User Guide of the 2nd Version of DASOS

Milto Miltiadou*[1,2,3], Neil DF Campbell[1], Matthew Brown[1], Susana Gonzalez Aracil[3], Mark Warren[2], Daniel Clewley[2], Darren Cosker[1] and Michael Grant[2]

[1]The Centre for Digital Entertainment, University of Bath, UK

[2]NERC Airborne Research Facility, Plymouth Marine Laboratory, UK

[3]Interpine Group Ltd, Rotorua, NZ

20/01/2017

*Corresponding author and main programmer.

Acknowledgments

Above all, I would like to express my great gratitude to my industrial supervisors Dr. Michael Grant who had supported me continuously during my research and gave me the freedom to create a project of my own interest.

Then, I would like to thanks Dr. Matthew Brown, who helped me during my first years of my studies by giving me valuable and informative feedback. He was always there to keep me working on the right track.

Equally important is my current supervisor Dr. Neil DF Campbell and he is not to be missed from the acknowledgements.

Furthermore, special thanks are given to Dr. Mark Warren, Dr. Daniel Crewely, Dr. Darren Cosker, MSc Susana Gonzalez Aracil and Dr. Ross Hill who occasionally advised me during my studies.

It further worth giving credits to my data providers, the Natural Environment Research Council's Airborne Research and Survey Facility (NERC ARSF) and Interpine Group Ltd.

Last but not least, I am extremely grateful to my funding organisations, the Centre for Digital Entertainement and Plymouth Marine Laboratory, who supported financially and consequently made this research possible.

Disclaimer

DASOS was developed to support an EngD thesis and as a research software could be file format and CPU dependant.

Identified bugs and a list of potential improvements can be found here: `<https://docs.google.com/spreadsheets/d/10yE5p463cLA_GtKkyiaWEzScW7N9cVxbPs5y0muXuZY/edit?usp=sharing>`

License

DASOS is released under the GNU General Public Licence, Version 3. The full description of the usage licence is available here:<https://github.com/Art-n-MathS/DASOS/blob/master/License.txt>

The following paper is the paper that introduced DASOS and it must be cited in any publications, software or other media using DASOS:

Open source software DASOS: efficient accumulation, analysis, and visualisation of full-waveform lidar. Miltiadou, M., Michael, G. G., Campbell, N. D., Warren, M., Clewley, D., & Hadjimitsis, D. G. (2019, June). *In Seventh International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2019)* (Vol. 11174, p. 111741M). *International Society for Optics and Photonics.* (Available at: <https://www.researchgate.net/publication/334069759_Open_source_software_DASOS_efficient_accumulation_analysis_and_visualisation_of_full-waveform_lidar>)

The 1^{st} sample dataset provided for testing was collected by the NERC Airborne Research and Survey Facility (ARSF). Copyright is held by the UK Natural Environment Research Council (NERC). The data are free for non-commercial use, but NERC-ARSF must be acknowledged in any publications, software or other media that make use of these data.

The 2^{nd} sample dataset provided by Interpine Group Ltd. Copyright is held by Interpine Group Ltd and the data are free for non-commercial use, but Interpine Group Ltd must be acknowledged for any publications, software or other media that make use of these data.

Contents

1 Introduction

FW LiDAR systems have been available for a number of years but there still very few uses of FW LiDAR data. NERC-ARF has been acquiring airborne data for the UK and overseas since 2010 and it has more than 100 clients of new and archived data. Many clients request FW LiDAR data to be acquired, but despite the significant number of requests,the majority of research still only uses discrete LIDAR. Some of the factors regarding this slow intakes are:

- Typically FW datasets are 5 – 10 times larger than discrete data, with data sizes in the range of 50GB – 2.5TB GB for a single area of interest. NERC-ARF's datasets are up to 100GB each because most clients are research institutes but for commercial purposes each FW dataset is a couple of TB.
- Existing workflows are only able to work with the discrete data since the increased amount of information recorded within the FW LiDAR makes handling the quantity of data very challenging.

The open source software DASOS was developed to encourage foresters to use the FW LiDAR data. DASOS was named after the Greek word "*δάσος*" (=forest) and it was firstly presented at the 36th International Symposium of Remote Sensing of the Enviroment, 2015. The main way of interpreting FW LiDAR data in DASOS is fundamentally different from the state-of-art available software packages. In a few words, the FW LiDAR data are voxelised by inserting the wave samples into a 3D discrete density volume. It accumulates intensity values from multiple shots and stores them into a 3D regular grid, resolving this way the problem with the sinusoidal footprints pattern of the Leica system. For more information please refer to the related paper at `<https://www.researchgate.net/publication/277347868_Alignment_of_hyperspectral_imagery_and_full-waveform_LIDAR_data_for_visualisation_and_classification_purposes>`

This user guide aims to give an in depth understanding of DASOS's functionalities. In a few words there are three functionalities of DASOS:

1. the construction of 3D polygon meshes;
2. the generation of 2D metrics aligned with hyperspectral Images and
3. the characterisation of objects using 3D priors/signatures.

2 Installation Guide

2.1 Windows

The windows executable are available at: `<https://github.com/Art-n-MathS/DASOS/tree/master/DASOS_win>`

DASOS is a command line software and a terminal is required. For Windows XP,Vista and 7, Command Prompt is the default terminal and it can be found from the search tab on the start menu. If you are using Windows 8, then right click on the start icon and choose 'Command Prompt'. Find the directory where DASOS is saved (the command 'dir' shows the files inside the current directory and the command 'cd' to open folder). Once you are in the correct directory, execute the following command to test that the program works fine.

$: DASOS --help [1]

Information of all the available commands should be printed. If an error is occur, then either a .dll file is missing or DASOS is not supported at your computer.

2.2 Linux

The source code is available at: `<https://github.com/Art-n-MathS/DASOS>`.

For compiling DASOS on linux, there are three dependancies:

1. qmake-qt4 (or later release) / qtcreator
2. gmtl library - please update the .pro file to point to the correct directory
3. -std=c++11

Once those are installed compile DASOS as shown below:

```
$: qmake-qt4
$: make
```

[1] the '$:' is not included in the command. It just illustrate the start of a command in the terminal

To test it, write the following command:

$: DASOS --help

Information of all the available commands should be printed.

3 Instructions

3.1 Overview

DASOS is a command line software and can either be used in Command Prompt on windows or a linux shell.

At first change directory (cd) to go to the directory DASOS is saved or compiled in. Then for windows run:

```
$: DASOS <arg1> <arg2> ... <argN>
```

or `$: DASOS.exe <arg1> <arg2> ... <argN>`

On Linux run:

```
$: ./DASOS <ar1> <ar2> ... <argN>
```

For consistency this guide uses the 1^{st} windows example since all of the inputs, parameters and output arguments are the same for both windows and linux.

The tags of DASOS are divided into three groups: Inputs, Parameters and Outputs.

1. Inputs (Section 3.2)
2. Parameters (Section 3.3)
3. Outputs (Section 3.4)

Even though many tags are optional or contain default values, it's essential to follow the order <inputs> <parameters> <outputs> because if the outputs are defined first unexpected results may occur, due to adding outputs to the stack before parameters are initialised. The aforementioned Sections give an explanation of all the possible tags of DASOS.

Before proceeding to the explanation, it worth highlighting and numbering the three main outputs of DASOS. The corresponding sub-sections do not only explain the output but also the parameters that are only specific to the corresponding output. The user guide refers to those outputs using their numbers:

1. The generation of 3D polygonal meshes (Section 3.4.1)
2. The 2D metrics aligned with Hyperspectral Imagery (Section 3.4.2
3. The 3D priors for local inspection of data (Sub-section 3.4.3)

At the end of this guide, there is a list of limitations and the dependencies.

3.2 Inputs

The inputs are divided into FW LiDAR, Hyperspectral and fieldplots. The FW LiDAR are compulsory for all the functionalities of DASOS. The Hyperpsectral Inputs are optional for the 1st and 2nd output of DASOS (3D polygonal meshes and 2D metrics), while the fieldplots option is compulsory for the 3rd option, the 3D priors.

Table 1 explains the tags for loading the FW LiDAR files files. Please note that it is compulsory to load one of those options. If more than one FW LiDAR files are loaded then it is essential to keep consistency between projects; load only one type of full-waveform LiDAR data simultaneously. Table 2 outlines how the hyperspectral imagery is loaded, how to subtract a pre-calculated DTM and also how the file with fieldplots is loaded if the 3rd output option is chosen (3D priors).

Tags	Description
-las <file1> <file2> ... <fileN>	The name/directory of a number of LAS files (i.e. `"C:\Dir Las\LAS1.las"`). It is further suggested to manually define the boundaries of the area of interest when more than one files are loaded (use command -userLimits explained at Table 3). Otherwise the boundaries of the first LAS file loaded is used. Furthermore DASOS only supports LAS1.3 with waveform packet format 4.
-pw <file1> <file2> ... <fileN>	loads a number of pulsewave files (*.pls). Same rules apply as the -las tag
-volume <file>	loads an exported volume, generated using DASOS, instead of reading a LAS or pulsewave file.
-vols <dir>	loads all the exported volume that are inside the given directory "dir". This option must and only be used for generating 3D priors (Section 3.4.3).

Table 1: DASOS fundamental file inputs

Tags	Description
-igm <igmFileName>	The name/directory of the .igm file that defines the geolocaiton of the hyperspectral pixels.
-bil <bilFileName>	The name/directory of the .bil file that contains the hyperspectral cube.
-fodis <fodisFile>	The name/directory of the fodis .bil file for hyperspectral imagery
-dtm <dtmFileName>	loads a pre-calculated DTM and subtracts it from the position of each waveform sample before importing it to the volume. Please note that the DTM file format must be .bil and saved into float pointing numbers. Potential further file format limitations may exist. This is optional.
-csv <fieldplots.csv>	The input csv file that lists all the trees from a number of field-plot. This is a compulsory input for generating 3D priors;

Table 2: Optional or output request dependant.

3.3 General Parameters

All the general parameters has been pre-defined and they are therefore optional. Nevertheless, parameters are advised to be adjusted for each project. Table 3 contains information about all the parameters and how to modify the the voxelised FW LiDAR data during construction. Figures 1 and 2 show how the results are affected when these parameters are modified.

Tags	**Description**
-userLimits <maxNorthY> <minNorthY> <maxEastX> <minEastX>	User define boundaries of the area of interest. If not defined then the boundaries of the first file loaded are used (as defined in the header).
-vl <voxel-Length>	The voxel length controls the resolution of the output; the bigger the voxel length is the lower the resolution and the number of cubes are. Default value is 2.5m
-nl <noiseLevel>	The noise level is the threshold of the low level filtering applied during voxelisation. Default value is 25. Please note that the intensity of each wave sample is not transformed to volts. Additionally, it is recommended to use the -exportPulses tag to export the amplitude of a few pulses and use those as sample data to define an appropriate noise threshold.
-iso <isolevel>	The iso-level is the intensity boundary that defines whether a voxel is inside or outside the voxelised object. The default value is zero. Please note that noise level and isosurface level are closely related but only the isolevel can be modified from an exported volume.
-calBoundaries <yes or no>	It calculates the boundaries of the voxelised space according to the maximum and minimum (x, y, z) values of the points stored within the file. This option only works for discrete LAS1.2 data with Point Record Format 2.

Table 3: Description of the all the available tags that customises voxelisation of the FW LiDAR data.

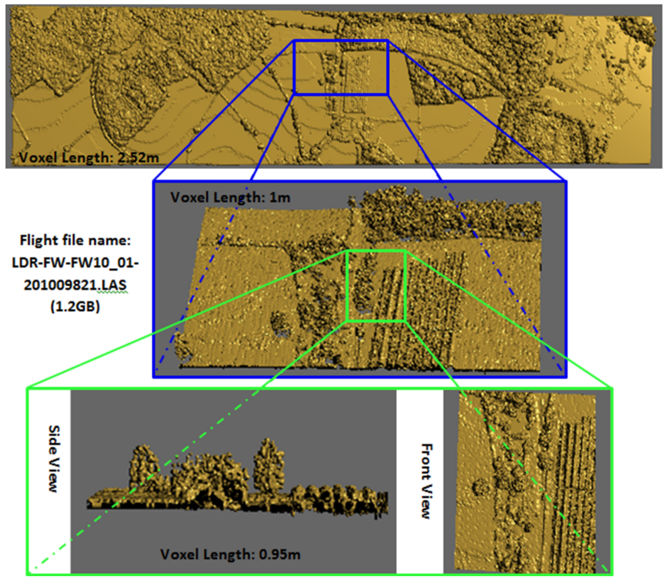


Figure 1: Selecting Region of Interest

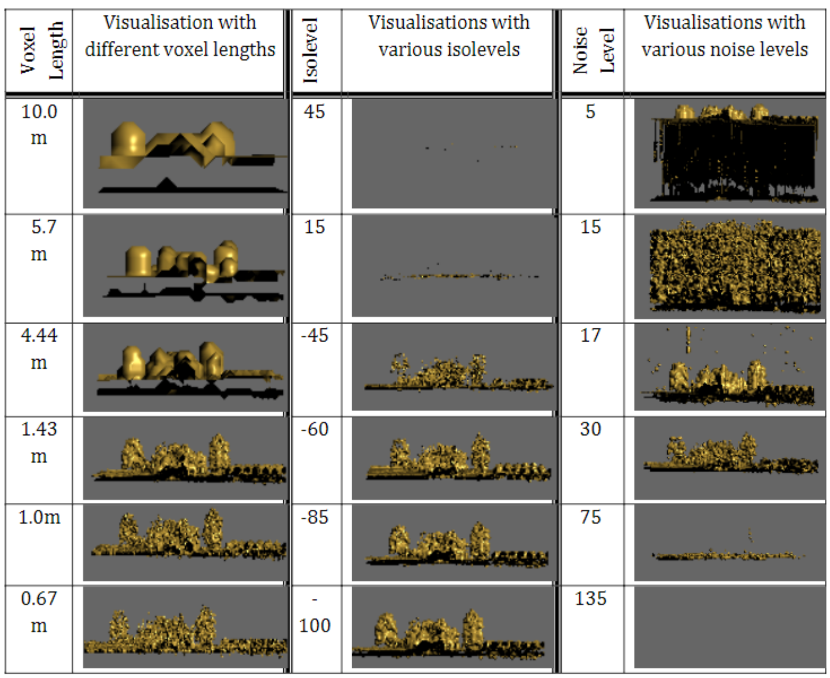


Figure 2: Modifying the user defined parameters; voxel length, isolevel and noise level.

3.4 Outputs

DASOS has three main outputs and three supplementary. At least one of them must be requested for the program to run.

The main outputs are the following:

1. **Polygonal Meshes**: exported into an .obj which is a standard graphics format that stores the vertices, edges and faces of the polygon. An image is also exported if hyperspectral data are loaded. (Section 3.4.1)

2. **2D Metrics**: information about the scanned area in .asc format. If hyperspectral Images are loaded then aligned metrics from both datasets are available. (Section 3.4.2)

3. **3D Priors**: exported into .csv files. Each row of the spreadsheet contains information about a local 3D cylinderal or cubic area. (Section 3.4.3)

The three suplementary outputs are explained in Table 4 while the main outputs are explained in Sections 3.4.1, 3.4.2 and 3.4.3 respectively.

Tags	**Description**
--help	It prints a list with all the available commands along with their description.
-exportPulses <noOfPulses> <fileName.csv>	Method that exports a number of pulses into a .csv file. The input <noOfPulses> the number of sample pulses to be exported into the <fileName.csv> file. It is used for deciding the noise level threshold for each project.
-exportVolume c <volumeFileName>	Exports the volume into an ASCII file to speed up future interpolation of the data. 'c' refers to compressed and it's an implicit functionality. If 'c' is not included then a non compressed file is exported, which sometimes is too big to be read back into DASOS. Therefore 'c' should always be included.

Table 4: The supplementary ouput options of DASOS.

3.4.1 Polygonal Meshes

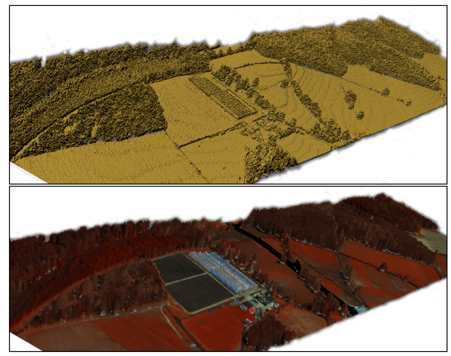
1st Main Output: 3D Polygon Mesh Generation		
Tags	**Description**	**Output Example**
-obj <objFileName>	The input <objFileName> is the name of the .obj file where the polygon representation of the LiDAR file will be exported to. A texture is also exported when hyperspectral images are loaded.	

Table 5: Description of generating polygonal meshes and example outputs

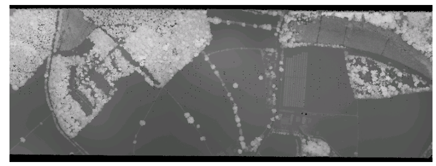
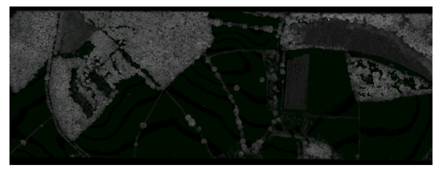
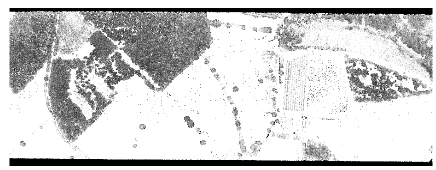
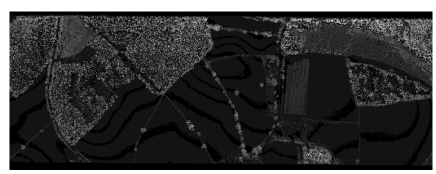
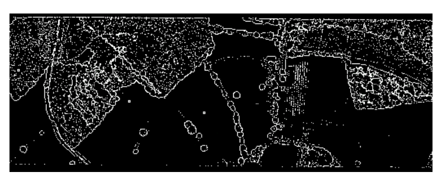
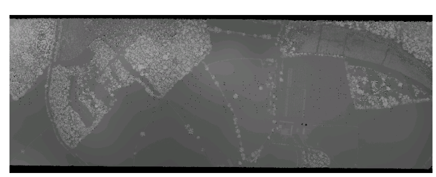
3.4.2 2D Metrics Aligned with Hyperspectral Imagery

<table>
<tr><th colspan="2">2nd Main Output: Generation of 2D metrics
aligned with hyperspectral imagery</th></tr>
<tr><th>Tags</th><th>Description</th></tr>
<tr><td>-map <type> <outputName></td><td>The available types are the following. Full description of each option is given in Table 7 along with output examples.

• HEIGHT
• THICKNESS
• DENSITY
• FIRST_PATCH
• LAST_PATCH
• AVERAGE_HEIGHT_DIFFERENCE
• LOWEST_RETURN
• INTENSITY_MAX
• INTESNSITY_AVG
• HYPERSPECTRAL_MEAN
• NDVI
• ALL_FW

All the maps are exported into .asc format and can be loaded into QGIS and other software packages. The ALL_FW option generates one metric for each available full-waveform LiDAR related metric and their names are: outputName+metricsType+.asc
Table 7 explains what each metric is and gives output examples.</td></tr>
<tr><td>-map HYPERSPECTRAL <band> <outputName></td><td>The hyperspectral map needs an extra parameter defining which band will be output.</td></tr>
</table>

Table 6: DASOS ouput options

Metric Description	Example
HEIGHT: The distance between the top non-empty voxel and the lower boundaries of the volume.	
THICKNESS: The distance between the first and last non empty voxels in every column of the 3D volume.	
DENSITY: Number of non-empty voxel over all voxels within the range from the first to last non-empty voxels.	
FIRST_PATCH: The number of non-empty adjacent voxels, starting from the first/top non-empty voxel in that column.	
LAST_PATCH: The number of non-empty adjacent voxels, starting from the last/lower non-empty voxel in that column.	
AVERAGE_HEIGHT_DIFFERENCE: An edge detection algorithm.	
LOWEST_RETURN The height of the lowest non empty voxel (the actual heights are very low and close to each but the example image has been scaled and the different seems bigger)	

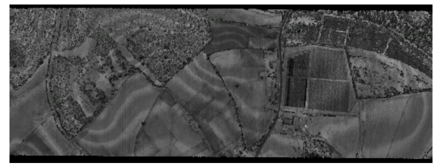
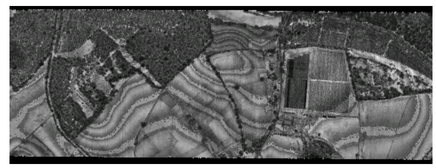
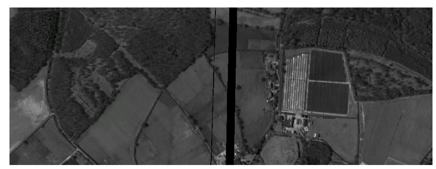
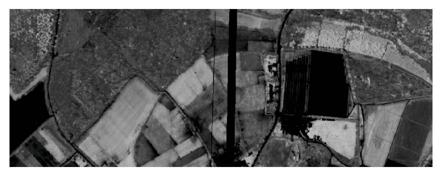
INTENSITY_MAX The maximum intensity of each column	
INTENSITY_AVG The average intensity per column	
HYPERSPECTRAL_MEAN The mean of the hyperspectral spectrum.	
NDVI The Normalised Difference Vegetation Index indicates whether green vegetation exists or not and it is derived from the electromagnetic spectrum as follow: $$NDVI = \frac{NIR - VIS}{NIR + VIS} \quad (1)$$ where the NIR is the near-infrared region of the spectrum (700-2500nm) and VIS is the Visible/Visual spectrum (430-770) [?].	
HYPERSPECTRAL A single user defined hyperspectral band.	

Table 7: Description of generating polygonal meshes and example outputs

3.4.3 3D Priors

This is useful for characterising object inside the 3D space (e.g. trees). For each column of the voxelised FW LiDAR, information around its local area are exported.

Similar to the previous functionalities of DASOS, the program requires <inputs> <parameters> and <outputs>. Those requirements are described in Tables 8, 9 and 10 respectively. Please note that these inputs are also described with the rest of the inputs in Section 3.2.

3rd Main Output: 3D Priors - Inputs	
Tags	**Description**
-vols <volDir>	the directory of the volume of interest generated beforehand.
-icsv <fieldplots.csv>	the input csv file that contains all information about the fieldplots.

Table 8: Explanation of how to define the two compulsory inputs to get the 3rd main output of DASOS

Figure 3 shows an example of a file with fielplots. A file may contain multiple fieldplots, but it has to have at least 6 columns: the 3 columns define the fieldplot (northing, easting and radius) and the rest give information about the trees (northing, easting and class). The order of the columns has no significance. Figure 3 shows an example. The labels of the those columns could vary and can be defined as explained in Table 9.

IsDead	Northing	Easting	X	Y	RADIUS
Live	60	70	55	75	40
Live	60	70	75	85	40
Dead	60	70	65	55	40
Live	60	60	20	60	40

Figure 3: Example of fieldplot input

Additionally, the size and shape of the prior is user defined and Table 9 lists all the related, modifiable parameters.

3rd Main Output: 3D Priors - Parameters	
Tags	**Description**
-column <label>	the label of the column that defines the class of each entry (e.g. <label> = isDead)
-class <className or ALL>	the name of the class (e.g. dead or alive) of interest or ALL. If a class is chosen, then only the columns that contain a tree of that class are taken into consideration; a prior is exported for each tree that belongs to this class only. The ALL option is the area of interest and generates a template for each column that lies inside the voxelised space.
-ttype square <x> <y> <z>	generates a squared prior of size x, y, z voxels. The systems finds the first non empty voxel starting from the top of the column. By default it moves one voxel upwards and sets that to be the top of the prior. It is highly recommended to use odd numbers, otherwise the centre of the prior will be wrongly set and unpredicted output values may occur.
-ttype cylinder <h> <r>	generates a cylindrical template with height h and diameter $(2 \times r + 1)$ voxels and height h. The systems finds the first non empty voxel starting from the top of the column. By default it moves one voxel upwards and sets that to be the top of the prior.
-mheight <n>	moves the template into the y-axis n voxels upwards instead of one which is the default. The value n must be a positive number.
-eparameters <raw or processed>	the 'raw' option saves all the intensity values of the template and the 'processed' option saves parameters derived from the raw intensities. Table 11 explains how each processed parameters is derived.

Table 9: Explanation on how to Modify the Parameters of the 3rd Main Output of DASOS

3rd Main Output: 3D Priors - Outputs	
Tags	**Description**
-ocsv <nameStart>	For each .vol file found in the given directory (using -vols), a csv file is exported. The name of each file exported is: <nameStart> + <volFileName> + ".csv" and it contains the priors generated from the corresponding volume

Table 10: Explanation of the tag that exports 3D priors

Figure 4 shows examples of two priors exported: one with processed parameters and one with raw intensities. In each .csv file exported, each line is a prior. The first column is the ID of the prior as it defined during run time. The second and third columns define the centroids of each prior. The other columns contain either processed or raw parameters. If they are processed, then information like mean height and standard deviation of heights are listed. Table 11 is a full list of all the proccessed parameters. If the parameters are raw, then the corresponding voxel intensity values are exported. The label of each voxel is "v_x_y_z", where "v_0_0_0" is the lower voxel of the prior and it has the minimum easting and norming of the prior as well.

Index	centroid_x	centroid_y	Height_Middle_Column	Height_Mean	Height_Median	Height_Std	Sum_Int_Diff_X	...
0	251836.109	6048994.5	36	35.5	36	0.943	95.125	...
1	251843.906	6048980.5	19.8	20.1	20.4	0.671	0	...
2	251846.312	6048979	16.8	16	15.6	1.02	169.167	...
3	251849.312	6049022.5	36	35.7	36.6	0.964	169.278	...
4	251851.703	6048988	17.4	16.2	16.2	0.346	408.065	...
5	251852.906	6048975	27	26.4	26.4	0.917	68.537	...
6	251857.109	6048974	17.4	17.4	18	0.849	162.25	...
7	251858.312	6049010.5	40.8	40	39.6	1.02	251.36	...
8	251860.703	6048984	17.4	16.6	16.2	0.663	67.883	...
9	251861.312	6049000	19.8	20.1	20.4	0.671	0	...

Index	centroid_x	centroid_y	V0_0_0	V0_0_1	V0_0_2	V0_0_3	V0_0_4	V0_1_0	V0_1_1	V0_1_2	V0_1_3	...
0	251836.109	6048994.5	7	14	10	26	0	0	9	10.25	11.875	...
1	251843.906	6048980.5	0	0	0	0	0	0	0	0	0	...
2	251846.312	6048979	9	60.75	70.75	13	8	0	0	0	7.667	...
3	251849.312	6049022.5	48.556	93.222	20.5	0	7	0	0	0	0	...
4	251851.703	6048988	100.2	53.222	10.5	7.143	0	0	0	0	47.25	...
5	251852.906	6048975	0	0	0	0	0	26.875	0	10.444	13.182	...
6	251857.109	6048974	0	0	0	0	0	45.667	93	16.333	7.25	...
7	251858.312	6049010.5	0	0	0	0	0	0	0	8	6	...
8	251860.703	6048984	0	45.75	8	7.333	0	0	0	0	6.8	...
9	251861.312	6049000	0	0	0	0	0	0	0	0	0	...

Figure 4: Example of .csv files with priors exported.

Explanation of the 3D priors Output with the Processed Intensities	
Label	**Description**
Height_Middle_Column	The height of the middle column of the prior
Height_Mean	The Mean height of all the columns included in the template
Height_Median	The Median height of all the columns included in the template
Height_Std	The Standard Deviation of the heights of the columns included in the template
Sum_Int_Diff_X	The Mirror Summed Difference of the intensities using the middle column in the x-axis as the axis of symmetry
Sum_Int_Diff_Y	The Mirror Summed Difference of the intensities using the middle column in the y-axis as the axis of symmetry
Sum_Int_Diff_Z	The Mirror Summed Difference of the intensities using the middle column in the z-axis as the axis of symmetry
Max_Int	The maximum intensity found inside the prior
Min_Int	The minimum intensity found inside the prior
Ave_Int	The average intensity of the voxels that contain an intensity above the isolevel
Median_Int	The median intensity of the voxels
Per_Int_Above_Iso	Percentage of voxels that contain an intensity above the isolevel
Dis_Mean	Mean distance from the central voxel to every voxel that contain san intensity above the isolevel
Dis_Median	Median distance from the central voxel to every voxel that contains an intensity above the isolevel
Dis_Std	The Standard Deviation of the distances between the central voxel and every voxel that contains an intensity above the isolevel

Top_Patch_Len_Middle_Col	The length of the top patch of the middle column of the prior
Top_Patch_Len_Mean	The Mean length of all the top patches
Top_Patch_Len_Median	The Median length of all the top patches
Top_Patch_Len_Std	The Standard Deviation of all the top patches
Mirror_Diff_X_Mean	The Mean Mirror Difference of the voxel intensities with the middle column of the x-axis as the symmetric axis
Mirror_Diff_X_Median	The Median Mirror Difference of the voxel intensities with the middle column of the x-axis as the symmetric axis
Mirror_Diff_X_Std	The Standard Deviation Mirror Difference of the voxel intensities with the middle column of the x-axis as the symmetric axis
Mirror_Diff_Y_Mean	The Mean Mirror Difference of the voxel intensities with the middle column of the y-axis as the symmetric axis
Mirror_Diff_Y_Median	The Median Mirror Difference of the voxel intensities with the middle column of the y-axis as the symmetric axis
Mirror_Diff_Y_Std	The Standard Deviation Mirror Difference of the voxel intensities with the middle column of the y-axis as the symmetric axis
Mirror_Diff_Z_Mean	The Mean Mirror Difference of the voxel intensities with the middle column of the z-axis as the symmetric axis
Mirror_Diff_Z_Median	The Median Mirror Difference of the voxel intensities with the middle column of the z-axis as the symmetric axis
Mirror_Diff_Z_Std	The Standard Deviation of the Mirror Difference of the voxel intensities with the middle column of the z-axis as the symmetric axis

Table 11: The three functionalities of DASOS

4 Exercises

4.1 Sample Data

These exercises will give you an in depth understanding of DASOS, while working with real examples. At first, copy the folder "DASOS_userGuide" into your C:\ drive. This folder is available to dowload from <https://github.com/Art-n-MathS/DASOS/tree/master/DASOS_win>: To ease typing, all the example commands are given into the ExerciseCommands.bat file, which can be opened in a text editor.

There are three datasets provided for the following exercises and they are available at: <https://www.dropbox.com/sh/hzpl16gue5xvjmb/AADQsJOsqKkx0lCX4mJjvBPVa?dl=0>. Please copy the data inside the directory <
DASOS_userGuide
SampleDATA> check that the following files are included:

1. 1^{st} sample dataset inside < C:\DASOS_userGuide\SampleDATA\DATASET_1>:
   (a) LDR-FW-FW10_01-201009821.LAS
   (b) e098211b_FODIS.bil
   (c) e098211b_FODIS.bil.hdr
   (d) e098211b_masked.bil
   (e) e098211b_masked.bil.hdr
   (f) e098211b_osgn.igm
   (g) e098211b_osgn.igm.hdr
   (h) Readme.txt
2. 2^{nd} sample dataset inside < C:\DASOS_userGuide\SampleDATA\DATASET_2>
   (a) Australia_1.pls
   (b) Australia_1.wvs
   (c) Australia_1_dtm.bil
   (d) Australia_1_dtm.hdr

(e) `Australia_2.las`

(f) `Australia_2.wdp`

(g) `Australia_2_dtm.bil`

(h) `Australia_2_dtm.hdr`

(i) `Australia_3.las`

(j) `Australia_3.wdp`

3. 3^{rd} sample dataset inside $<$ `C:\DASOS\_userGuide\SampleDATA\DATASET_3`$>$

(a) `myTestVol_.vol`

(b) `myTestVol_flat.vol`

(c) `testFieldplot.csv`

Information about data usage and related license are given in Section .

Once all the files are copied across, open the command Prompt and type:

```
$: cd C:\DASOS_userGuide\DASOS
```

This will bring you to our working directory. In case you are using a different directory then go to your work directory inside the folder DASOS and the rest of the commands should work OK.

A full guide of all the available tags is given with the following command.

```
$: DASOS --help
```

The same information can be found inside the Readme.txt file and this User Guide (Section 3).

4.2 Exercises

4.2.1 Deciding Noise Threshold

The following examples export the amplitudes of 12 pulses into a .csv file to help us decide what noise threshold to use.

```
$:DASOS -las ..\SampleDATA\DATASET_1\LDR-FW-FW10_01-201009821.LAS -exportPulses 12
   ..\LAS21pulsesSamples.csv
```

```
$: DASOS -las ..\SampleDATA\DATASET_2\Australia_2.las -exportPulses 12
   ..\Australia_2_pulsesSamples.csv
```

4.2.2 Exporting metrics from DASOS

The following commands export a height map into .asc files. These files can be used in QGIS. This will give us the location of the flightlines and the relation between them.

```
$: DASOS -las ..\SampleDATA\DATASET_2\Australia_2.las -nl 6 -vl 2 -map height
   ..\Australia_2_vl2_height
```

```
$: DASOS -las ..\SampleDATA\DATASET_2\Australia_3.las -nl 6 -vl 2 -map height
   ..\Australia_3_vl2_height
```

Generating a single map at the beginning is useful for deciding which flightlines lie inside the area of interest.

4.2.3 Loading Multiple Flightlines

As mentioned before, for loading multiple flightlines it is suggested to manually define the boundaries of the area of interest. The following command loads two flightlines, generates a volume from the area of interest and exports it into the Australia2-3.vol file.

```
$: DASOS -las ..\SampleDATA\DATASET_2\Australia_3.las
   ..\SampleDATA\DATASET_2\Australia_2.las -nl 6 -vl 2 -iso 4 -userLimits 6199990 6199639
   762405 761951 -exportVolume c ..\Australia2-3.vol
```

4.2.4 Exporting Metrics

The following command loads the pre-computed volume and creates a height map and all the FW related metrics. Please note that height is also a FW related metric, therefore it will be created twice.

```
$: DASOS -volume ..\Australia2-3.vol -map height ..\Australia2-3 -map all_fw
   ..\Australia2-3
```

4.2.5 Subtracting Pre-computed Digital Terrain Model

The next command loads two LAS files, a pre-computed DTM file is subtracted from the wave samples' positions while the volume is created, the volume is exported into the `Australia2-3_dtm.vol` file and finally it exports a height metric.
Please note that when a DTM is introduced, a new volume must be created. Since the volumetric files are raster data and contain no information about pulses.

```
$: DASOS -las ..\SampleDATA\DATASET_2\Australia_3.las ..\SampleDATA\DATASET_2\Australia_2.las
   -dtm ..\SampleDATA\DATASET_2\Australia_2_dtm.bil -nl 6 -vl 2 -iso 4 -userLimits 6199990
   6199639 762405 761951 -exportVolume c ..\Australia2-3_dtm.vol -map height
   ..\Australia2-3_vl2_dtm_height
```

You may then use the same volume to export more metrics:

```
$: DASOS -volume ..\Australia2-3_dtm.vol -map AVERAGE_HEIGHT_DIFFERENCE
   ..\Australia2-3_dtm_AVG_height_diff
```

4.2.6 Pulsewave Data

As mentioned before, it is suggested to first export the amplitudes of a few pulses to decide on an appropriate noise threshold.

```
$: DASOS -pw ..\SampleDATA\DATASET_2\Australia_1.pls -exportPulses 15
   ..\PLS_amplitudeSamples.csv
```

And then you can generate the desired metrics:

```
$: DASOS -pw ..\SampleDATA\DATASET_2\Australia_1.pls -nl 5 -dtm
   ..\SampleDATA\DATASET_2\Australia_1_DTM_1m.bil -vl 3 -map thickness PLS_vl3_thickness
   -exportVolume ..\Australia_1_vl3_dtm.vol
```

4.2.7 Polygon Representation

DASOS create 3D polygon representation using the '-obj' tag. The 3D polygon representations are exported into .obj format, which can be visualised using animation software packages. For this workshop we are using Meshlab because it is a free tool and it can handle millions of triangles. Meshlab is available to download from here: `<http://meshlab.sourceforge.net/>` and it is also included into our working directory `"DASOS_userGuide"`.

An example of generating polygons is given below:

```
$: DASOS -las ..\SampleDATA\DATASET_1\LDR-FW-FW10_01-201009821.LAS -nl 20 -vl 1.7 -obj
   ..\LAS21.obj -exportVolume c ..\LAS21_vl1.7.vol
```

The generated volume is also saved because we need it for the following exercises.

4.2.8 Hyperspectral Imagery

One of the key functionalities of DASOS is the alignment with the hyperspectral imagery. DASOS can export 3D coloured polygon representations and aligned metrics between FW LiDAR and hyperspectral data.

For the 3D coloured polygon representations you must not use any directory for the exported .obj file Analysisname because the link between the texture and the .obj file will not work. Here is an example:

```
$: DASOS -volume ..\LAS21_vl1.7.vol -bil ..\SampleDATA\DATASET_1\e098211b_masked.bil -igm
   ..\SampleDATA\DATASET_1\e098211b_osgn.igm -fodis
   ..\SampleDATA\DATASET_1\e098211b_FODIS.bil -rgb 240 78 23 -obj
   LAS21_coloured.obj
```

The LAS21.obj file will be saved into the current directory, which in our case is `C:\DASOS_userGuide\DASOS`.

Please note that the following command should give the same results, but as mentioned before importing an exported volume is faster than generating from scratch.

```
$: DASOS -las ..\SampleDATA\DATASET_1\LDR-FW-FW10_01-201009821.LAS -nl 20 -vl 1.7 -bil
   ..\SampleDATA\DATASET_1\e098211b_masked.bil -igm
   ..\SampleDATA\DATASET_1\e098211b_osgn.igm -fodis
   ..\SampleDATA\DATASET_1\e098211b_FODIS.bil -rgb 240 78 23 -obj LAS21_coloured.obj
```

An examplre of generating aligned metrics is given below. The NDVI map is quite slow, so we may need to wait a bit for that.

```
$: DASOS -volume ..\LAS21_vl1.7.vol -bil ..\SampleDATA\DATASET_1\e098211b_masked.bil -igm
   ..\SampleDATA\DATASET_1\e098211b_osgn.igm -fodis
   ..\SampleDATA\DATASET_1\e098211b_FODIS.bil -map hyperspectral 140 ..\LAS21_band140 -map
   height ..\LAS21_height -map NDVI ..\LAS21_ndvi
```

4.2.9 All Commands Together

Of course, we are able to use multiple outputs into a single command, even though that's not suggested due to the big processing time. An example of merging previous commands into one is given below:

```
$: DASOS -las ..\SampleDATA\DATASET_1\LDR-FW-FW10_01-201009821.LAS -nl 20 -vl 1.7 -bil
   ..\SampleDATA\DATASET_1\e098211b_masked.bil -igm
   ..\SampleDATA\DATASET_1\e098211b_osgn.igm -fodis
   ..\SampleDATA\DATASET_1\e098211b_FODIS.bil -rgb 240 78 23 -obj LAS21_coloured.obj -map
   hyperspectral 140 ..\LAS21_band140 -map  height ..\LAS21_height -map NDVI ..\LAS21_ndvi
   -exportVolume ..\LAS21_vl1.7.vol
```

4.3 Exporting 3D priors from exported voxelised FW LiDAR

This examples takes as input two test .vol files and a fieldplot file. The file "myTestVol_flat.vol" contains a flat surface, while inside the "myTestVol_.vol" the middle column of the first dead tree that is defined inside the "testFieldplot.csv" is one voxel higher. The covered area of the two .vol files is identical and for that reason the fieldplot circle lies inside both files. The following commands produce the processed priors of dead trees and the raw priors of all the columns.

```
$: DASOS -vols ..\SampleDATA\DATASET\_3  -icsv
..\SampleDATA\DATASET_3\testFieldplot.csv -eparameters processed -column isDead
-class dead -ttype square 3 3 5 -ocsv templatesProcessedCuboid
```

```
$: DASOS -vols ..\SampleDATA\DATASET_3  -icsv
..\SampleDATA\DATASET_3\testFieldplot.csv -eparameters raw -column isDead -class
 ALL -ttype cylinder 5 3 -ocsv templatesALLRawCylinder
```

5 Limitations

Limitation and bugs have been reported throughout the report, but here is a short summary of them.

- Exporting polygon representation could end up generating a punch of cones instead of a nice smooth surface.
- Subtracting DTM depends on the input file format and by subtracting the height the input data may end up outside the boundary of the volume.
- DASOS may be Hardware/CPU dependant since development and testing was done on two computers only.
- The raw waveform amplitude is used as intensity and it hasn't been converted to an absolute digitizer voltage.
- Intensities also have not been calibrated.
- Sometimes memory allocation exceptions occur.

For full bug reports and under development improvements please check the following link:
<https://docs.google.com/spreadsheets/d/10yE5p463cLA_GtKkyiaWEzScW7N9cVxbPs5y0muXuZY/edit?usp=sharing>

6 Related Forums and Social Media

Online social media are used for sharing DASOS updates and discussing issues or potential improvements. Information about DASOS can be found in the following:

- Google Groups: DASOS - the native full-waveform (FW) LiDAR software
  `<https://groups.google.com/forum/#!forum/dasos---the-native-full-waveform-fw-lidar-software>`
  This group is used for bringing potential issues and possible improvements up in discussion.

- Blogger: ART & M@thS
  `<http://miltomiltiadou.blogspot.co.nz/2015/03/las13vis.html>`
  This blog is more general. The blog contains updates and explanation of DASOS but usually the code used in DASOS is broken down into small projects and explained how they can be used in other applications.

- Twitter: `@MiltoMiltiadou`
  Milto Miltiadou's twitter, where all the updates and news of DASOS are published.

7 List of Related Publications

- **Detecting dead standing Eucalypt trees from voxelised full-waveform LiDAR using 3D multi-scale windows for tackling height and size variations**
  Miltiadou, M., Agapiou, A., Gonzalez Aracil, S., & Hadjimitsis, D. (2020, January). *Forests, MDPI journal* - impact factor 2.11
  Available at: <`https://www.mdpi.com/1999-4907/11/2/161`>

- **Open source software DASOS: efficient accumulation, analysis, and visualisation of full-waveform lidar.**
  Miltiadou, M., Michael, G. G., Campbell, N. D., Warren, M., Clewley, D., & Hadjimitsis, D. G. (2019, June). *In Seventh International Conference on Remote Sensing and Geoinformation of the Environment (RSCy2019)* (Vol. 11174, p. 111741M). *International Society for Optics and Photonics.*
  Available at: <`https://www.researchgate.net/publication/334069759_Open_source_software_DASOS_efficient_accumulation_analysis_and_visualisation_of_full-waveform_lidar`>

- **Improving and Optimising Visualisations of full-waveform LiDAR data**
  Miltiadou M., Campbell N.D.F, Brown M., Cosker D., Micheal G. (2016). Improving and Optimising Visualisations of Full-waveform LiDAR data. *ACM Digital Library* .
  Available at: <`https://www.researchgate.net/publication/316621634_Improving_and_optimising_visualisations_of_full-waveform_LiDAR_data`>

- **Alignment of hyperspectral imagery and full-waveform LiDAR data for visualisation and classification purposes**
  Miltiadou, M., Warren, M. A., Grant, M., & Brown, M. (2015). *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences-ISPRS Archives*
  Available at: <`https://www.researchgate.net/publication/277347868_Alignment_of_hyperspectral_imagery_and_full-waveform_LIDAR_data_for_visualisation_and_classification_purposes`>

- **Reconstruction of a 3D Polygon Representation from full-waveform LiDAR data**
  Miltiadou, M., Grant, M., Brown, M., Warren, M., & Carolan, E. (2014). *Conference Proceedings of RSPSoc Conference*
  Available at: <`https://www.researchgate.net/publication/277622377_Reconstruction_of_a_3D_Polygon_Representation_from_full-waveform_LiDAR_data`>