters*, Vol. 4, No. 2. pp. 33-39.

Brattberg, O.; Tolt, G., 2008. Terrain Classification Using Airborne LiDAR Data and Aerial Imagery, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*. Vol. XXXVII. Part B3b. Beijing 2008, pp. 261-266.

Buckheit, J. B. and Donoho, D. L., 1995. Wavelab and reproducible research. Wavelets and Statistics, Springer-Verlag.

Charaniya, A.; Manduchi, R; Lodha, S., 2004. Supervised Parametric Classification of LiDAR Data, IEEE Workshop on Real Time 3D Sensor and Their Use, Washington DC, June 2004, 8 pages.

Chauve, A., Mallet, C., Bretar, F., Durrieu, S., Pierrot-Deseilligny, M., Puech, W., 2007. Processing full-waveform LiDAR data: Modelling raw signals. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 36 (Part 3/W52), pp. 102-107.

Ducic, V., Hollaus, M., Ullrich, A., Wagner, W., Melzer, T., 2006. 3D vegetation mapping and classification using full-waveform laser scanning. In: Proc. Workshop on 3D Remote Sensing in Forestry. EARSeL/ISPRS, Vienna, Austria, 14-15 February 2006, pp. 211-217.

Eaton, J. W., Bateman, D., Hauberg, S., 2008. GNU Octave Manual Version 3, Network Theory Ltd., United Kingdom

Kohonen, T., 1990. The Self-Organizing Map, Proc. IEEE, Vol. 78, No. 9, pp. 1464-1480.

Kohonen, T., Hynninen, J., Kangas, J., Laaksonen, J. 1996. SOM_PAK, The Self-Organizing Map Program Package, Technical Report A31, Helsinki University of Technology, Laboratory of Computer and Information Science, FIN-02150 Espoo, Finland, 27 pages.

Laky, S., Zaletnyik, P., Toth, C. 2010. Compressing LiDAR waveform data, In: Proc. International LiDAR Mapping Forum, Denver, USA, 3-5 March 2010.

Maas, H.G., 1999. Fast determination of parametric house models from dense airborne laserscanner data, *International Archives of the Photogrammetry and Remote Sensing*, Bangkok, Thailand, Vol. XXXII, 2/W1, pp. 1-6.

Mallet, C., Soergel, U., Bretar, F., 2008. Analysis of full-waveform lidar data for an accurate classification of urban areas. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences* 37 (Part 3A), pp. 85-92.

Mallet, C., Bretar, F., 2009. Full-waveform topographic LiDAR: State-of-the-art, *ISPRS Journal of Photogrammetry and Remote Sensing*, Volume 64, Issue 1, pp. 1-16.

Optech Incorporated, 2005. Airborne Laser Terrain Mapper (ALTM) Waveform Digitizer Manual, Optech Incorporated, Toronto, Ontario, Canada, Document No. 0028443/Rev A.5.

Shan, J., and Toth, C. K., 2009. *Topographic Laser Ranging and Scanning—Principles and Processing*, CRC Press Taylor & Francis, London, 590 pages.

Tóvári, D., Vögtle, T., 2004. Object Classification in Laserscanning Data, *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, Freiburg, Germany, Vol. XXXVI, Part 8/W2, pp. 45-49.

Vögtle, T., Steinle, E., 2003. On the quality of object classification and automated building modeling based on laserscanning data, *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, Dresden, Germany, Vol. XXXIV, 3/W13, pp. 149–155.

Wagner, W., Ullrich, A., Ducic, V., Melzer, T. and Studnicka, N., 2006. Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner. *ISPRS Journal of Photogrammetry & Remote Sensing* 60(2), pp. 100–112.

Zaletnyik, P., Laky, S., Toth, C., 2010. LiDAR waveform classification using Self-Organizing Map, ASPRS, San Diego, CA, 28-30 April 2010.