

Basic Full Waveform LiDAR Tools

Emma Carolan ARSF-DAN

Mark Warren ARSF-DAN

1 Introduction

This practical aims to introduce you to a number of full waveform LiDAR tools in order to view and access the data stored in the LAS1.3 files. We will also show you one method that can be used to extract extra information from the full waveform files.

Unfortunately there are a limited number of freely-available tools for full waveform processing.

2 Software

2.1 PulseWaves

Similar to and driven by the same developer as LASTools, PulseWaves is an open-source library of tools designed to manage and view full waveform LiDAR data. Although it has its own internal file format it is also capable of directly reading LAS1.3 files.

There are just three tools available at present (sign up to the google group at <http://pulsewaves.org> or check the website given below for any updates). The tools are available as a stand alone GUI or command line functions.

Website and installation via: <http://rapidlasso.com/category/pulsewaves/>

Running under Linux

In order to run the PulseWaves tools under Linux you will need make the .exe files executable. In the pulsewavesDLL folder run:

```
chmod a+x *.exe
```

You will also need to have wine installed on your computer.

Download: <http://www.winehq.org/>

2.2 Full waveform extraction tool

The full waveform extraction tool written by ARSF-DAN to extract individual waveforms from LAS1.3 files. It requires the user to specify an area of interest (N,S,E,W) and it creates an ascii file containing the waveform information for each point within the specified area (one file for each point).

In order to use this tool you will need to have python installed on your computer.

Download: <http://www.python.org/downloads/>

You will also need to add the directory of your Python distribution to your PATH environment variable.

More information: <http://docs.python.org/2/using/windows.html#excursus-setting-environment-variables>

The fw_extract.py script can be downloaded from the ARSF github page: https://github.com/pmlrsg/arsf_tools

2.3 Fityk

Fityk offers various non-linear fitting methods, simple background subtraction and other data manipulations, support for analysis of series of datasets and automation of common tasks with scripts. It is freely available up to version 0.9.8.

Subsequent versions require a license to access the software.

Manual: <http://fityk.nieto.pl/fityk-manual.html>

Downloads:

Windows - <https://github.com/wojdyr/fityk/downloads/>

Linux - <http://download.opensuse.org/repositories/home://wojdyr/>

3 Example Data

The data used in this practical were collected on the 26th of July 2012 by the ARSF over Milton Keynes, approximate location Lat 52°03' 12.89" Lon +0°47' 9.87".

External File access by download



Figure 1: Location of test data - ARSF Eagle Image

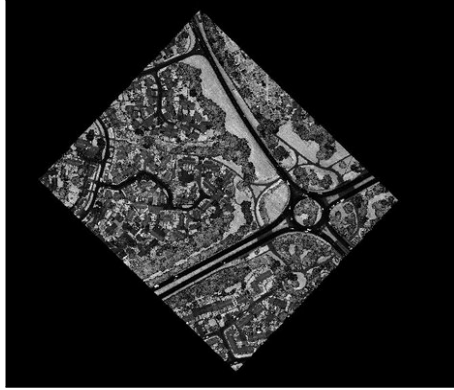


Figure 2: Location of test data - ARSF LiDAR Intensity Image

If you are not using our computers then the data is still available from our download site (See Appendix for further details).

http://arsf-dan.nerc.ac.uk/files/full_waveform_tutorial_data.LAS

Using PulseWaves we will load in the LAS1.3 files and visualise the associated waveform data. We will then go on to extract some waveforms for a small area of the site using the full waveform extraction tool. Fityk will then be used to identify peaks from the extracted waveforms.

4 PulseWaves

This is a basic viewing tool for full waveform data, and gives an idea of what is contained within the LAS1.3 files.

4.1 Launching pulseview and loading in a file

Load in the file LDR120726_094707_4.LAS using either the command-line or the GUI instructions below:

Command-Line Instructions:

```
pulseview.exe -i <input_las_file.LAS>
```

To get help or further instructions run:

```
pulseview.exe -h
```

Loading using the GUI:

Find the downloaded pulsewavesDLL folder and double click on the pulseview.exe file.

Now using the GUI load in the LAS1.3 file which we want to view. Use the browse tab to navigate to the folder containing the data and double click on the file required. Then select VIEW from the top right corner.

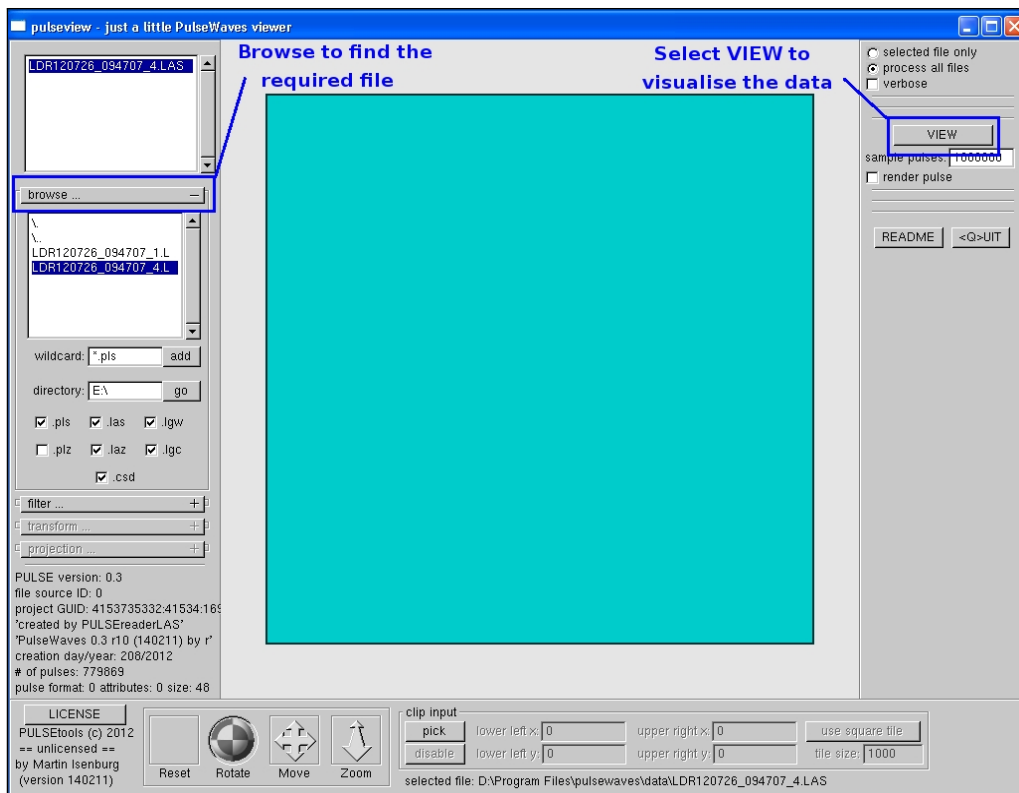


Figure 3: pulseview load using the GUI

4.2 Exploring the data

Use the mouse to navigate the data. You can change the view settings using the following options:

- <space> switch between tilt/translate/zoom/pan
- <c> change color mode
- <i> select a point
- <W> turn the waveform display on/off

Further information is available in the pulseview_README.txt file, contained in the pulsewavesDLL folder.

You can also right-click to bring up a menu and select options from there.

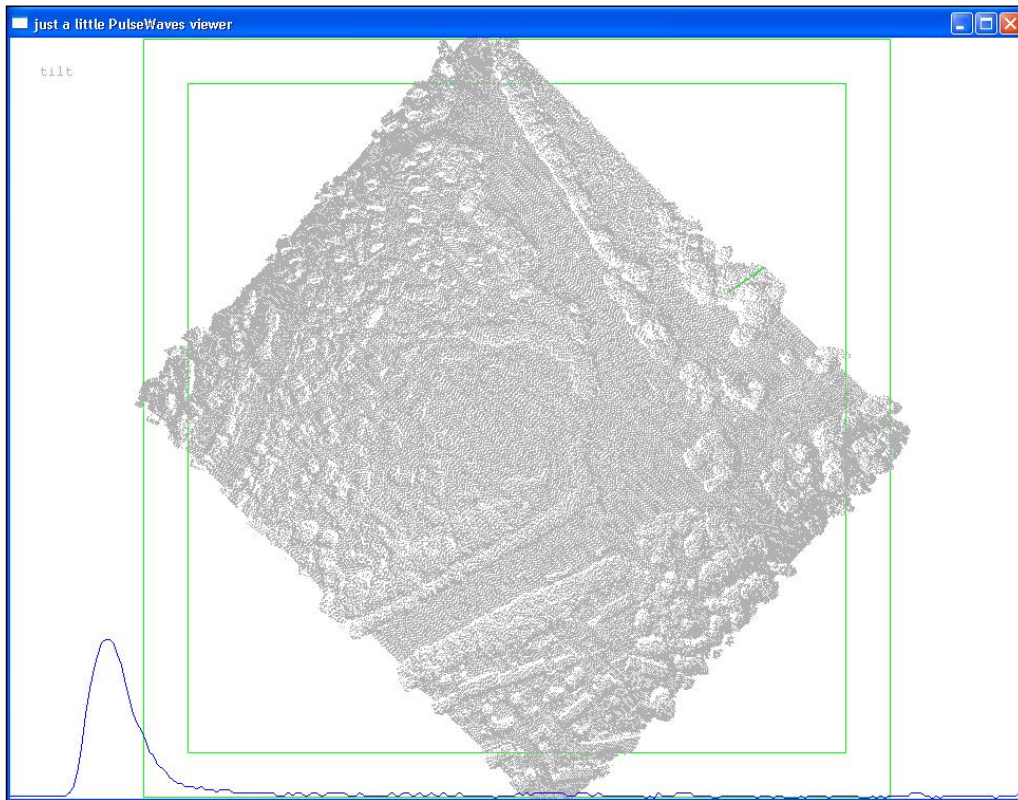


Figure 4: pulseview initial view

5 Full waveform extraction

The full waveform extraction tool does not have an associated GUI so needs to be run from the command line.

To run the extraction tool under Linux type

```
fwf_extract.py <input_las_file.LAS>
```

Alternatively to run under Windows precede the command with 'python', e.g.

```
python fwf_extract.py <input_las_file.LAS>
```

Run the command using the example file LDR120726_094707_4.LAS. When prompted, enter the following limits as the area to extract data from: 240166 240165 483292 483291

For example:

```
emca@pmpc1323-bash: ~/scratch/workshop/workshop_docs/2014/fwf_example<946>fwf_extract.py LDR120726_094707_4.LAS
Output directory was not specified, will use current directory
Enter limits of the area to extract:
Enter Max North (Range 239950.041000-240469.431000) 240166
Enter Min North (Range 239950.041000-240166.000000) 240165
Enter Max East (Range 482995.250000-483507.803000) 483292
Enter Min East (Range 482995.250000-483292.000000) 483291
Starting to process 779869 points
Will output ASCII files to /data/weddell1/scratch/emca/workshop/workshop_docs/2014/fwf_example/
Number of extracted waves: 3
```

You can also specify an area of interest on the command line using the '--area' argument, and an output directory using the '-o' argument.

```
fwf_extract.py <input_las_file.LAS> --area <North><South><East><West>
-o <output_directory >
```

For example:

```
emca@pmpc1323-bash: ~/scratch/workshop/workshop_docs/2014/fwf_example<947>fwf_extract.py LDR120726_094707_4.LAS
-o waveforms/ --area 240166 240165 483292 483291
Starting to process 779869 points
Will output ASCII files to /data/weddell1/scratch/emca/workshop/workshop_docs/2014/fwf_example/waveforms/
Number of extracted waves: 3
```

Note the output folder needs to exist first.

You can examine the extracted waveform files, which will be of the format waveform_<GPS_seconds>_<GPS_microseconds>_<return_type>.txt, e.g waveform_380962.592896_1.txt. In this case the file contains information relating to a point at GPS time 380962.592896, return number 1.

Note that there is also a file called waveform_380962.592896_2.txt, which contains information relating to return 2 of the same pulse. The sampled waveform data will be the same for both of these points.

The following describes the values represented in the ascii file:

Point:	XYZ position of associated discrete point
Return Number:	Return number of associated discrete point
Number of returns for this pulse:	Total Number of returns for this pulse
Time:	GPS time (Seconds of week) of associated discrete point
Scan Angle:	Scan angle of associated discrete point
Classification:	Classification of associated discrete point
Temporal Sample Spacing:	Sampling rate in nano seconds
AGC gain:	Automatic Gain Control
Digitiser Gain/Offset:	Digitiser gain and offset. Used to convert the raw digitiser value to an absolute digitiser voltage
No. of Samples:	Number of waveform samples
Sample Length:	Sample separation in metres
Return Point Location:	Offset in nano seconds from first digitised point to the associated discrete point
Point in Waveform:	Offset in metres between first digitised point and the associated discrete point
X/Y/Z Offset:	XYZ offset in metres between samples
Intensity X Y Z:	List of the sampled waveform data

Create a folder called 'waveforms.2' and run the extraction tool again on a second area 240283 240282 483269 483268, using the '-o' and '-- area' parameters.

6 Fityk

Fityk will be used to visualise the returned waveform and to extract additional information from the waveform data. This is done by trying to identify peaks in the waveform.

Open fityk under Linux by typing:

```
fityk
```

Under Windows, navigate to the fityk folder and double click on the fityk application.

Go to Data -> Load File. Navigate to the directory containing the extracted waveforms and select file waveform_380962_592896_1.txt.

Select column 0 for x and column 1 for y. Click 'Replace @ 0' and then click 'Close'

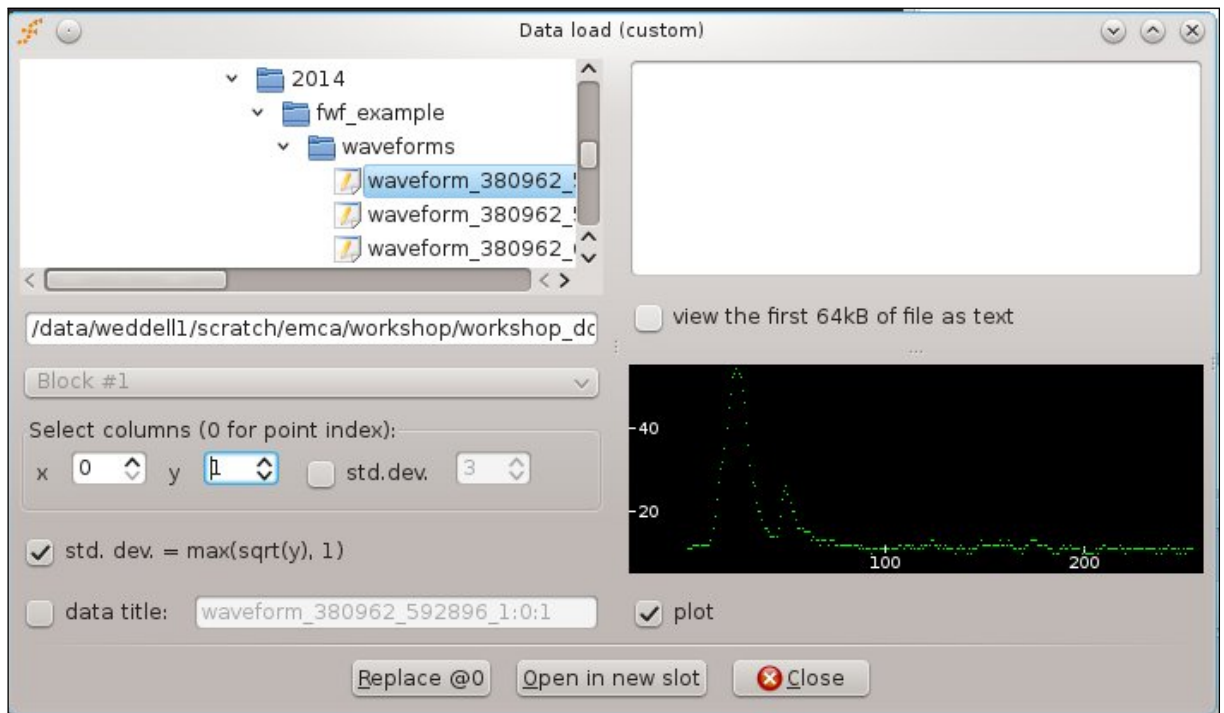


Figure 5: Fityk load screen

Tick the 'line' option on the right hand side. Note here that there are two distinct peaks visible. These correspond to discrete returns 1 and 2.

The next step is to manually fit a baseline and subtract the baseline to remove background noise.

To do this select 'Baseline Mode' from the tool bar (third icon from the left). The baseline determines where you think the noise threshold should be. Click ONCE on the graph to insert a flat baseline. Where you click determines the position of the baseline.

If you want to remove the baseline, go to GUI -> Baseline Handling -> Clear Baseline.

Subtract the baseline by selecting 'Strip Baseline' (icon to the left of Gaussian/Spline - see Figure 6)

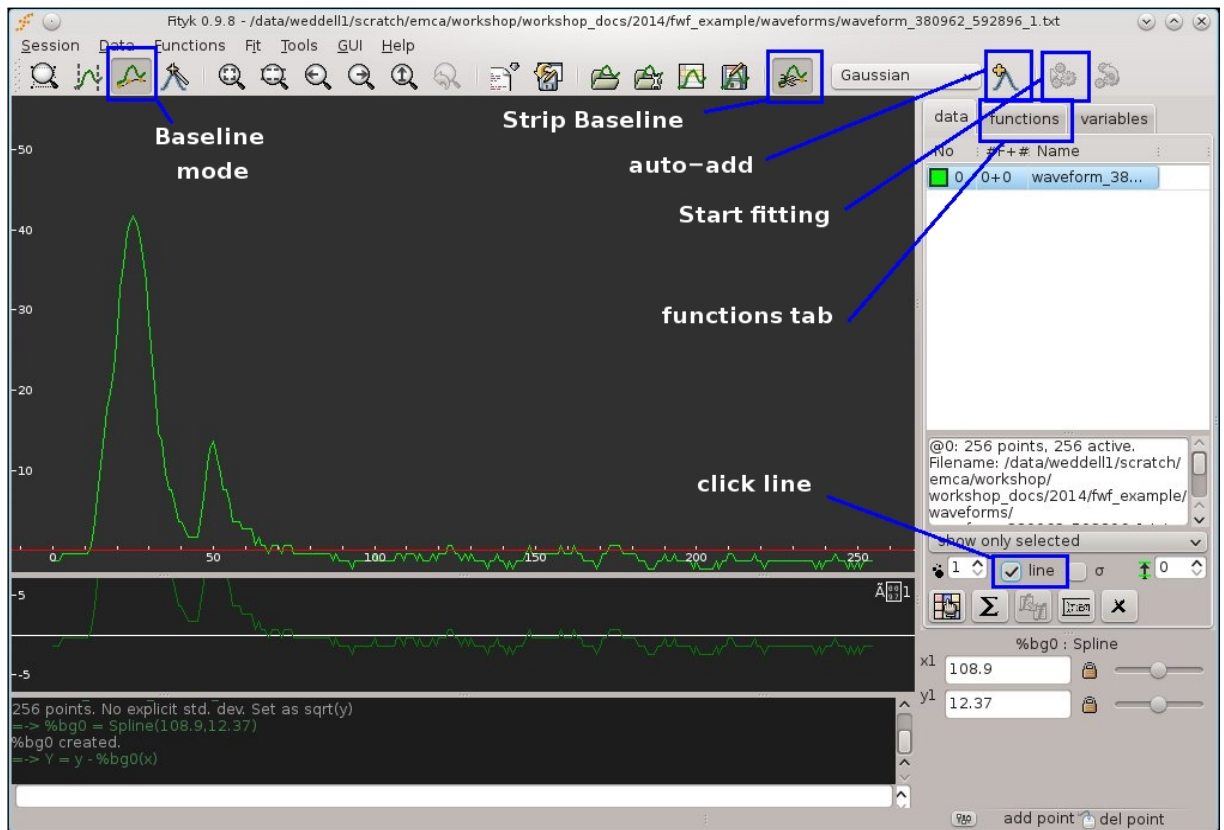


Figure 6: Fityk icon identification

Click on 'auto-add' to add peaks. Then select 'Start fitting'. Each of the fitted functions generates a number of values e.g. peak center, peak area, height, FWHM, intensity width. You can see these values by selecting the 'functions' tab.

The value given as the center of the peak should correspond approximately to the Return Point Location value given in the extracted waveform ascii file.

To clear the window click on Session -> Reset.

Now load file waveform_380961_552645_1.txt from waveforms.2 folder in a similar manner. In this case we are looking at a single return (Number of returns for this pulse is 1).

Note that a second peak can be distinguished which was not detected by the discrete system. As before, we can fit peaks to this waveform and use the

information extracted to create an additional discrete point corresponding to the second visible peak.

In order to work out the position of the new point, we use the center of the peak and the X,Y,Z offset values from the ascii file.

$$\begin{aligned}x_i &= x_0 + (x_t) * (r) \\y_i &= y_0 + (y_t) * (r) \\z_i &= z_0 + (z_t) * (r)\end{aligned}$$

where x_0 is the anchor point (first X sampled point) , x_t is the X offset and r is the return point location (center of peak).

Using the example shown here, the values we get are:

$$\begin{aligned}x &= 483268.878 \\y &= 240282.105 \\z &= 85.226\end{aligned}$$

Note that you may get slightly different values depending on where you place your baseline and on how many times you fit your data.

7 Alternative Tools

The following is a list of some additional tools that support full waveform LiDAR data.

SPDLib: <https://bitbucket.org/petebunting/spdlib>

OPALS: <http://geo.tuwien.ac.at/opals/html/index.html>

TerraScan: <http://www.terrasolid.com/products/terrascanpage.html>

8 Appendix - File list

http://arsf-dan.nerc.ac.uk/files/full_waveform_tutorial_data.LAS

<http://arsf-dan.nerc.ac.uk/trac/wiki/Downloads/Tutorials>

Filename	Description
Full_waveform_lidar_practical.pdf	PDF copy of instructions
LDR120726_094707_4.LAS	LAS1.3 file