

AMAP



# AMAPVox

## LIDAR data voxelization software

---

### User's Manual

Release date : July 6, 2016

Designer: Grégoire Vincent

Developer : Julien Heurtebize

Contributors : Jean Dautat, Claudia Lavalley



# Summary

System requirements.....	4
Starting up.....	4
I – Configuration files.....	5
I.1 - Save a configuration file.....	5
I.2 - Task list.....	5
I.3 - Editing a configuration file.....	6
II – Voxelization.....	7
II.1 - Parameters.....	8
II.1.1 - Bounding-box.....	8
II.1.2 - Transformation matrices.....	8
II.1.3 - Filtering.....	9
II.1.3.1 - Shot filtering.....	9
II.1.3.2 - Echoes filtering.....	9
II.1.4 - Echo weighting.....	11
II.1.5 - Limits.....	12
II.1.6 – Leaf Angle Distribution (LAD).....	12
II.2 - ALS.....	14
II.2.1 – Supported formats.....	14
II.2.2 - Process.....	14
II.3 - TLS.....	15
II.3.1 - Supported formats.....	15
II.3.2 - Process.....	15
II.4 - Dart export.....	17
II.4.1 – plots.xml.....	18
II.4.2 – maket.txt / triangles.txt.....	19
III – Virtual measures.....	20
III.1 – Transmittance light map.....	20
III.1.1 – Input parameters.....	21
III.2 – Canopy analyzer.....	21
III.2.1 – Input parameters.....	21
III.3 – Hemispherical photography.....	22
III.3.1 – Input parameters.....	22
IV – Visualization.....	22
IV.1 - Load a file.....	23
IV.2 - Visualize a raster.....	23
IV.3 – Toolbox.....	24
IV.3.1 - Voxel viewing parameters:.....	24
IV.3.2 – Scene.....	26
IV.3.3 - Tools.....	26
V - Annex.....	26
V.1 - File formats.....	26
V.1.1 - Point files (.las).....	26
V.1.2 - Compressed point files (.laz).....	27
V.1.3 - Trajectory file.....	27
V.1.4 - Riscan scan file (.rxp).....	27
V.1.5 - Riscan Project (.rsp).....	27
V.1.6 - DTM (Digital Terrain Model).....	27

V.1.7 - Voxel file.....	27
V.1.8 - Matrix file.....	28
V.1.9 - Configuration file.....	29
V.2 – Algorithms.....	31
V.2.1 - Merging trajectory and las (ALS).....	31
V.2.2 - Transmittance processing (ALS+TLS).....	31
V.2.3 - Computation of the optical path (ALS+TLS).....	35
V.2.4 – Resampling voxels (ALS).....	36
V.2.5 - Scan merging (TLS).....	36

## System requirements

Operating System	64bits Tested OS : Windows 7 – 8 – 10, Ubuntu
RAM	Minimum : 2 gb Recommanded : 8 gb
CPU	A multi-core CPU is advised, to take advantage of multi-threading for parallel task execution
Java	JRE 8 from Oracle is required
OpenGL	Version 3 minimum (for visualization)

## Starting up

-Launch script `click_me.bat` (Windows) or `click_me.sh` (Linux)

This script will allocate the best maximum memory to use by the JVM based on the currently available memory.

-Or from command line, set current directory where file « `AMAPVoxGUI-1.0-rxxxxxxx.jar` » is located and enter :

```
java -jar -Xmx8g AMAPVoxGUI-1.0-rxxxxxxx.jar
```

Replace `xxxxxxx` by the actual program revision.

Parameter `-Xmx` followed by `8g` value indicate set the maximum heap size value of the JVM to 8 giga-bytes. This amount can be edited and it is advised to do so.

### Java known issues

→ `XmX` value need to be increased if you get following exception : `java.lang.OutOfMemoryError: Java heap space`

If you don't specify the `Xmx` parameter then the JVM is going to use as much memory as is available. To avoid crashing or reduced performance you should set this parameter so that 2go are reserved for the system.

→ If multiple versions of Java are installed on the system, you might have program launching issues. To see which version is currently active you can open a terminal and execute the following command :

```
java -version
```

If the Java version appears to be anterior to version 8 even though you have installed Java 8 you can find help with online at <https://www.java.com/fr/download/help/path.xml>

You need to install JRE/JDK provided by Oracle. Get 64 bits version (x64).

Download link : <http://www.oracle.com/technetwork/java/javase/downloads/jre8-downloads-2133155.html>

## I – Configuration files

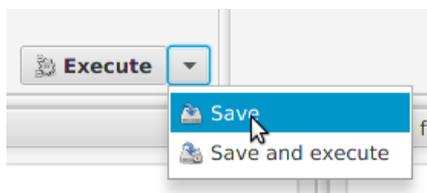
Before executing a task, it is advised to save your configuration.

When you execute the task without saving a temporary configuration file is automatically generated anyway, check the log to know the path.

### I.1 - Save a configuration file

The button « Execute/Save/Save and execute » is available from each task panel who required an execution from a configuration file.

Those buttons are the way to go to create a configuration file.



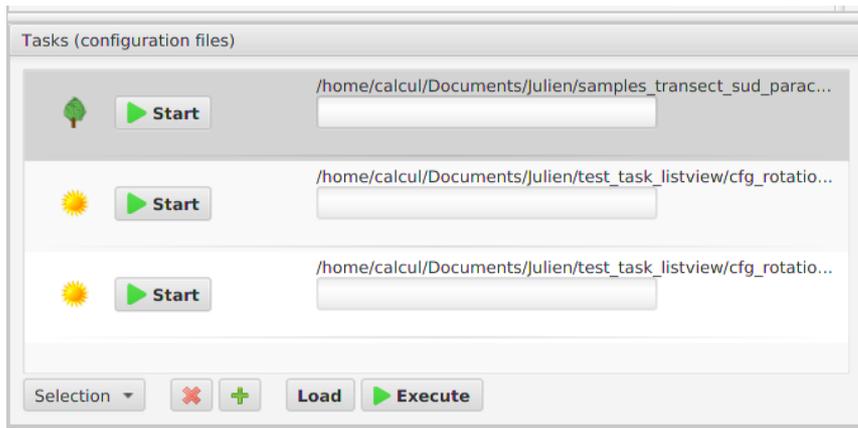
*Execute* : Execute the task with the current parameters, create a temporary file but this is transparent for the user.

*Save* : Write a configuration file and add it to the task panel (bottom left).

*Save and execute* : Write a configuration file, add it to the task panel and execute the task

### I.2 - Task list

« Tasks » panel in the bottom left corner of the window may contain a list of configuration files.



You can execute the tasks one by one by pressing the « Start » button.

You can also execute a selection of tasks :

- select the tasks from the list
- click on the « Execute » button

At this point, the program will ask you if you want to execute the tasks in a sequential or parallel way.

 All unsaved parameters will not be taken into account to the execution of the task. That's why it is necessary to save any modification before executing.

### 1.3 - Editing a configuration file

To edit a configuration file previously saved you need :

If the file is not in the « Tasks » panel you have to click on the « + » button and open the file.

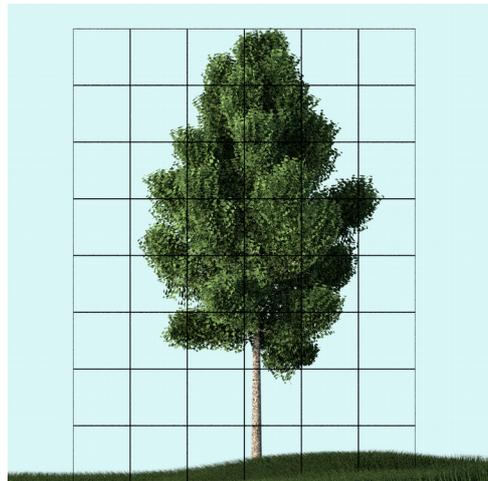
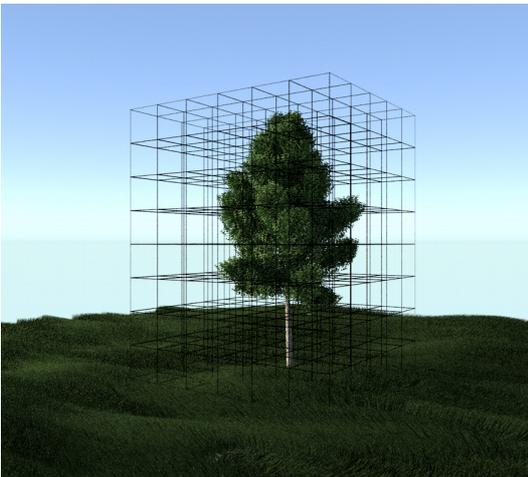
Next, select the file and click on the « Load » button, file parameters will fill in the relevant fields in the window.

You can now edit fields in the different panels and erase the old file or create a new one by saving the configuration file (see [Save a configuration file section](#)).

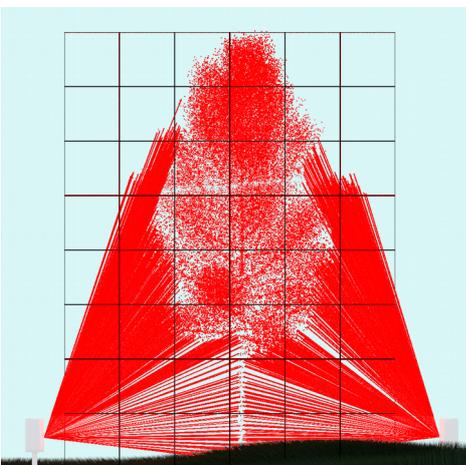
## II – Voxelization



*Illustration 1: Isolated tree*



*Illustration 2: Space clustering*



## II.1 - Parameters

Voxelization requires a certain number of parameter values to be set, some of which are mandatory and the others are optional.

### II.1.1 - Bounding-box

Voxel number by axes is automatically calculated from minimum and maximum points and resolution. Those values are expressed in unit meters (m).

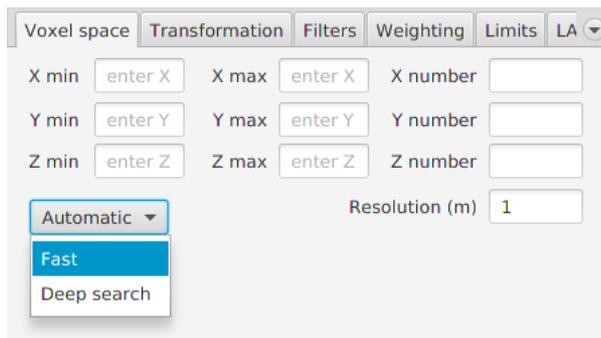


Illustration 4: Voxel space settings panel

Automatic button allows the automatic calculation of the extremums given a set of points (ALS).

Fast option choice read the extremums in header file (als, laz).

In case where an additional transformation is set (VOP matrix = Voxel Orientation and Position), the option behaviour will be the same as the « Deep search» option.

« Deep search » option choice read all points from the file, apply a VOP matrix if there is any and does filtering by classification as defined in the tab « Filter » → « Other ».

### II.1.2 - Transformation matrices

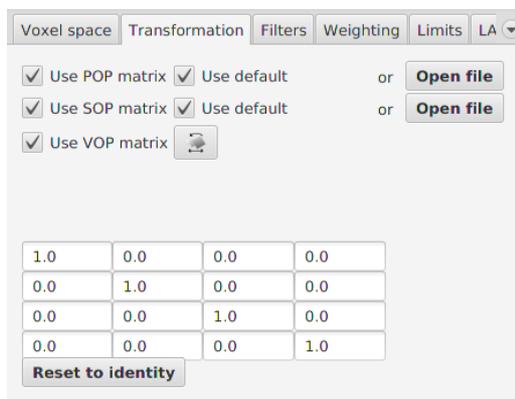


Illustration 5: Shot transformation panel

#### POP matrix

Project Orientation and Position, TLS only

Projection matrix of a Riscan Pro project, this is defined in the project file (\*.rsp).

That matrix is automatically filled when a Riscan Pro project file is open, being read in the file.

Using single scan (\*.rxp) voxelization, it is possible to defined POP matrix, either by opening a matrix file (see file formats in annexe), or by choosing a Riscan Pro project file.

### SOP matrix

System Orientation and Position, TLS only.

Each scan from Riscan Project has its own SOP matrix which is included in Riscan Pro project file.

If a single scan (\*.rxp) is selected, by clicking on the « Open file » button next to POP matrix you can choose the Riscan Pro project file and it will automatically configure the POP matrix and the SOP matrix of that scan.

### VOP matrix

Voxel Orientation and Position.

Optional transformation matrix.

## II.1.3 - Filtering

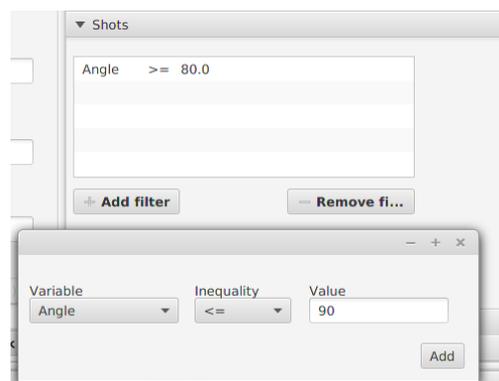
From the « Filters » tab you have the possibility to add a filter on the echos (points) and shots (source point and direction vector).

### II.1.3.1 - Shot filtering

The only available shot filter is the angle filter.

The angle is the zenithal angle viewed from the ground.

« Shots » panel.



*Illustration 6: Shot filtering section*

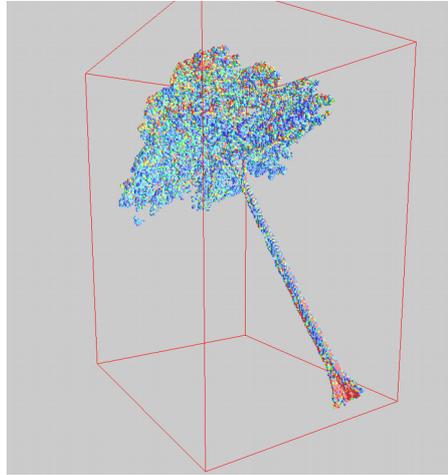
### II.1.3.2 - Echoes filtering

-From a Digital Terrain Model : this filtering removes all points above a certain height from ground (those points will therefore be ignored in the vegetation density analysis).

« DTM » panel.

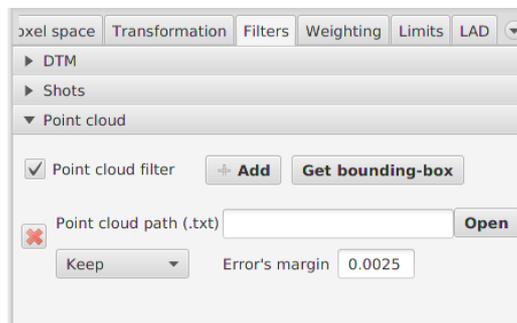
### From one point cloud or more (TLS only)

With this filter, you can keep a point cloud subset from a TLS project in order to sample the voxel space with those points only.  
Basically it allows you to extract a tree from a forest or remove the wood of a tree.



*Illustration 7: Voxelization of an isolated tree*

In order to do that, you have to go to the « Point cloud » panel.  
You can add one or more point file(s) with two possible actions, « Keep » or « Discard ».



*Illustration 8: By point cloud filter section*

For example, you can add a point cloud of a tree (wood and leaves) with the « Keep » action and add the wood point cloud of this tree with the action « Discard ». This will only keep the foliage.

To deal with rounding issues for echos matching research, you can specify the error's margin. A value of 0.0025 is advised, but it depends of the exporting options.

The algorithm uses an octree to perform the matching research, the margin's error is the diameter of the sphere intersecting octree nodes.

### From echo classification

Echo filtering can be performed from « Echo » panel.

ALS : filter with classifications. See format specification for information about the classifications : <http://www.asprs.org/Committee-General/LASer-LAS-File-Format-Exchange-Activities.html>

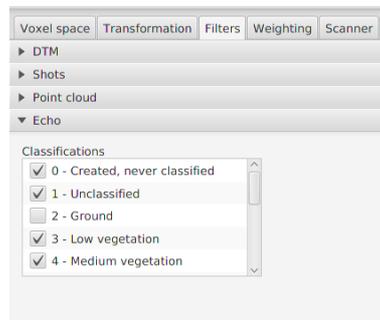


Illustration 9: Echoes filtering section for ALS voxelization

TLS : filter only available for Riscan files (rxp, rsp). The filtering can be performed on the following fields : reflectance, amplitude, deviation

### II.1.4 - Echo weighting

Energy attenuation can be simulated by giving a weight associated to the echo rank. Echoes weighting can be activated in « Weighting » tab, assuming you are in « Voxelization » tab.

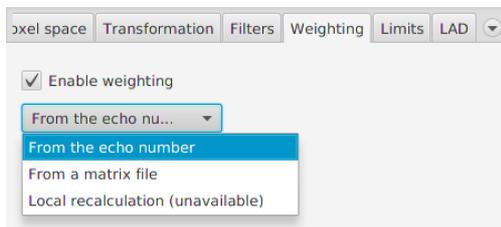


Illustration 10: Echoes weighting tab

The only weighting mode available for the moment is the « by rank » weighting, this is going to weight the echoes according to the number of echoes in shot and echo actual rank. You can also specify a (7x7) matrix file to customize the weighting.

A weighting table is applied, column index corresponding to echo rank and row index corresponding to echo number in the shot.

1.00	/	/	/	/	/	/
0.62	0.38	/	/	/	/	/
0.40	0.35	0.25	/	/	/	/
0.28	0.29	0.24	0.19	/	/	/
0.21	0.24	0.21	0.19	0.15	/	/
0.16	0.21	0.19	0.18	0.14	0.12	/
0.15	0.17	0.15	0.16	0.12	0.19	0.06

1.00	/	/	/	/	/	/
0.50	0.50	/	/	/	/	/
1/3	1/3	1/3	/	/	/	/
0.25	0.25	0.25	0.25	/	/	/
0.20	0.20	0.20	0.20	0.20	/	/
1/6	1/6	1/6	1/6	1/6	1/6	/
1/7	1/7	1/7	1/7	1/7	1/7	1/7

## II.1.5 - Limits

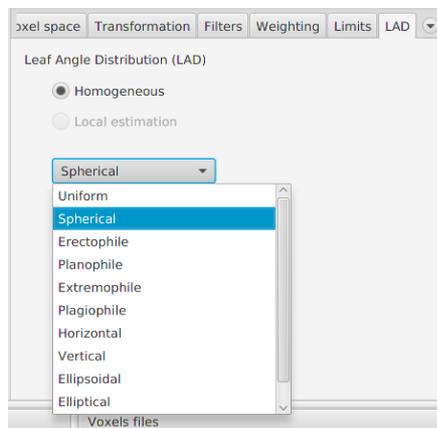
Inside the « Limits » tab you can set the maximum value of PAD (Plant Area Density). A limit is set to trim unlikely high values which may occur notably in ALS mode under locally low sampling intensity.

The default limit is 5 for 1 m<sup>3</sup> voxel size. It is advisable to adjust this parameter based on the voxelization resolution.

## II.1.6 – Leaf Angle Distribution (LAD)

At the moment, we consider an homogenous Leaf Angle Distribution (LAD) inside the voxels.

You can change the LAD type in « Voxelization » section in « LAD » tab.



*Illustration 11: Leaf Angle Distribution tab*

The types of LAD available are as follows: Uniform, Spherical, Erectophile, Planophile, Extremophile, Plagiophile, Horizontal, Vertical, Ellipsoidal, Elliptical, Two-parameter beta distribution.

The leaf phase function  $G(\theta)$  is the projection of foliage area and is characterized by the Leaf Angle Distribution (LAD).

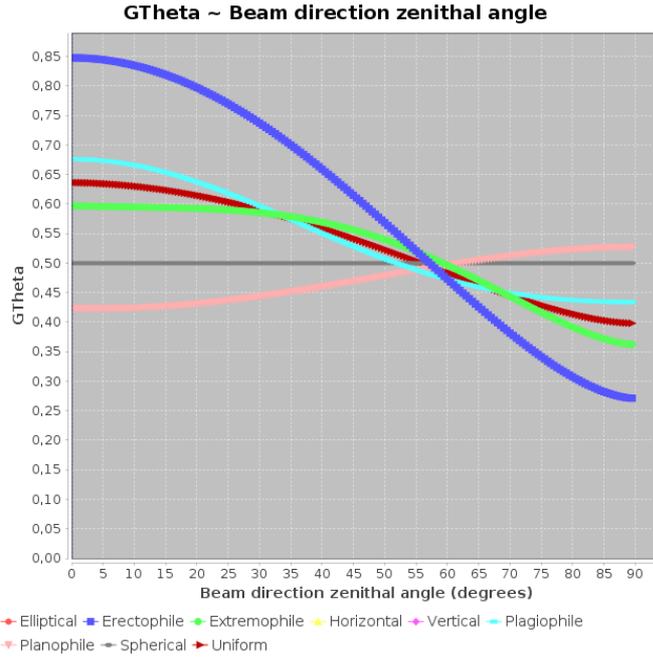


Illustration 12:  $G\theta$  function

$G(\theta)$  is function of the beam direction zenithal angle which is the lidar shot direction.

$$G(\theta) = \sum_j^N \left( \int_{\theta_{j-1}}^{\theta_j} A(\theta, \theta_L) d(\theta_L) \cdot f_j \right)$$

From : Comparison of leaf angle distribution functions: Effects on extinction coefficient and fraction of sunlit foliage W.-M. Wang

$$A(\theta, \theta_L) = \begin{cases} \cos\theta \cos\theta_L, & |\cot\theta \cot\theta_L| > 1 \\ \cos\theta \cdot \cos\theta_L \cdot \left(1 + \left(\frac{2}{\pi}\right) \cdot (\tan\psi - \psi)\right), & \text{otherwise} \end{cases}$$

$$\psi = \cos^{-1}(\cot\theta \cot\theta_L)$$

$N$  is the total number of leaf angle intervals

$f_j$  is the leaf area fraction of interval centered at  $\theta_j$

$f(\theta_L)$  is the probability density function (PDF),  $\theta_L$  the leaf inclination angle in radian. Leaf inclination (zenithal angle) is relative to the surface normal of the leaf.

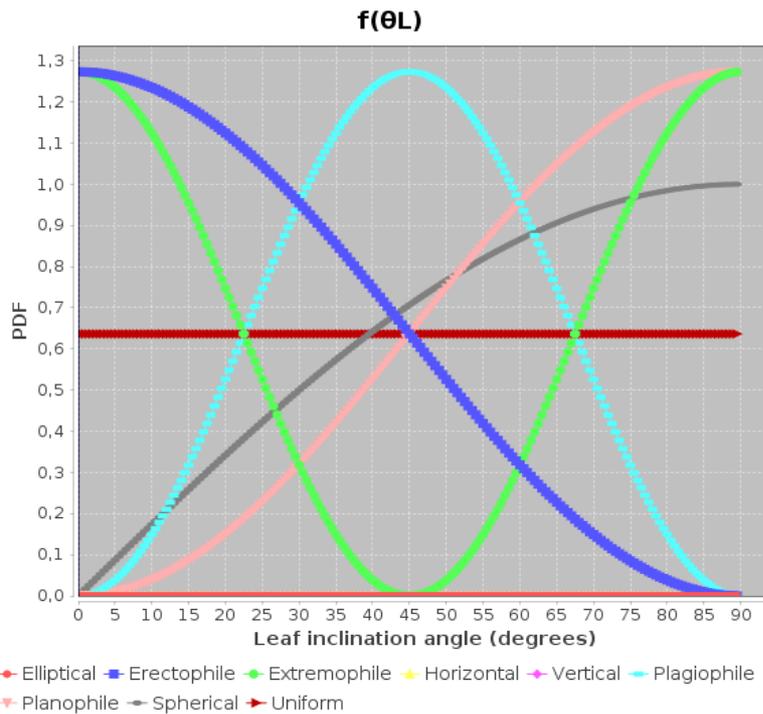


Illustration 13: PDF function

## II.2 - ALS

### II.2.1 – Supported formats

Public formats \*.las and \*.laz only

### II.2.2 - Process

To voxelize ALS data you need to specify an input file (\*.las or \*.laz), a trajectory file (see [file formats in appendix](#)) and an output file.

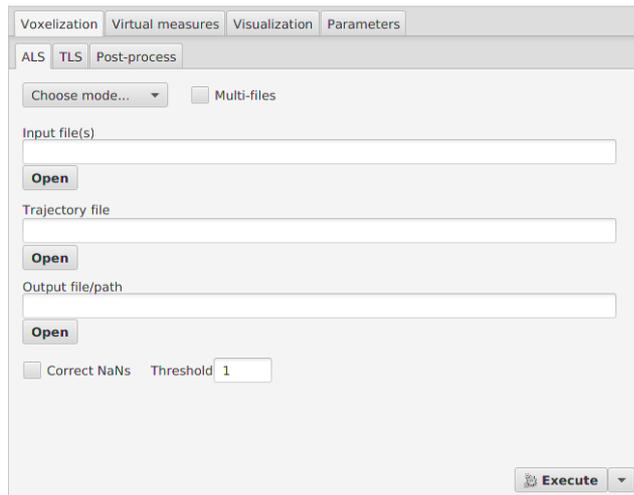


Illustration 14: ALS voxelization tab

Transformation matrices are not mandatory but you should at least configure the bounding-box.

You can automatically generate a bounding-box with « Automatic » button (see [Bounding-box section](#)).

Special : NaNs correction

The « Correct NaN's » checkbox can be selected in order to fill all not informed voxels (transmittance and PAD equals to NaN, meaning the value could not be determined). This happens when voxel sampling is insufficient or null .

See the [algorithm](#) in annex.

## II.3 - TLS

### II.3.1 - Supported formats

Riegl: \*.rxp (scan) and \*.rsp (Project)

Leica / FARO: \*.ptx, \*.ptg

Format specifications :

<http://w3.leica-geosystems.com/kb/?guid=5532D590-114C-43CD-A55F-FE79E5937CB2>

<http://www.xdesy.de/freeware/PTG-DLL/PTG-1.0.pdf>

Others lidar manufacturer can be supported, because some of the formats below are available at export by the different softwares.

### II.3.2 - Process

To voxelize in TLS you need to specify an input file (see [supported formats](#)) and an output file.

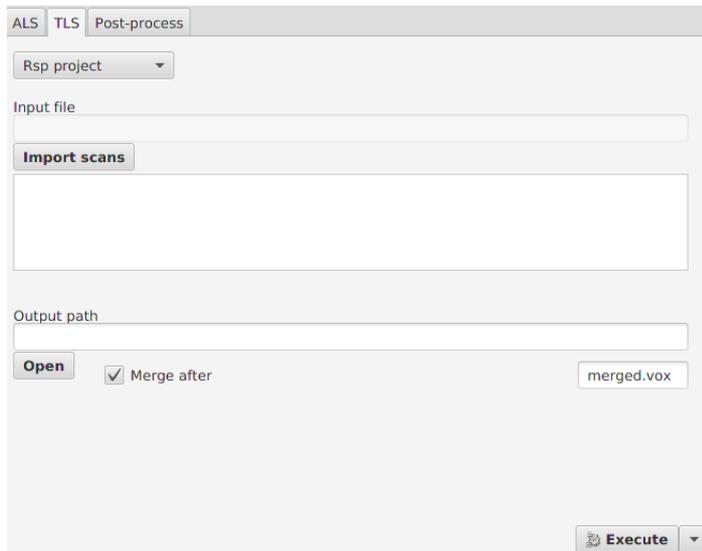


Illustration 15: TLS voxelization tab

Into Riscan project, you can have decimated scan files.  
They are recognizable by their names that contains «mon ».  
A « mon » file is a decimated file of the original (~0.5%)

In order to check parameters, it is advised to begin a voxelization on those files before launching voxelization process on the original much larger files.



In Riscan project, only registered scans will be processed

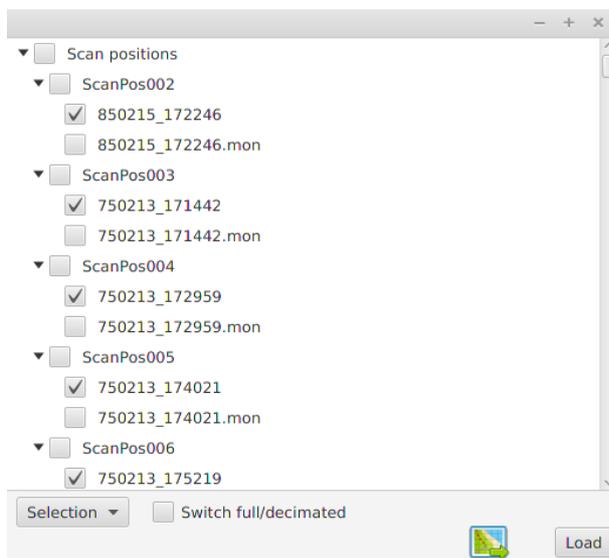


Illustration 16: Riscan project import frame

### The transformation process

The transformation applied to a shot origin is the result of the following matrix product :  
SOP %\*% POP %\*% VOP

The rotation part of this matrix is applied to the direction vector of the shot.

*Execution time on 8 cores CPU :*

*Parameters: Voxel space splitting : 24x144x70  
Scan number : 98  
Average points by scan : 22 millions*

*Decimated scans : 1 minute  
Full points scans : 35 minutes*

### Merging

In TLS, we generate a voxel space file for each scan.  
Then we apply a voxel space merging to reconstruct the full file.

The merging process can be performed with two options :

1-First is to merge after generation of voxel files, you just need to check the check-box « merge after » in « TLS » tab and give the output file name.

2-The second option is merging the voxel space files in post-process a separate mode and you need to select voxel files to merge.

You have to go in « Voxels files » panel, select the « merging » tab, drop your files in the list and choose an output file directory and name.

Next thing to do is generating a configuration file. Click on the button « Save», which create a configuration file and add it to « Tasks » panel on the bottom left.

Finally click on the «Execute » button.

You can see in appendix the merging algorithm.

## II.4 - Dart export



Dart formats are going to be changed by the developer team soon.

Dart is a software developed by Cesbio (<http://www.cesbio.ups-tlse.fr/>). « It models radiative transfer in the system « Earth - Atmosphere », from visible to thermal infrared».

Dart can create model files as an input from its graphical user interface.

By creating/editing a model from the interface a « plots.xml » file is created/edited in the input folder of the Dart project simulation.

By running the module « Maket » in Dart it will read the file « plots.xml » located in the input folder and generate the file « maket.txt ». This file will be written in the output folder of the Dart project simulation.

AMAPVox can convert voxel file to Dart file (« maket.txt » or « plots.xml ») so it can be used as an intermediary.

By analogy, a voxel from voxel file is a cell in the Dart file.

Cell of vegetation type contains LAI (Leaf Area Index) information, which is the surface density ( $\text{m}^2$  leaf /  $\text{m}^2$  of ground per plot) .

Voxel file contains PAD (Plant Area Density), it is converted into LAI by the formula :

$$LAI(\text{cell}) = \text{cell's PAD} * (\text{cell size } z)$$

To export a voxel file in Dart format :

You need to open a voxel file from « Voxels files » list in bottom right corner and click on « Export » button then « Dart » choice.

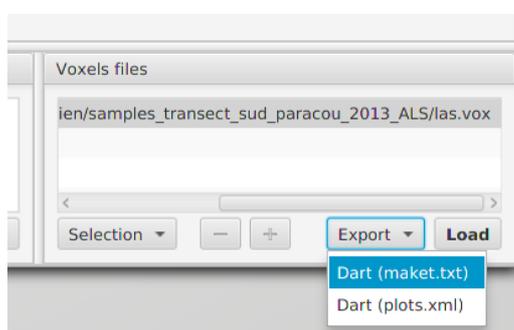


Illustration 17: Export voxel file button

There are two output formats :

-A « plots.xml » file

-A « maket.txt » file + in option a « triangles.txt » file.

The first describes voxelized space in xml format, this file need to be moved inside the « input » folder of the Dart simulation.

The second represents voxelized space in maket Dart format, a text file.

### II.4.1 – plots.xml

There are 5 types of plots/cells in Dart « plots.xml » file:

- ground (type = 0)
- vegetation (type = 1)
- ground + vegetation (type = 2)
- Fluid (type = 3)
- Water (type = 4)

AMAPVox exports only the vegetation type, ground exports is not available yet.

### Leaf Phase Function

You can assign a leaf phase function to each vegetation cell by editing the « plots.xml » file generated by AMAPVox.

The « VegetationOpticalPropertyLink » tag of each plot contains the index and id of the phase function.

Phase functions can be created from Dart in the « Optical and temperature properties » section.

When you create a vegetation phase function the index value is 0 for the first created, 1 for the second, etc.

Leaf phase function is characterized by the Leaf Angle Distribution (LAD).

See LAD section (1.1.6) for further information.

Process :

Replace « plots.xml » (located in the input folder) of Dart simulation project by the generated one.

For dealing with the inability to exports ground plot, you can import a DEM from Dart interface inside the earth scene section.

The DEM needs to have the same size as the scene dimensions, and the supported format is the Tiff format with the tfw aside.

### II.4.2 – maket.txt / triangles.txt

From Dart documentation : « Maket.txt file stores the 3D cell matrix of the Dart scene. » It contains the simulated scene generated from « plots.xml » and various parameters.

The plot/cell types are many, so a short list below :

- Empty cell (type = 0)
- Opaque :
  - Air (type = 1)
  - Ground (type = 2)
  - Water (type = 3)
- Opaque DEM :
  - Ground\_DEM (type = 7)
  - Plot\_DEM (type = 14)
- Turbid :
  - Crown (type = 6)

Opaque DEM category is like the Opaque one but we specify here that the plot contains DEM.

AMAPVox exports « Ground\_DEM » and « Crown » types.

The ground type is exported if the following conditions are satisfied :

- DTM filter has been configured for the voxelization process.
- All voxels over the ground will be exported as a cell with opaque ground type.

and - « triangles.txt » export option is enable, and the original ascii grid file (.asc) has been set up (the one used for voxelization as the voxel space doesn't contains the DTM).

A cell can have multiple LAI values and to each one a leaf phase function is bounded. AMAPVox exports a single LAI value. Phase functions are created from Dart interface.

The leaf phase function is represented by an index. Phase functions can be created from Dart in the « Optical and temperature properties » section.

When you create a vegetation phase function the index value is 0 for the first one created, 1 for the second, etc...

At the moment, the leaf phase function index exported by AMAPVox is 0, assuming an homogeneous canopy.

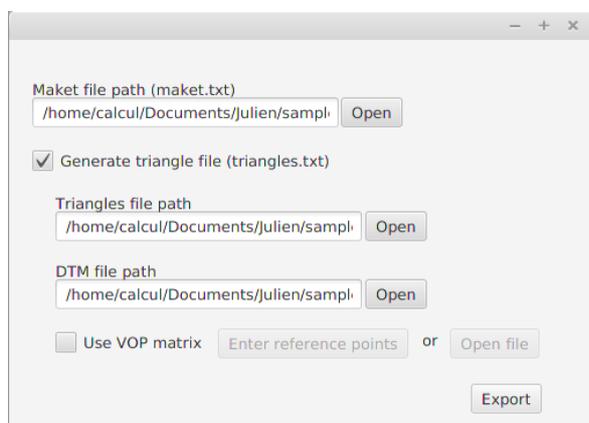


Illustration 18: Export options for dart format

### III – Virtual measures

Contributor : Jean Dauzat

AMAPVox allows you to perform virtual measures from voxel files.

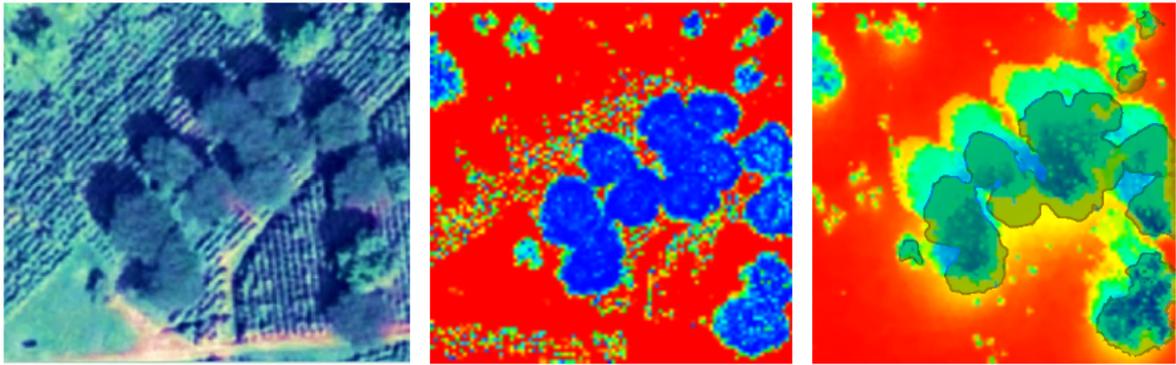
#### III.1 – Transmittance light map

Generate a bitmap/text file of the transmittance light from a voxel file ,sensors and sun parameters.

Calculated the light transmitted by ray-tracing through the voxel space in a set of directions according to the Den Dulk's « TURTLE » model (Den Dulk, 1989) and for a set of positions.

Light extinction is calculated using the Beer-Lambert law.

Each direction belongs to an hemisphere sector and the amount of direct and diffuse components of the global radiation is computed for each sector from the clearness index set for a period of time and the plot latitude.



*Illustration 19: Plot of Casha. To the left an aerial view; to the middle a voxel space view; to the right the simulated light map transmittance with the tree crowns overlap. Blue areas are the most shaded and red areas are the most insolated.*

### III.1.1 – Input parameters

**Input file** – A voxel file

**Output**

**Generate text file** - Output file for the simulation result.

**Generate bitmap files** – Output directory for the transmittance light maps, one for each simulation period.

**Sensor(s)**

**Positions** – The sensors positions.

**Directions number** - The split number of the hemisphere.

**Repetitive scene** – Rays outgoing of the plot are extended starting from the opposite side. (toricity)

**Radiative conditions**

**Latitude** – Geolocalization of the plot.

**Simulation periods** – Periods of times to simulate, each period of time have a clearness index.

### III.2 – Canopy analyzer

Simulates LAI2000/2200 sensors from a voxel file and sensor parameters.

Generates lai2000/2200 output.

#### III.2.1 – Input parameters

Input file

Output file

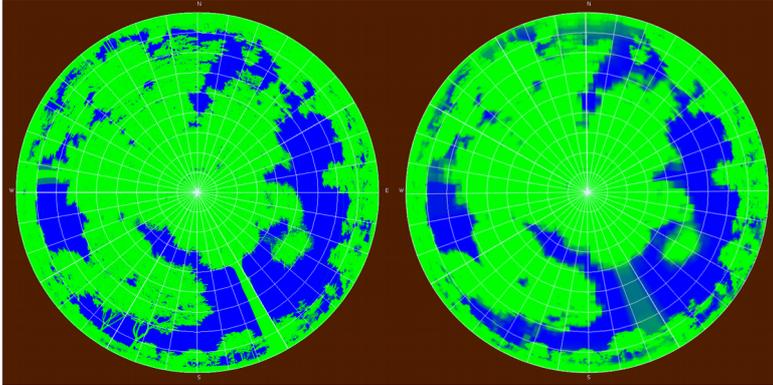
Directions number

Rings mask

Sensor positions

### III.3 – Hemispherical photography

Generates hemispherical photography from a voxel file or a single lidar scan.



*Illustration 20: To the left an hemispherical photography simulated from a lidar scan and to the right simulated from PAD computed at the plot scale.*

#### III.3.1 – Input parameters

Input file  
Output file  
Pixel number  
Azimuts number  
Zeniths number

### IV – Visualization

There are currently two modes of vizualisation, mode 1 and mode 2.

The first one allows you to visualize a voxel file with an optional dtm.

In this mode you can filter voxel attributes by values, change world parameters like sky color, camera settings...etc.

In the second mode you can visualize pointcloud data, rasters and voxel file at the same time. You can change the color gradient and the currently displayed scalar field.

The supported formats are :

Point cloud : las, laz, rxp, rsp, ptx, ptg, ascii (csv)

Raster : ascii grid format (asc)

Voxel file : AMAPVox voxel format

This mode is not fully implemented. You can't filter the values, edit the gradients or apply a transformation.

## IV.1 - Load a file

First mode :

- 1 - In Voxels files panel, open a voxel file by clicking on « + » button or drag and drop.
- 2 - In « Vizualisation » panel, choose an attribut.
- 3 - Click on « Open display window » button.

Second mode :

- 1 - In « Vizualisation » panel, open a file by clicking on « + » button in the second mode section.
- 2 - Click on « Open display window » button.

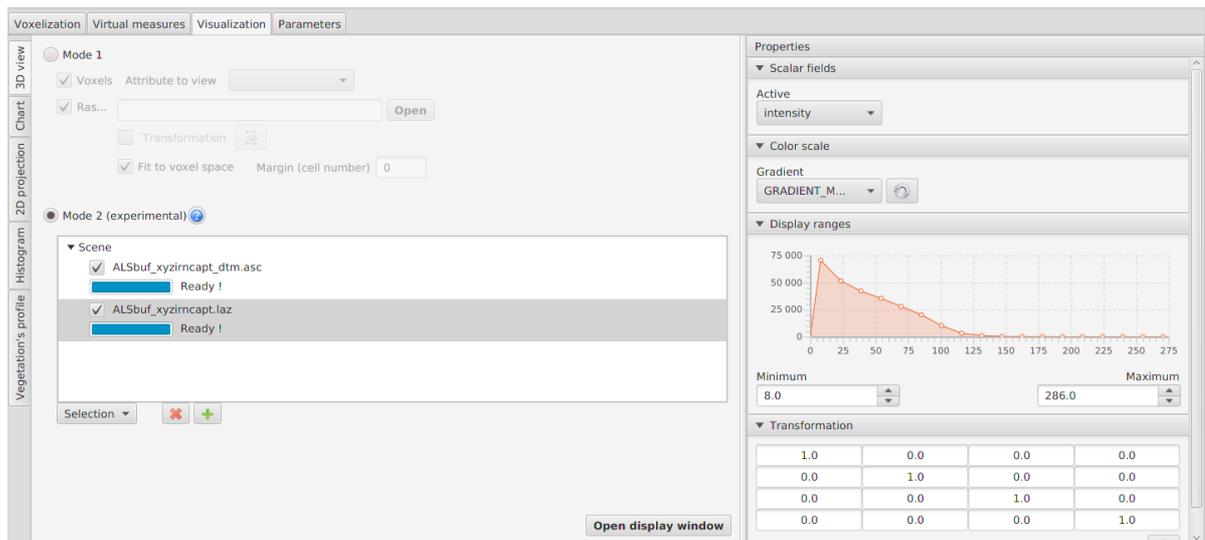


Illustration 21: Visualization tab - The displaymode 2 is active. To the left the loaded scene objects, to the right the currently displayed scalar field and the range of values.

## IV.2 - Visualize a raster

First mode :

You can in the « Vizualisation » tab choose a raster file, typically a DTM (Digital Terrain Model), in esri ascii grid format (.asc).



Illustration 22: Raster import parameters - visualization mode 1

Second mode :

Add the file to the scene object tree view

All modes :

The « Transformation » option allows you to apply a transformation to the raster object.

You can also restrict the DTM surface to the linked voxel file with the button « Fit to voxel space ».

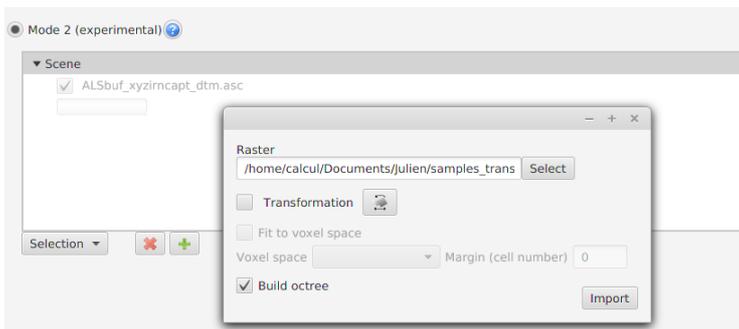


Illustration 23: Raster import parameters - visualization mode 2

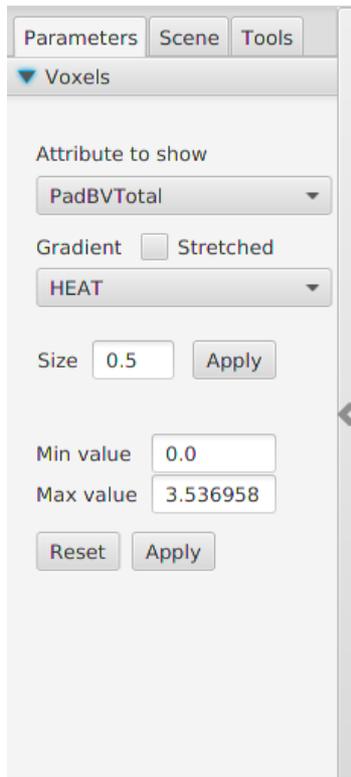
IV.3 – Toolbox

 First mode only !

IV.3.1 - Voxel viewing parameters:

Voxel section :

**Attribute to show** : Voxel attribut to represent (in a colouring way)



**Gradient** : Color gradient

**Stretched** : Stretch colors in order to use the entire color scale available.

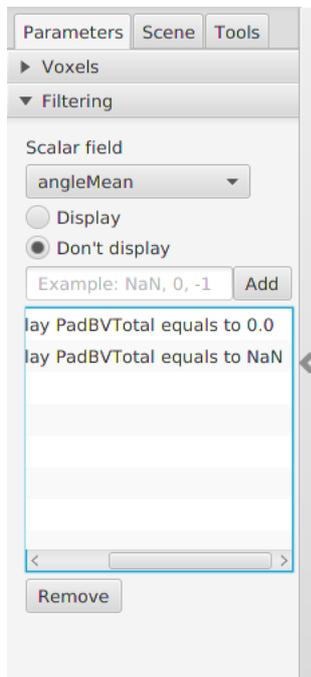
It may enhance readability but color scale is no longer valid.

**Size** : voxel size in unit meters, default value is the half of the voxel edge actual size.

**Min value/ Max value** : Define color scale from a value scale, restrict true value interval.

Illustration 24: Toolbox - legend section

**Filtering section** : Display or not voxels according to their scalar fields values.



**Syntax** :

Filter values from 4 included to 5 excluded : [4->5[

Filter unknown values : NaN

Filter values 1 and 2 : 1, 2

Illustration 25: Toolbox - Filtering section

## IV.3.2 – Scene

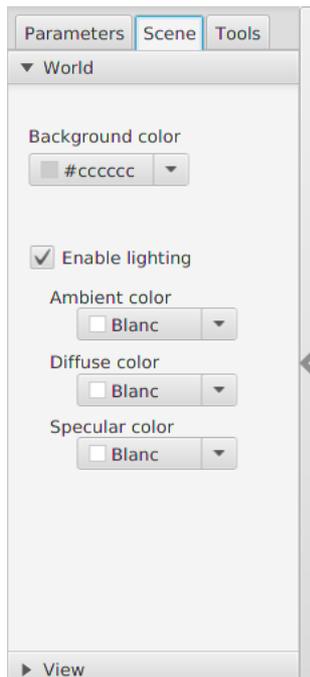


Illustration 26: 3D view - world settings

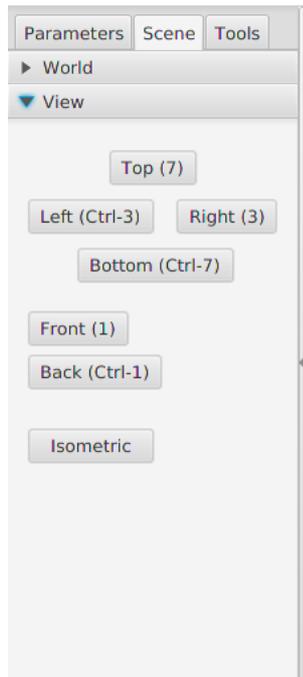


Illustration 27: 3D view - view projection

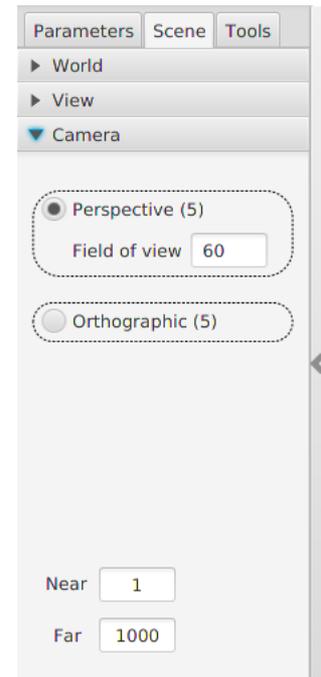


Illustration 28: 3D view - camera settings

**Background color** : the sky color

**Perspective** : perspective view, you can choose the field of view

**Orthographic** : orthographic view

**Near/Far** : Minimal/maximal drawing distance of the camera. Too low or too high values can create visual artefacts.

## IV.3.3 - Tools

Sectional view : Cross section view of the voxel space.

+ and – buttons move forward and backward inside voxel space according to camera position.

## V - Annex

### V.1 - File formats

#### V.1.1 - Point files (.las)

Public binary file format for the exchange of 3-dimensional point cloud data.

Format specifications : <http://www.asprs.org/Committee-General/LASer-LAS-File-Format-Exchange-Activities.html>

### V.1.2 - Compressed point files (.laz)

Compressed format of \*.las file, the tool to generate those files has been developed by Martin Isenburg and is integrated in the LASTools software.

This can be downloaded here:

<http://www.laszip.org/> or <http://www.cs.unc.edu/~isenburg/lastools/>

### V.1.3 - Trajectory file

Contains GPS position at a given time.

Existing format are \*.pof (binary, not supported) and custom text format.

The text file needs to have four columns with the easting, northing, elevation and time informations.

AMAPVox will use the time as an index to retrieve the linked point records, easting/northing/elevation gives the location of the lidar and should be in the same coordinates system as the points data records.

```
Easting[m],Northing[m],Elevation[m],Time[s]
349104.142,533198.318,6.660,305247.002716
349104.143,533198.303,6.660,305247.007717
349102.636,533202.338,9.682,305247.012716
349102.636,533202.323,9.682,305247.017715
```

*Example: extract of a trajectory file*

### V.1.4 - Riscan scan file (.rxp)

Binary and proprietary file format owned by Riegl.

Those files contain point cloud and other data and can be read with the RivLib library provided by Riegl or with Riscan Pro software.

### V.1.5 - Riscan Project (.rsp)

XML file format containing project information.

Containing in particular scans file paths, POP (Project Orientation and Position) matrix and SOP (System Orientation and Position) matrices for each scan.

### V.1.6 - DTM (Digital Terrain Model)

A DTM file represents height information of the ground.

At the moment, the only supported format is the Esri ascii grid (\*.asc).

This file can be generated by LASTools from als data:

- use lasground to classify ground points and non ground points
- use las2dem to generate the asc file : keep only the ground classified points (class = 2)

### V.1.7 - Voxel file

« Voxel » file contains a signature line, four meta-date lines and a header line.

```

VOXEL SPACE
#min_corner: -24.55 -19.08 9.21
#max_corner: 21.12 267.67 68.98
#split: 45 286 59
#type: ALS #resolution: 1.0
i j k Pad angleMean bvEntering bvIntercepted ground_distance lMeanTotal lgTotal nbEchos
nbSampling transmittance

```

« min\_corner » and « max\_corner » parameters are the extremums of the bounding box. The « split » parameter indicates voxel number along the 3D axis, it is directly related to resolution and extremums.

The parameter « type » can be either ALS or TLS.

Header line indicates columns names.

i	Voxel index along the x axis
j	Voxel index along the y axis
k	Voxel index along the z axis
Pad	Plant Area Density (m <sup>2</sup> /m <sup>3</sup> )
angleMean	Mean inclination angle of shots which entered the voxel
bvEntering	Sum of weighted fractions (optical path, divergence, attenuation) of entering laser pulses
bvIntercepted	Sum of weighted fractions (optical path, divergence, attenuation) of intercepted laser pulses
ground_distance	Distance from voxel center to the ground (can be negative) If the ground is not set (no DTM filter), this value is the height of the voxel relative to a plane with equation Z = 0
lMeanTotal	Mean length of optical path inside the voxel
lgTotal	Length sum of optical path inside the voxel
nbEchos	Total echoes count inside the voxel
nbSampling	Number of (possibly attenuated) pulses entering the voxel
transmittance	Transmittance (standardized per one meter optical path length)

### V.1.8 - Matrix file

Transformation matrix

```
0.95 0.29 0.0 -448120.04
-0.29 0.95 0.0 -470918.39
0.0 0.0 1.0 0.0
0.0 0.0 0.0 1.0
```

Splitting characters can be space or tabulation.  
File may contain line returns or not.

#### Weighting table

```
1.00, NaN, NaN, NaN, NaN, NaN, NaN,
0.62, 0.38, NaN, NaN, NaN, NaN, NaN,
0.40, 0.35, 0.25, NaN, NaN, NaN, NaN,
0.28, 0.29, 0.24, 0.19, NaN, NaN, NaN,
0.21, 0.24, 0.21, 0.19, 0.15, NaN, NaN,
0.16, 0.21, 0.19, 0.18, 0.14, 0.12, NaN,
0.15, 0.17, 0.15, 0.16, 0.12, 0.19, 0.06
```

The weighting table should contain seven rows and seven columns.  
NaN means the cell is not used.

The column indicates the echo rank and the row is the echo number of the corresponding shot.

#### V.1.9 - Configuration file

XML format, its content is function of the task type.  
Minimal structure:

```
<configuration>
  <process mode = "[mode_value]" type= "[type_value]">
  </process>
</configuration>
```

Possible values for « mode » : voxelization, multi-resolutions, merging

Possible values for « type » : ALS, TLS

*Example 1: Configuration file in « voxelisation » mode and « ALS » type*

```
<configuration>
  <process mode = "voxelisation" type= "ALS">

    <input_file type = "0" src="/home/Documents/input_file.las"/>
    <trajectory src="/home/Documents/trajectory_file.txt"/>
    <output_file src="/home/Documents/output_file.vox"/>

    <voxelspace xmin="-12 .0" ymin="-2 .0" zmin="8.0" xmax="12.0"
ymax="142" zmax="72" splitX="24" splitY="144" splitZ="64" resolution="1.0" />

    <ponderation mode="1"/>
      <matrix type_id="ponderation">
        1.0 NaN NaN NaN NaN NaN NaN
        0.62 0.38 NaN NaN NaN NaN NaN
        0.4 0.35 0.25 NaN NaN NaN NaN
        0.28 0.29 0.24 0.19 NaN NaN NaN
        0.21 0.24 0.21 0.19 0.15 NaN NaN
        0.16 0.21 0.19 0.18 0.14 0.12 NaN
        0.15 0.17 0.15 0.16 0.12 0.19 0.06
      </matrix>
    </ponderation>
    <dtm-filter enabled="true" src="/home/Documents/dtm.asc" height-min="1.0" />
    <transformation use-pop="false" use-sop="false" use_vop="true" >
      <matrix type_id="vop">
        0.95, 0.29, 0.0, -448120
        -0.29, 0.954, 0.0, -470918
        0.0, 0.0, 1.0, 0.0
        0.0, 0.0, 0.0, 1.0
      </matrix>
    </transformation>
    <limits>
      <limit name="PAD" min="" max="5.0" />
    </limits>
  </process>
</configuration>
```

<ponderation> tag :

« mode » value is an integer and means weighting type (0 : no weighting ; 1 : echo rank weighting).

<dtm-filter> tag:

This filter removes echoes below the ground , if it is enabled, the DTM path is set in « src » attribute and the height threshold in the « height-min » attribute.

The height threshold expressed in meter (m) is the threshold base from which an echo is considered as ground. If it that case it will be removed from vegetation analysis.

<transformation> tag :

Apply a transformation to the input file, the value is a 4x4 matrix.

<limits> tag:

Maximum PAD value.

Example: Configuration file for « merge » process and « TLS » type

```

<configuration>
  <process mode = "merging" type= "TLS">
    <output_file src="/home/Documents/merged.vox"/>
    <files>
      <file src="/home/Documents/tls_scan1.vox"/>
      <file src="/home/Documents/tls_scan2.vox"/>
      <file src="/home/Documents/tls_scan3.vox"/>
      <file src="/home/Documents/tls_scan4.vox"/>
    </files>
  </process>
</configuration>

```

## V.2 – Algorithms

### V.2.1 - Merging trajectory and las (ALS)

The trajectory file is in text format. Merging is based on the GPS time and laser time. Laser positions are determined by linear interpolation between two GPS clock ticks.

### V.2.2 - Transmittance processing (ALS+TLS)

The calculation of local transmittance (into a voxel) involves a weighting which aims at taking into account the appropriate contribution of each entering shot in the voxel. This weighting consists of three terms :

- Optical length trajectory in the voxel (always)
- The size of the footprint function of beam divergence and distance from the source (optional)
- The signal extinction because of previous obstacles (optional)

In TLS case, a fundamental indetermination exists because the non intercepted beam fraction is usually unknown.

$$Tr_{norm} = \frac{\sum_i^n \frac{PFOut_i}{PFEnt_i} \frac{1}{l_i} \cdot S_i \cdot l_i \cdot PFEnt_i}{\sum_i^n S_i \cdot l_i \cdot PFEnt_i}$$

with :

$PFOut_i$ , exiting fraction of pulse i

$$PVOutgoing_i = PVEntering_i - PVIntercepted_i$$

$PVEntering_i$ , incoming fraction of pulse i

$l_i$ , length of pulse i optical path

$s_i$ , cross section of pulse at voxel center (calculated assuming no fragmentation)

Weighting

Two options are available (minimal and total i.e. with or without weighting).  
 Total weighting includes signal extinction.  
 With minimal weighting,

$PF_{Outgoing_i} = 0$  if an echo of the shot is inside the voxel, 1 otherwise  
 $PF_{Entering_i} = 1$  if shot penetrate the voxel, 0 otherwise

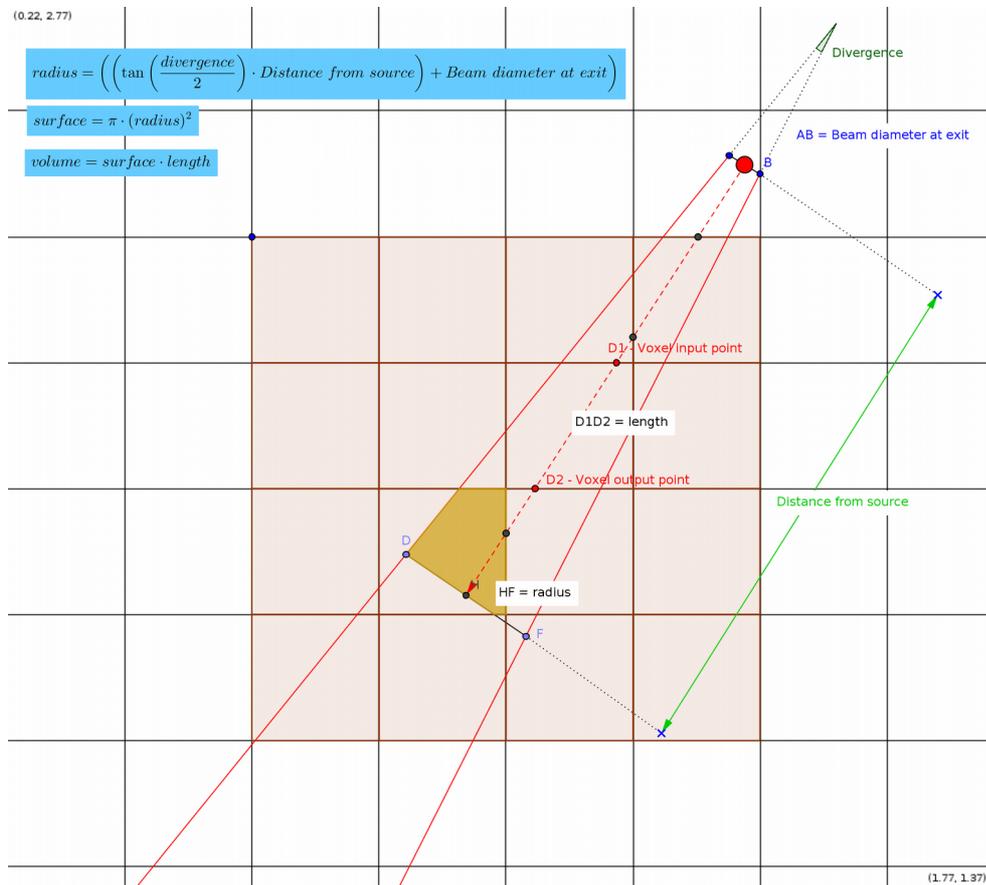


Illustration 29: Variables details (divergence angle is exaggerated for a better comprehension)

(0.09, 2.76)

$$BV_{\text{Entering}} = \sum_i^n (\text{surface} \cdot \text{length} \cdot \text{residual energy})$$

$$BV_{\text{Intercepted}} = \sum_i^n (\text{surface} \cdot \text{length} \cdot \text{beam fraction})$$

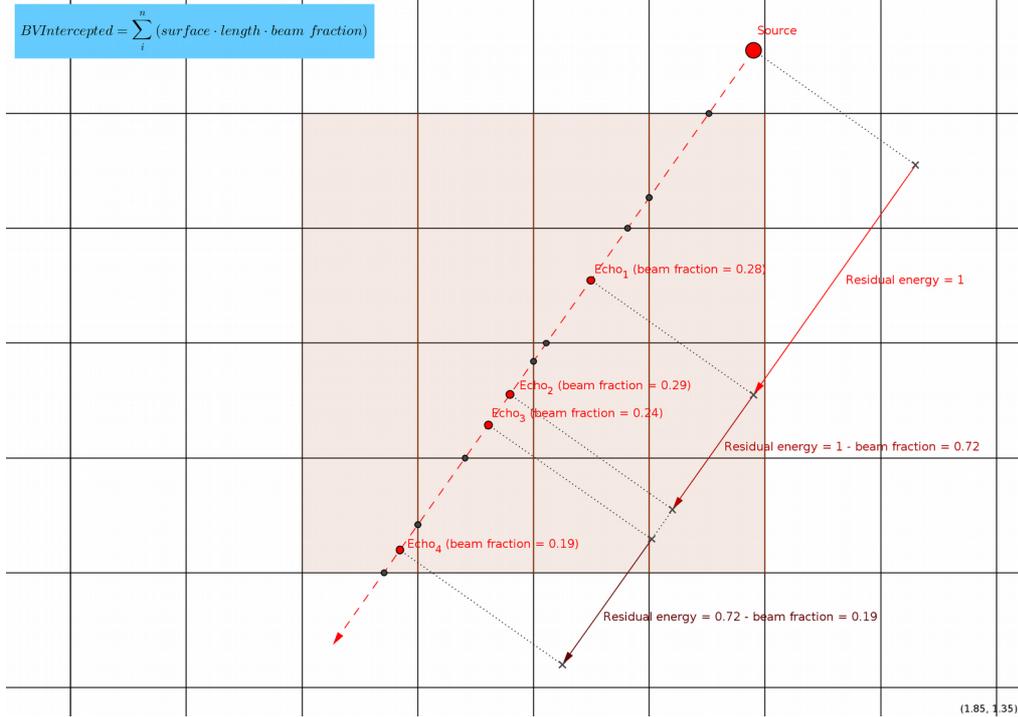
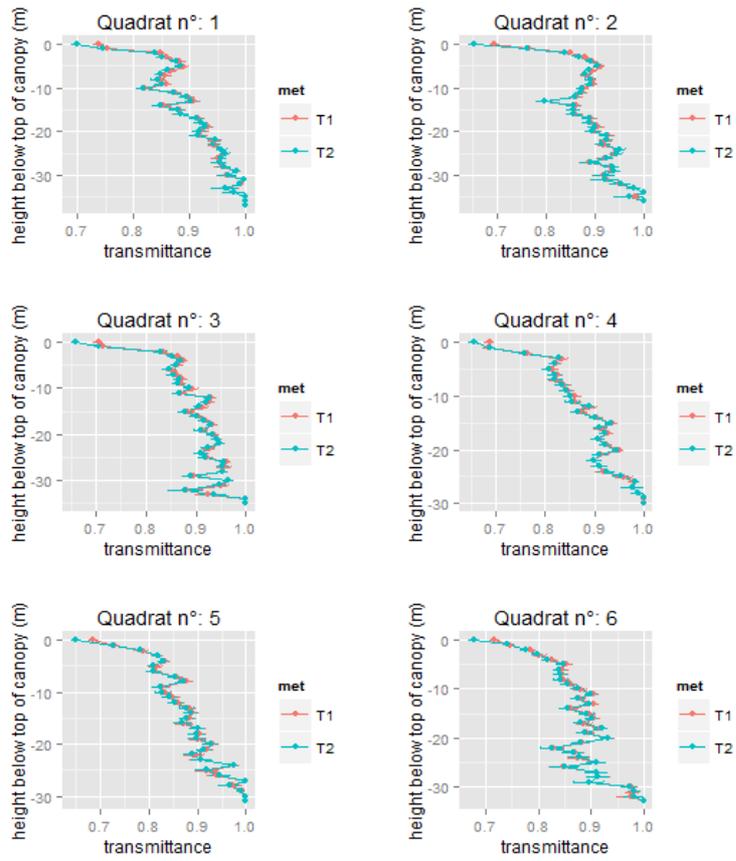


Illustration 30: Residual energy / beam fraction



*Illustration 31: Comparison of weighting modes influence on the transmittance of ParacouP15 (ALS)*

### V.2.3 - Computation of the optical path (ALS+TLS)

The computation of the optical path into a voxel is always the same. If a pulse is not entirely stopped then the path length is computed from the distance of the entering point to the outgoing point in the voxel. If the last echo from the shot occurs within the voxel then the optical path is computed from the distance of the entering point to the last echo.

Ground echo filtering

A Digital Terrain Model must be provided by the user (esrii grid format \*.asc).

All echos having with heights less than or equals to the raster altitude (+threshold) at the corresponding location are considered as non vegetation echoes and are exclude of the vegetation analysis.

Pulse tracing will be extended to the last echo meaning that pulse trajectory before hitting a non vegetation target is taken into account (contributes to transmittance estimate).

When computation transmitted energy to the ground, the information of the « ground type voxel » information is used (cf below).

Computation of transmitted energy to the ground

All ground voxels are derived from ground raster.

We extend the optical path length from the last echo until the first ground voxel.

Then we count for each corresponding raster cell the shot number which could potentially have reached the voxel (in the absence of vegetation) and the effective energy reaching reaching that cell.

The transmitted energy is the ratio of these two sums.

At present, the PAI calculation doesn't distinguish wood from leaves and assume a spherical angular distribution of the vegetation elements.

The formula is :

$$PAD = \min \left( \left( \frac{\log(transmittance)}{-0.5} \right), PAD_{max} \right)$$

With : PAD<sub>max</sub>, maximal eligible value (the default value is 5 for one m<sup>3</sup>).

This value depends of the resolution.

## V.2.4 – Resampling voxels (ALS)

```

For each (« voxel » from voxel space):

    current_nbSampling = voxel.nbSampling
    current_transmittance = voxel.transmittance

    While (current_nbSampling ≤ threshold or current_transmittance == 0):

        neighbours list = Get list of « neighbour » of the voxel according to Moore neighborhood, which comply with the following criteria : neighbour voxel is above the ground and below the top of canopy

        If (size (neighbours list) > 0) :

            sum_BVEntering = sum (neighbours.bvEntering)
            sum_bvEntering = sum (neighbours.bvEntering)
            sum_bvIntercepted = sum (neighbours.bvIntercepted )
            sum_lgTotal = sum (neighbours.lgTotal)

            current_transmittance =  $\frac{\sum BVEntering - \sum BVIntercepted}{\sum BVEntering} \frac{\sum BVEntering}{\sum lgTotal}$ 
            current_nbSampling = mean_nbSampling / size (neighbours)

        Else :
            current_nbSampling = 0

    If (size (neighbours list) > 0) :

        voxel.PadBVTotal = mean (neighbours.PadBVTotal)
        voxel.nbSampling = current_nbSampling
        voxel.transmittance = current_transmittance
    
```

*Illustration 32: Algorithm for fixing missing values*

## V.2.5 - Scan merging (TLS)

When data merging of different scans is processed, the following rules are applied :

field	method
$BVEntering = \sum_i^n BVEntering_i \cdot S_i \cdot l_i$	sum
$BVIntercepted = \sum_i^n BVEntering_i \cdot S_i \cdot l_i - \sum_i^n BVOutgoing_i \cdot S_i \cdot l_i$	sum
nbSampling	sum
nbEchos	sum
LgTotal	sum
AngleMean (angle = arcos(z_u))	Weighted mean (weight=occurrence number for a voxel of a scan / total

	occurrence number for all the scans for this voxel) Occurrences=nbSampling
LMean; PAD ; transmittance => recalculated after merging	$Tr_{norm} = \frac{\sum_i^n \frac{PFOut_i}{PFEnt_i} \cdot S_i \cdot l_i \cdot PFEnt_i}{\sum_i^n S_i \cdot l_i \cdot PFEnt_i}$ $PAD = \min \left( \left( \frac{\log(transmittance)}{-0.5} \right), PAD_{max} \right)$
Others (i,jk, hag, x,y,z,...)	No changes