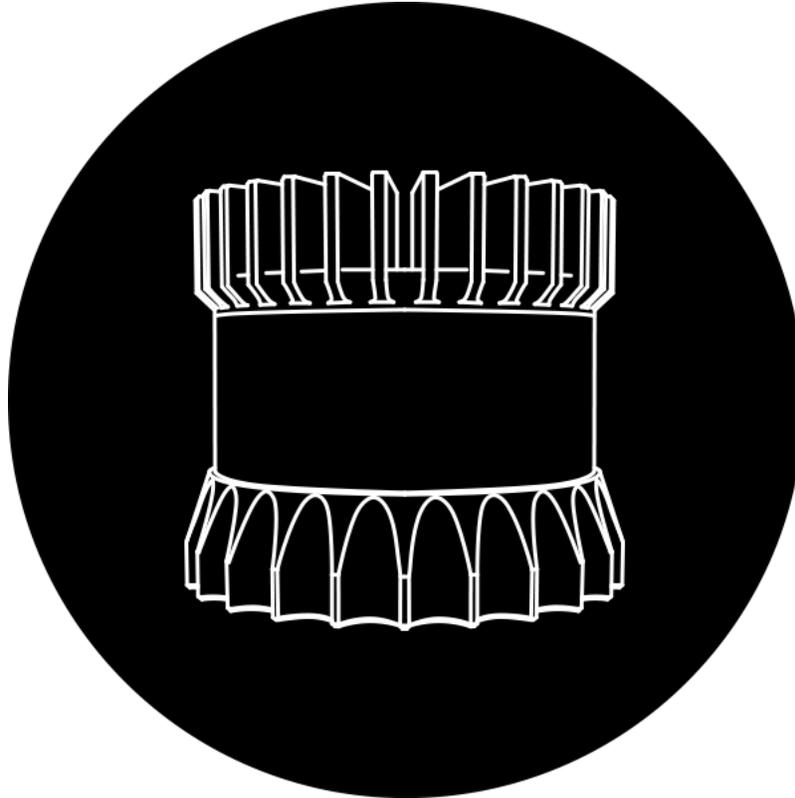


# Ouster Studio User Guide



*v1.0.1*

*Last Updated: July 16, 2019*

# I. Introduction

Ouster Studio is software provided by Ouster to visualize, record, and analyze data from Ouster lidar sensors. Ouster Studio is cross-platform, with official support for Windows, MacOS and Ubuntu. The software performs real-time visualization, processing, and recording of live 3D lidar data captured from Ouster lidar sensors. Ouster Studio is also able to replay stream data stored in .pcap files recorded from live streams directly within Ouster Studio.

This manual provides a guide to Ouster Studio's graphical interface. In particular, it will cover the acquisition, visualization, and analysis of live and pre-recorded data obtained from an Ouster lidar sensor. The guide assumes that the user already has access to an installed sensor or pre-recorded sensor data. If this is not the case, please contact your Ouster hardware provider for assistance.

If you do not have a copy of Ouster Studio, please visit (<https://www.paraview.org/ousterstudio/>) to download. If you require assistance with installing the application, please contact your Ouster hardware provider.

## II. Visualize and analyze data

There are two modes for visualizing data in Ouster Studio: live stream mode and playback mode. Live stream mode corresponds to the situation where an Ouster sensor is directly accessible over the network from the computer running Ouster Studio. In this case, the live stream mode enables visualization of the current output of the sensor. The playback mode corresponds to the situation where the user has recorded a live stream in .pcap format and wants to replay it.

### A. Overview of the user interface

The Ouster Studio user interface is composed of 6 toolbars (areas 3 to 8 in figure I.A.1) and three different views: the 3D view (area 1 in the image), the 2D view (area 2 in the image), and the spreadsheet view (not shown in the image).

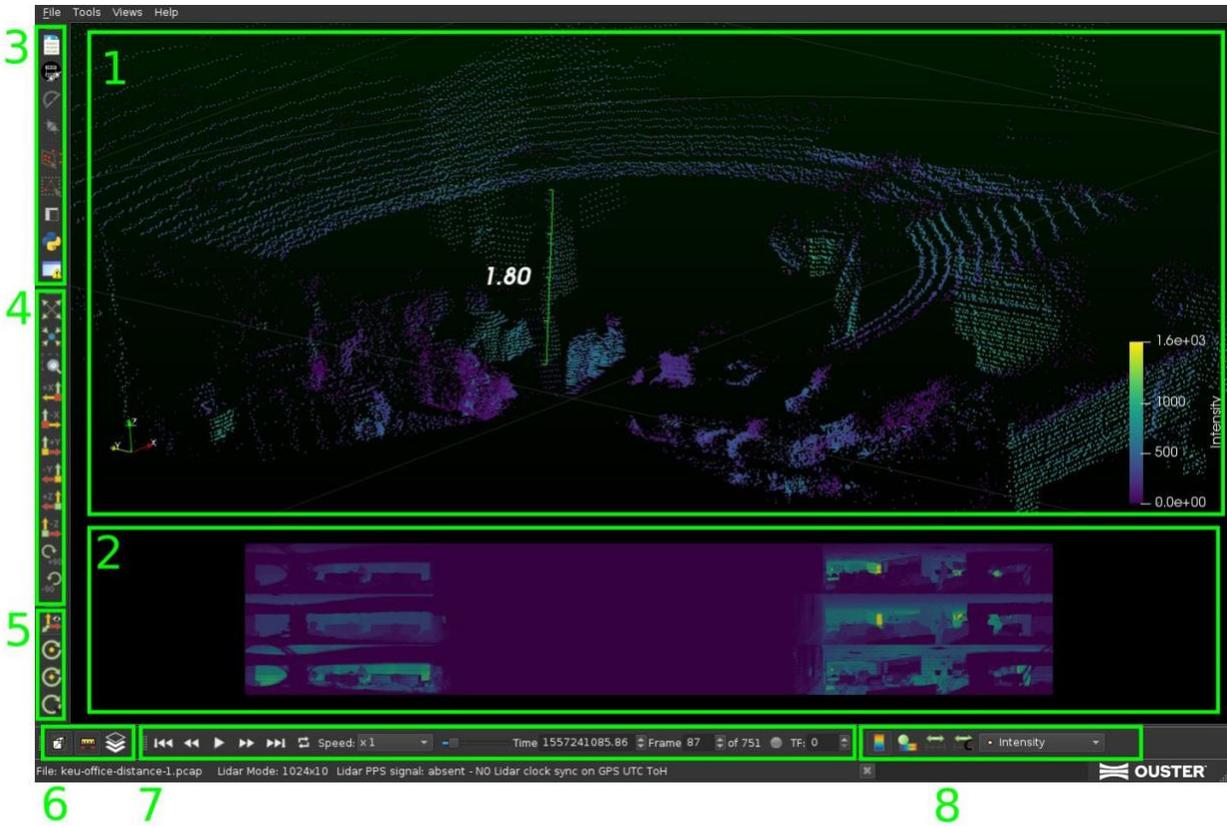


Figure II.A.1: Overview of Ouster Studio's graphical interface:  
 1) 3D View of point cloud data, 2) 2D 360° image view, 3) Basic control toolbar, 4 & 5) View toolbars,  
 6) Measurement and projection toolbar, 7) Player control toolbar,  
 8) Colormap toolbar.

### A.1 3D view

This shows the 3D points in an interactive viewer. To rotate the displayed area, move the mouse while holding the left mouse button. To translate, move the mouse while holding the middle mouse button, or on MacOS, press Ctrl+Shift and click and drag the window. The displayed area can be zoomed using the mouse wheel.

### A.2 2D view

This shows the 2D 360° image received by the sensor in an interactive viewer (translate by moving the mouse while left mouse button is clicked, zoom using mouse wheel).

### A.3 Spreadsheet view

Displays the same data as the two previous views, but in a spreadsheet that is updated in real time (see figure II.A.3).



Figure II.A.3: Spreadsheet view

In the spreadsheet view you can select the source of data to be displayed (Ouster sensor, GPS/IMU sensor when available, etc.), change the numerical precision, show only selected points, or hide some columns.

Rows may be sorted by column values by clicking the column's header.

#### A.4 ParaView Pipeline

Ouster Studio is built on top of ParaView which uses "Pipeline" to process data. If you want to use the features of ParaView you can go to the "Views" menu and show the "Pipeline Browser" and "Properties" menu (see figure II.A.4.1). Some of the features of ParaView include (x,y,z). These features are complex and most users of Ouster Studio will not need them.

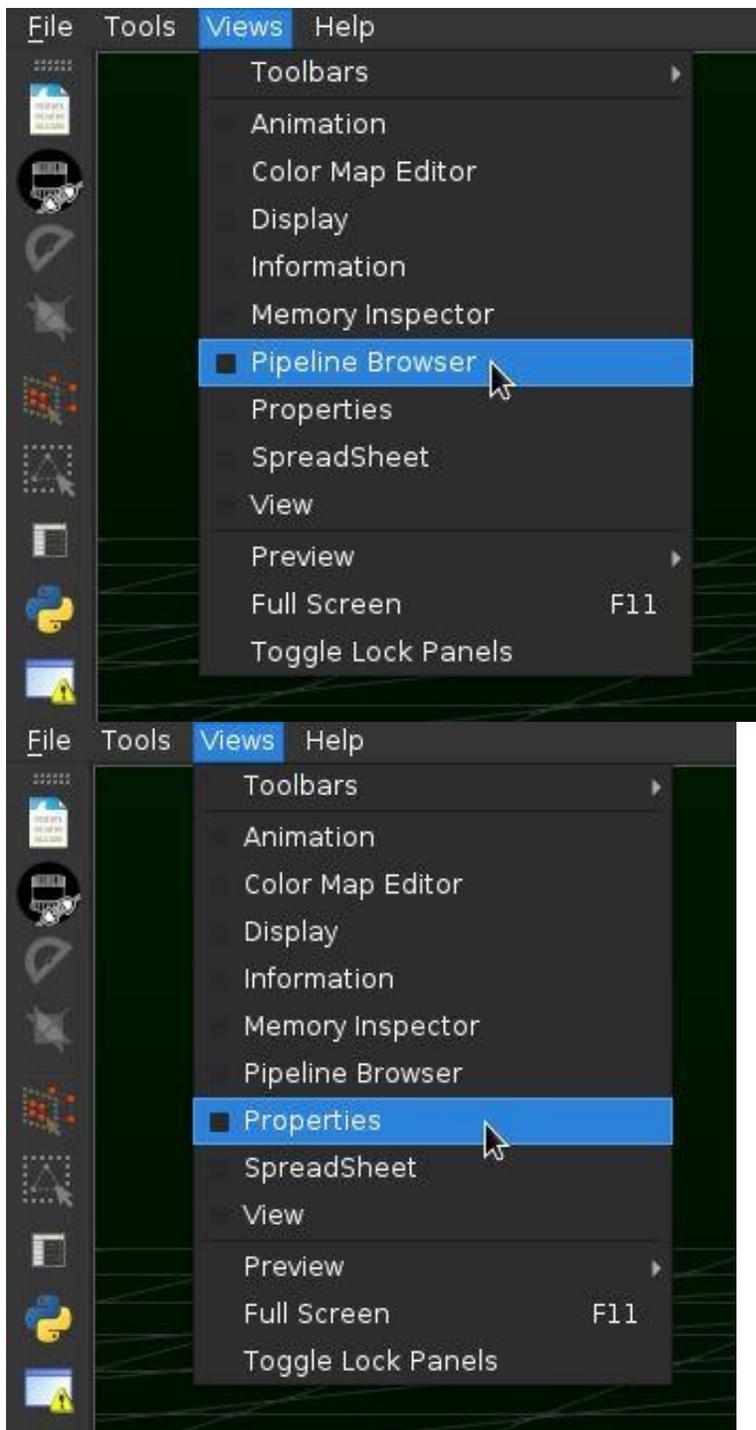


Figure II.A.4.1: Left: enable the “Pipeline Browser”, Right: enable the “Properties” panel

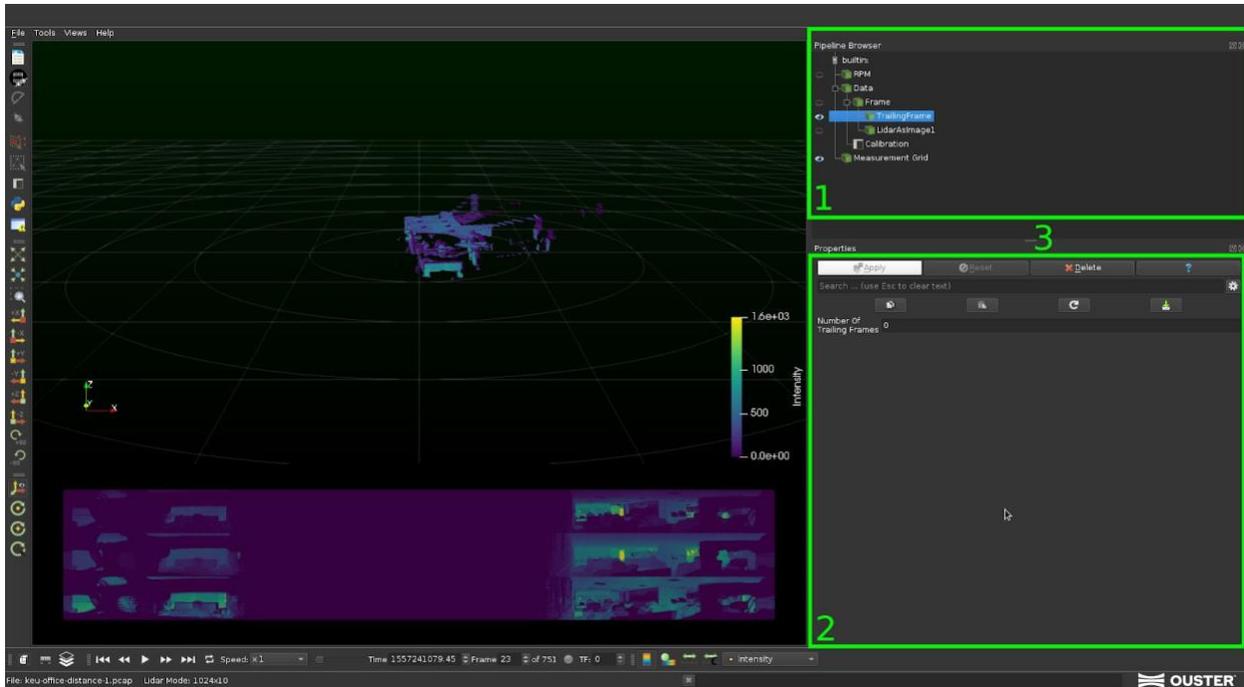


Figure II.A.4.2: 1) “Pipeline Browser” panel, 2) The “Properties” panel, 3) Button to adjust the sharing of vertical space between the two panels. Note that you can close a panel by clicking the cross located in its top right hand corner.

The content of the “Properties” panel depends on what element is selected in the “Pipeline browser”.

## A.5 Advanced “ParaView” user interface

Ouster Studio is a ParaView-based application and thus inherits some features from ParaView. This view is the default view for the application, however many of these features may not be useful for the average user. These features may be disabled by untoggling the “Advanced Features” mode found inside the “Help” menu (see figure II.A.5).

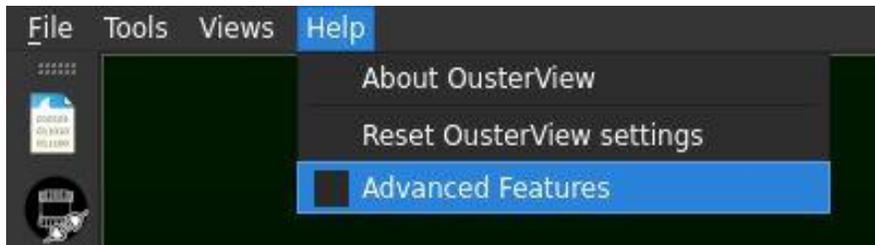


Figure II.A.5: Enable the “Advanced features” mode

Once the advanced features are disabled, several menus and toolbars will disappear. If you would like help using the advanced features, please refer to the ParaView documentation:

<https://www.paraview.org/documentation/>

## B. Live Stream Mode

### B.1 Open the sensor stream and sensor configuration dialog

If a compatible (OS1) Ouster sensor is accessible over the network, you can display the live output of the sensor by selecting the sensor stream button. (see figure II.B.1.1).

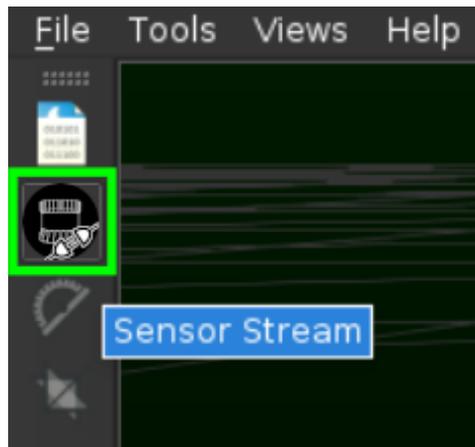
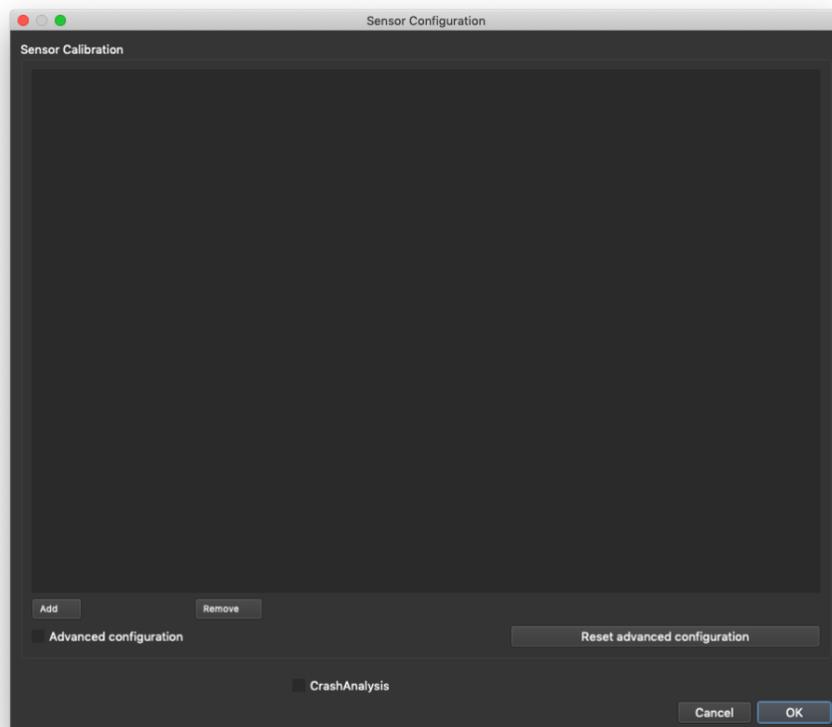


Figure II.B.1.1: sensor stream button

Once the Sensor Stream button has been clicked, a window named Sensor Configuration will open (see Figure II.B.1.2). You can change the network configuration if you click on the advanced configuration checkbox.



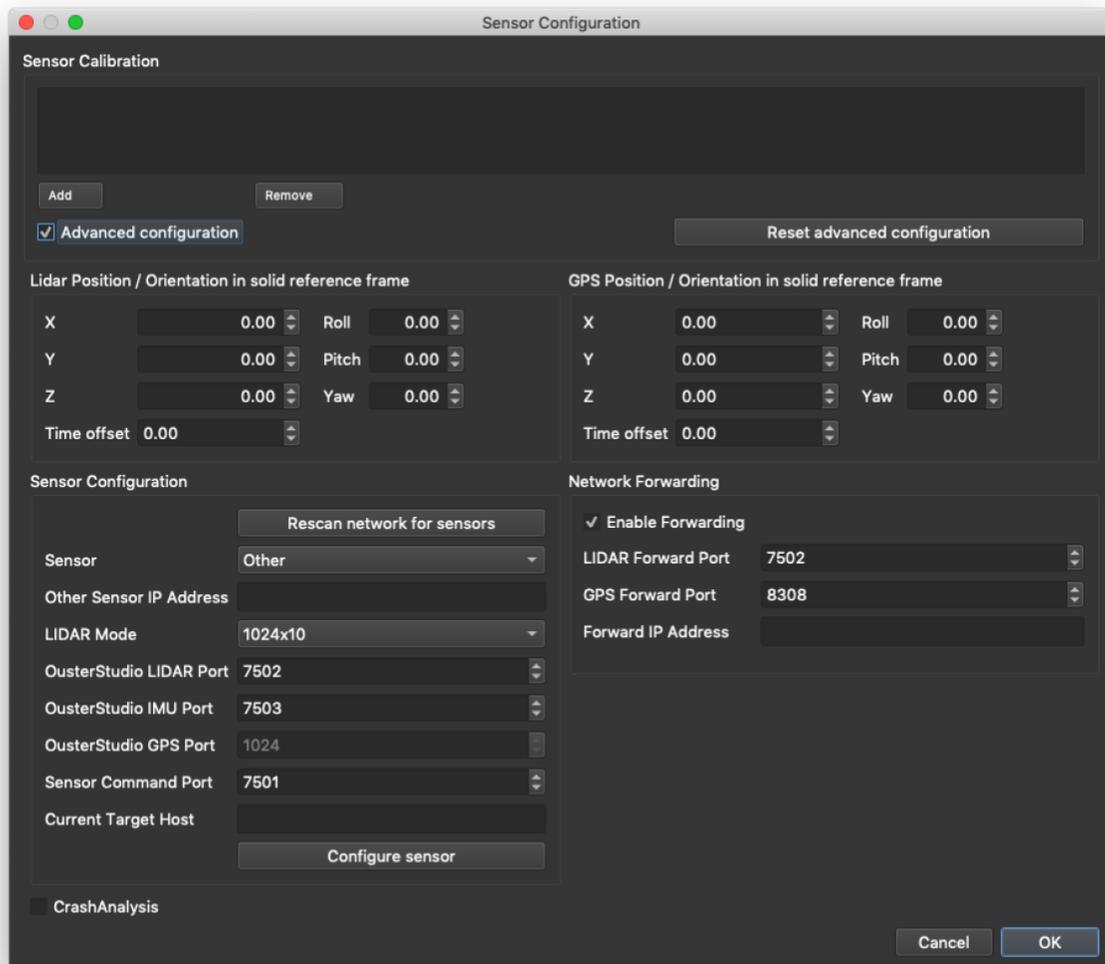


Figure II.B.1.2: Sensor configuration. Top: default option; Bottom: with advanced configuration options

## B.2 Sensor Calibration File

A calibration file is required to correctly interpret sensor data. The calibration data depends on the type of sensor and the operating mode (e.g. the number of firings and the scan frequency). The calibration file contains corrections for the altitude and azimuth angles of the sensor when operating in a specific mode. We provide a sample calibration file for all the sensor used to record our sample .pcap file in 1024x10 mode, but each sensor requires unique measurement corrections for the best accuracy. To add other sensor calibration files, simply click on the "add" button and select the desired file.

Remarks:

- If you use the wrong calibration file, no error message will be shown; however, the calculated point positions will be incorrect and most likely unrecognizable.

- The calibration file only affects the visualization of data within Ouster Studio. When recording a stream, the raw packet data is saved to the .pcap file directly. This data can be replayed later with the correct calibration file even if it was initially recorded with an incorrect calibration file.
- Calibration files can also be retrieved from connected Ouster sensors. Please see the “Live Sensor Configuration” section below for more details.

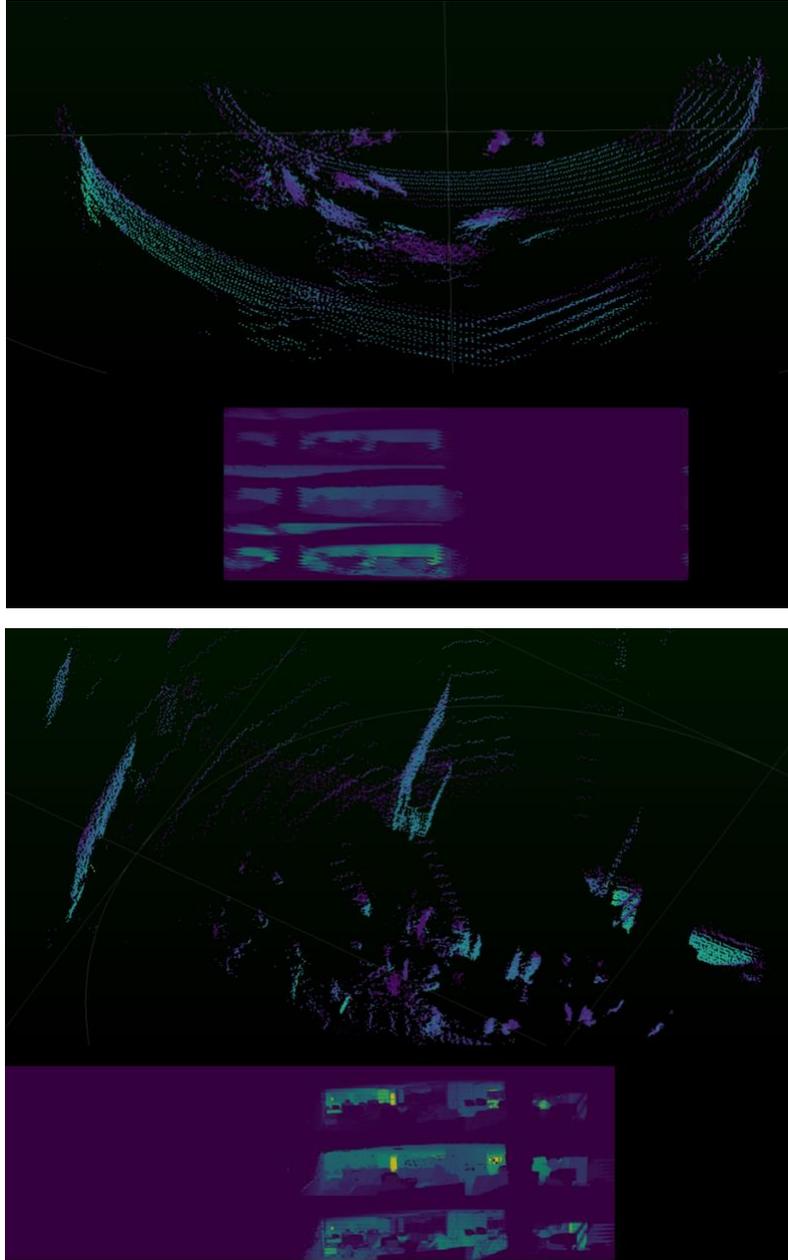
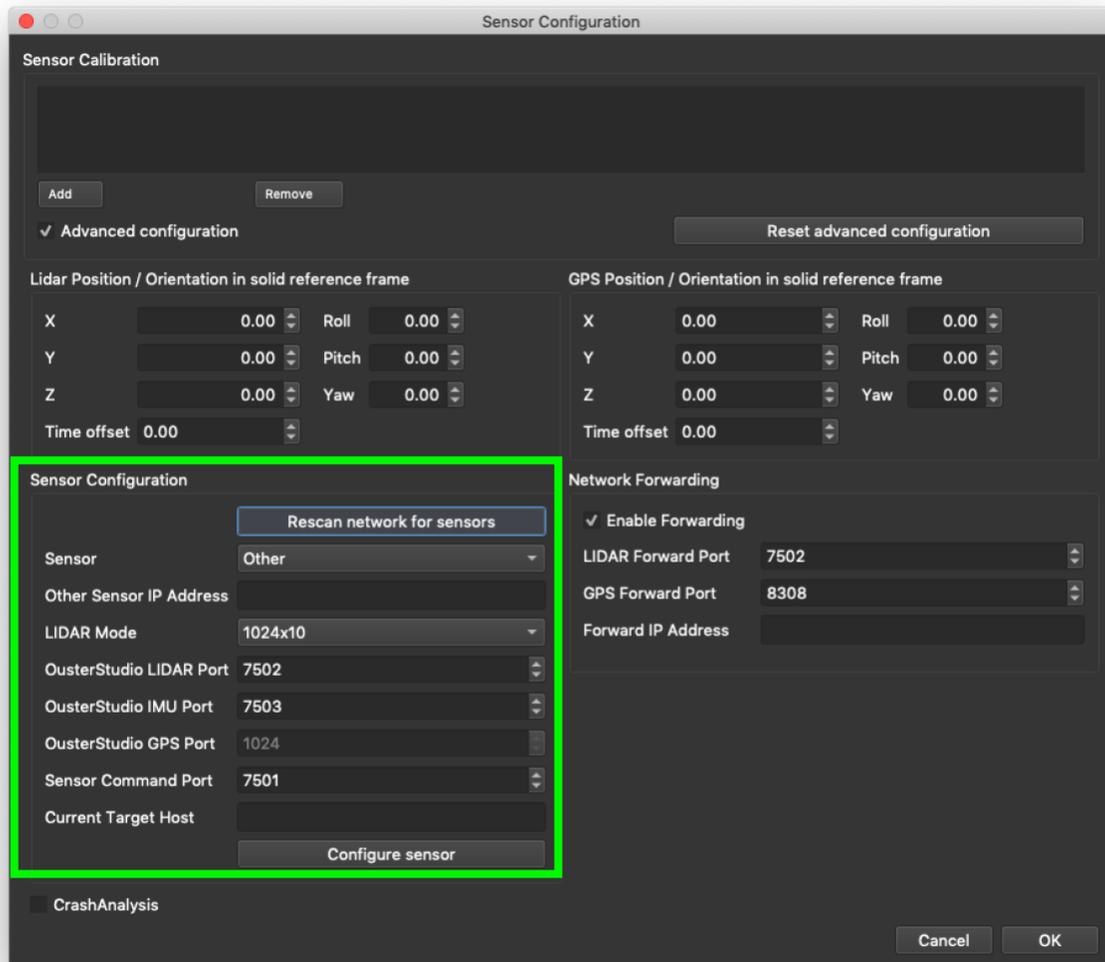


Figure II.B.1.3: Top: Distorted point cloud due to wrong configuration file (512x10), Bottom: Same recording of an office with correct configuration file (1024x10).

### B.3 Live Sensor Configuration



The “Sensor Configuration” section is used to configure Ouster Studio’s listening ports for live-stream capture. If the sensor is already streaming live data, simply set the Ouster Studio mode and lidar and IMU ports to the current values and then press OK to begin visualizing the live stream.

The sensor configuration section can also be used to update the sensor’s internal configuration if the sensor is directly accessible over the network (ex. via LAN or a link-local connection). Compatible Ouster lidar sensors should be detected by clicking on the “Rescan network for sensors” button. If the scan is successful, the combobox will be populated with the detected sensors and their IP addresses. A single sensor may appear multiple times in the list if there are multiple routes to it. For example, this may occur if the computer is connected to the same network via a cable and a wireless connection. In this case, it is up to the user to select the optimal network route for streaming.

If the scan fails, select “Other” in the combobox and manually enter the IP address of the sensor on the following line labeled “Other Sensor IP Address.”

Once a sensor has been chosen, make sure that the correct command port is entered in the “Sensor Command Port” field (7501 by default). Clicking the “Sensor” combobox should then retrieve the current configuration from the sensor and update the mode, Lidar and IMU ports, and “Current Target Host” fields. The mode and ports can be changed by updating the field values and then clicking the “Configure sensor” button, which will send the settings to the sensors. The sensor’s streaming destination will also be updated to stream to Ouster Studio’s host.

When the sensor is updated, it will return a new calibration file that the user will be prompted to save. This file will then be added and automatically selected from the list of calibration files at the top of the dialog. The proposed name will include the mode of the sensor. The user can safely overwrite previous files for the same mode from the same sensor as these should be identical. Nevertheless, double-check to be sure, since a correct calibration file will be required to replay recorded stream data.

### Network Forwarding:

You have the possibility to forward the lidar data to another computer by specifying the IP and port. The lidar (respectively GPS) forward port can be the same as the original lidar port (respectively GPS port). It is possible to repeat the forwarding on the other computer, and so on. The only difference inside Ouster Studio between unforwarded and forwarded usage is that when data received is coming from a forwarding Ouster Studio (and not directly the lidar sensor), it is not possible to ask the lidar for new messages because the computer is unable to send messages to the lidar. This is because the lidar’s streaming target is automatically updated to the computer from which it receives commands.

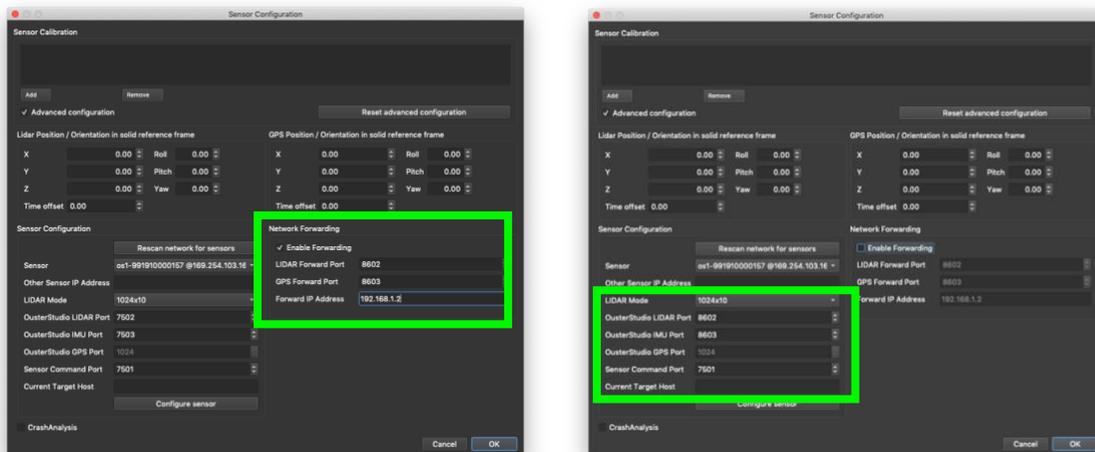


Figure II.B.1.4:

Left: First instance of Ouster Studio which receives on its ports [7502; 7503] and then forwards the received data to the IP address 192.168.1.2 on the ports [8602; 8603].

Right: Second instance of Ouster Studio running on another computer, with IP address 192.168.1.2, which receives the forwarded data on the ports [8602; 8603], and does not itself forward the received data.

Once the configuration is done click “OK” and the streaming data should appear in Ouster Studio.

## B.4 Visualization & Interaction

You can record a live stream into a .pcap file by clicking the record button (see figure II.B.2) and choosing a filename and location on the computer.



Figure II.B.4: the record button, located at the bottom center of the interface

## C. Playback Mode (.pcap files)

If you have a .pcap file containing Ouster lidar sensor data (recorded using Ouster Studio) it is possible to display the data in Ouster Studio by using the “Playback Mode” button (see figure II.C.1).

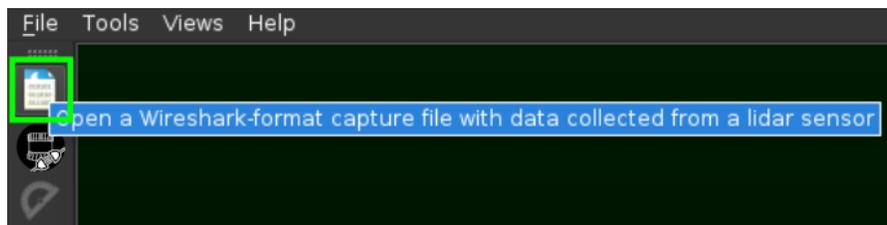


Figure II.C.1: playback mode action

Once the Playback Mode button has been clicked, a configuration window named Sensor Configuration will open. For more information about the Sensor Configuration dialog and how to set it, please refer to section B - 1 Open sensor stream and sensor configuration dialog.

## D. Ouster Studio Toolbars

Once you are visualizing data in Ouster Studio, different tools are available to analyze, enhance, and understand data.

### D.1 Basic Controls Toolbar

The basic controls toolbar is located at the top left of the Ouster Studio interface.



Figure II.D.1: Basic controls. From top to bottom: open .pcap, sensor stream, Choose calibration file, Crop returns, Select all points, Select dual returns, Spreadsheet view, python console, Error console



#### Open .pcap:

Open a .pcap file containing recorded Ouster sensor data



#### Sensor stream:

Open a UDP socket to receive data from connected sensor in real-time.



#### Change the calibration:

Open the calibration dialog to change the calibration parameters (calibration file, GPS parameters, and network parameters). Note that the “Open .pcap” and “Sensor stream” button will automatically open the calibration dialog before displaying the point cloud. You can make adjustments if you did something wrong during calibration with this button.

#### Crop returns:

This feature enables you to crop part of the point cloud to keep only the captured points that belong to a “cropping volume”. (ex. The lidar is on a car and you do not want to display points belonging to the car). The cropping volume can be defined using cartesian (x, y and z) or spherical (azimuth, vertical angle and radius) coordinates.



There are two ways to enable cropping of returns: either by clicking on  or by using the “Pipeline Browser”. Using the button is the recommended approach to cropping.



Be careful if you use  as the settings can be remembered between two executions of Ouster Studio, and could cause all points to be cropped out, resulting in no points being displayed.

Please refer to section A.4 for instructions to open the “Pipeline Browser” panel and the “Properties” panel. Once the two panels are opened and a .pcap is opened, select the “Frame” source in the Pipeline Browser then toggle the advanced properties, as shown in figure II.D.2.

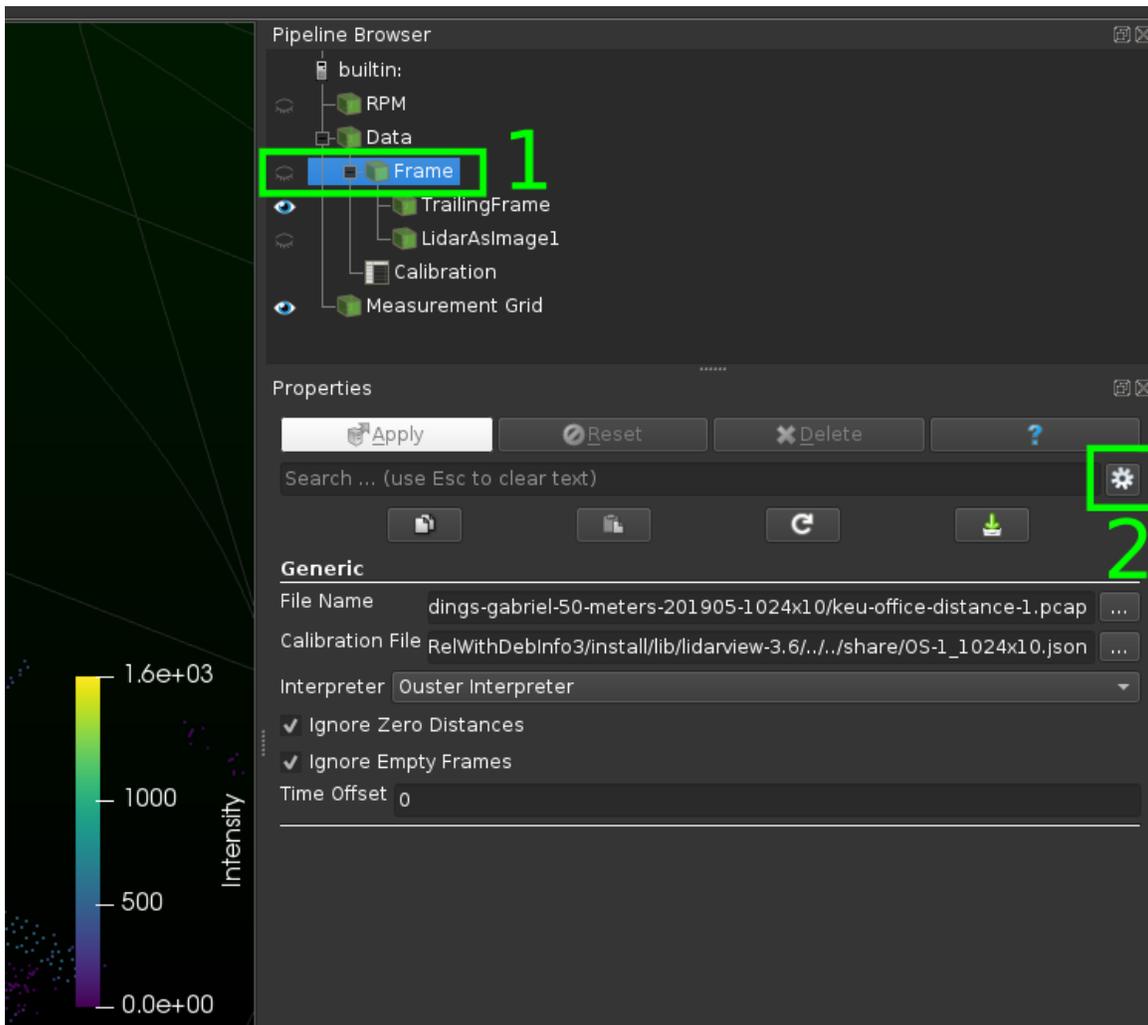


Figure II.D.2: 1) first select the “Frame” source then 2) toggle the advanced properties. You should then have a visual similar to the one in Figure II.D.3

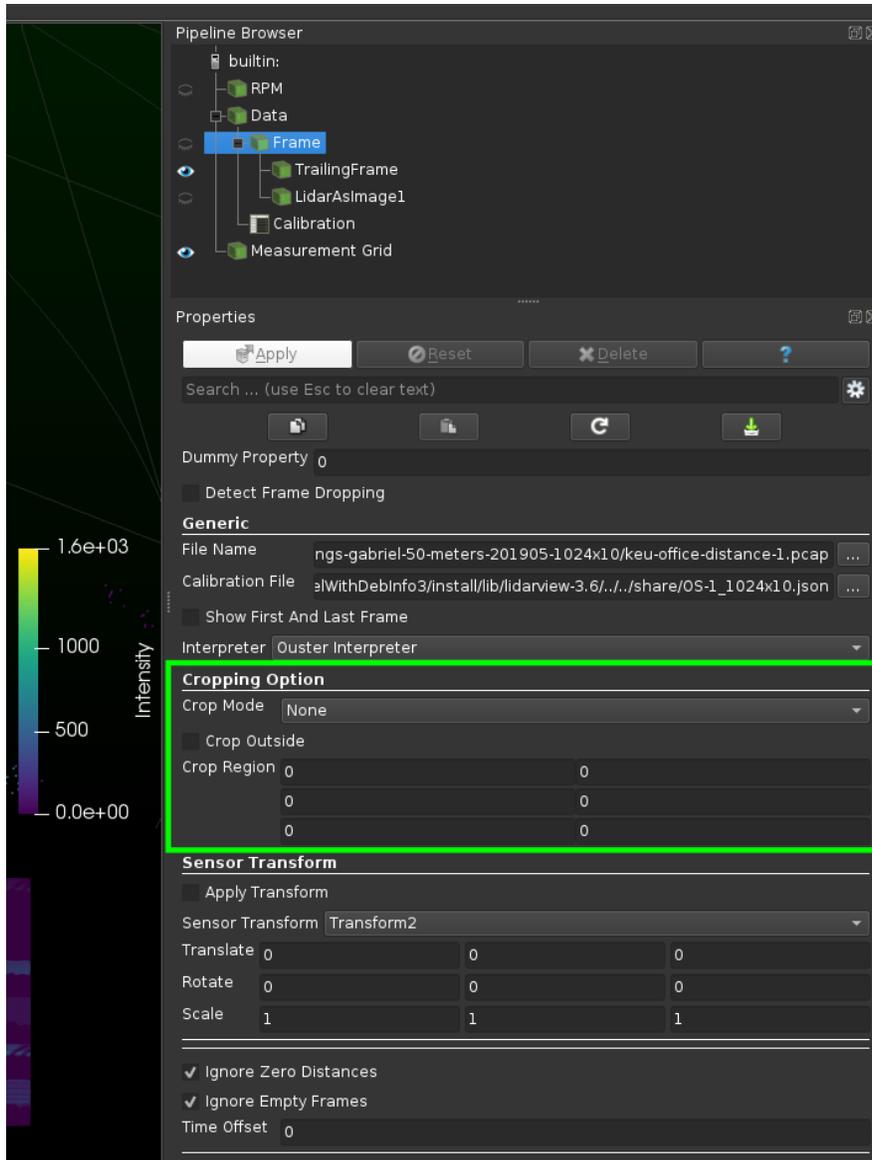
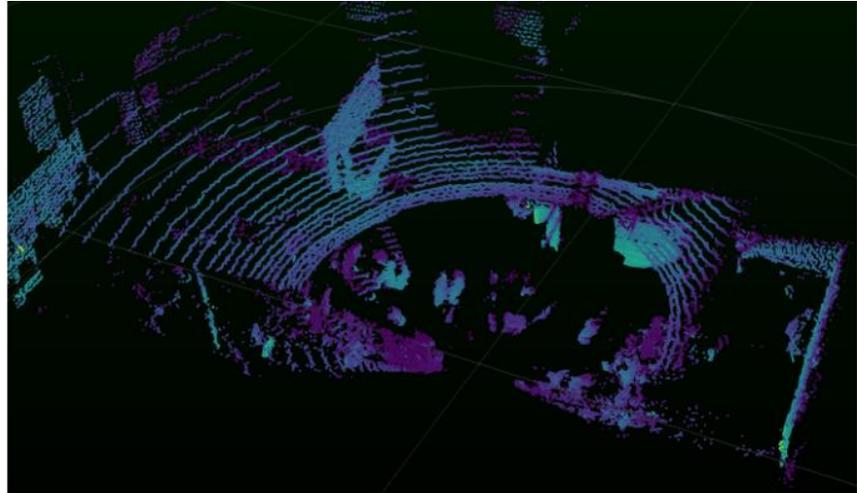


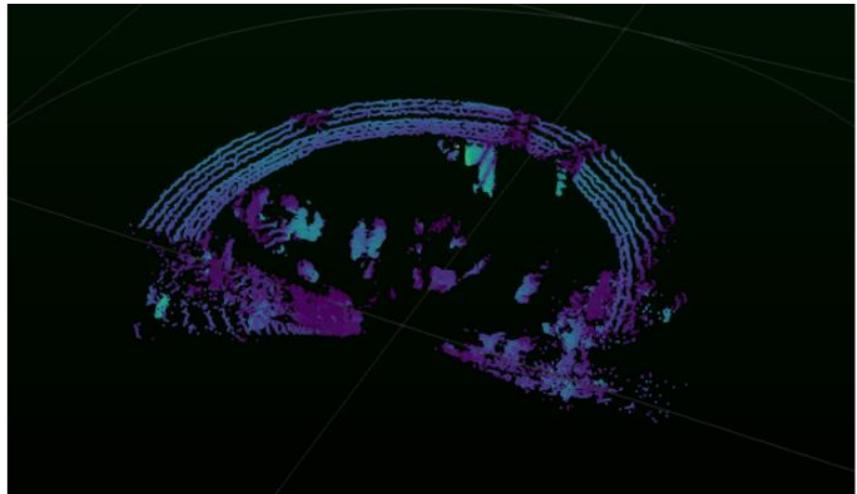
Figure II.D.3: The property panel in “advanced mode” with cropping options.

See the figure II.D.4 for an example of the “Spherical” cropping mode. When the mouse pointer hovers on the input fields, tooltips are shown to remind you of what kind of values are expected and their units.

Cropping Option	
Crop Mode	None
Crop Outside	
Crop Region	0
	0
	0



Cropping Option	
Crop Mode	Spherical
Crop Outside	
Crop Region	0
	360
	-90
	90
	0
	5



Cropping Option	
Crop Mode	Spherical
<input checked="" type="checkbox"/> Crop Outside	
Crop Region	0
	360
	-90
	90
	0
	5

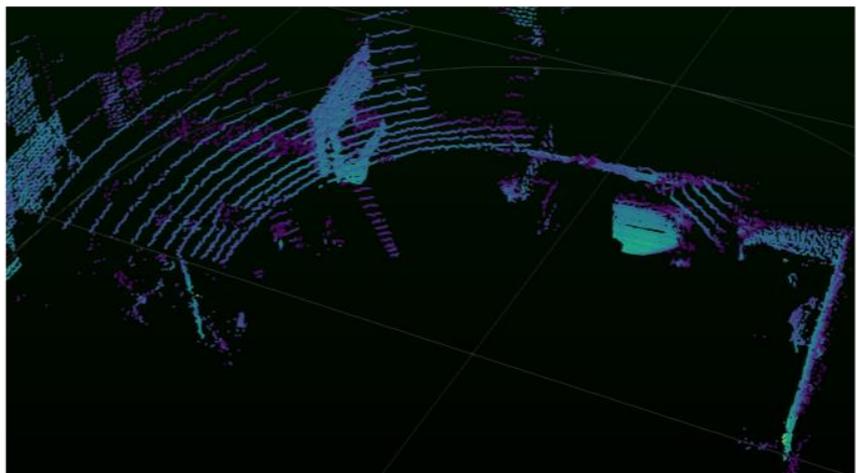


Figure II.D.4: Behavior of the cropping in mode "Spherical". First example: no cropping, second example: we crop a sphere of radius 5 meters, third example: we "crop outside" a sphere of radius 5 meters. The third example corresponds to the most common usage (when you want to hide points belonging to the Lidar physical support - in general 1 to 2 meters are sufficient).



#### Select points:

Select points in a defined region. The region is defined with the mouse. Once drawn, the defined zone and viewpoint is saved, and all the 3D points falling in this zone from this viewpoint will be selected. This will work even if you move the camera around or change the time-step. To deselect points, click the button a second time and click once anywhere on the point cloud screen.



#### Spreadsheet view:

Open the spreadsheet view.



#### Python console:

Open the python console (intended for developers).



#### Error console:

Open the error console.

## D.2 Display controls toolbar

You can affect the way the data is displayed in Ouster Studio using the Playback Control toolbar (see figure II.D.2).

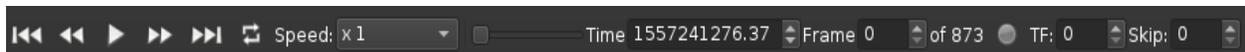


Figure II.D.2: Playback Control toolbar. From left to right: Go to start, Go to previous frame, Play / Pause, Go to next frame, Go to end, Loop, Playing speed, Frame selection slider, Frame indicator, Record button (disabled in replay), Trailing frame, Skip



#### Go to start:

Go to the first frame of the .pcap file.



#### Go previous frame:

Go to the previous frame of the .pcap file.



#### Start / Stop:

Start or stop the playback of the .pcap file.



#### Go to next frame:

Go to the next frame of the .pcap file.



Go to end:

Go to the last frame of the .pcap file.



Loop:

Enable or disable the loop mode. Loop mode replays the playback once the last frame is reached.



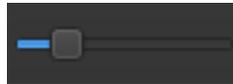
Speed:

Control the speed of the playback. The numbers indicate the playback speed relative to real time. The “All frame” option asks Ouster Studio to show all frames as fast as possible whereas in other modes Ouster Studio is allowed to drop frames in order to reach the required speed.



Trailing frame:

Display the last X frames at the same time. It is useful when you have point clouds expressed in a fixed referential (with GPS/IMU or slam data for example).



Select frame slider:

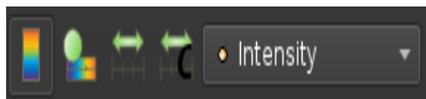
Indicates the current frame. You can move to the frame you want using the slider or the spinbox.

### D.3 Color controls toolbar

The color control tool (see figure II.D.3.1) is useful to display the information attached to each point. You can choose the information you want to use for color-coding and choose the exact color map or range of colored data.



Figure II.D.3.1: from left to right: Color legend, Edit color map, Fit color range to data, Custom color range, color-coding data selection



Data selection:

Changes the data information used for color coding. By default, the intensity of each Lidar Return is displayed. However, you can select other information such as X, Y or Z coordinates, timestamp, laser id... (see Figure II.D.3.2).

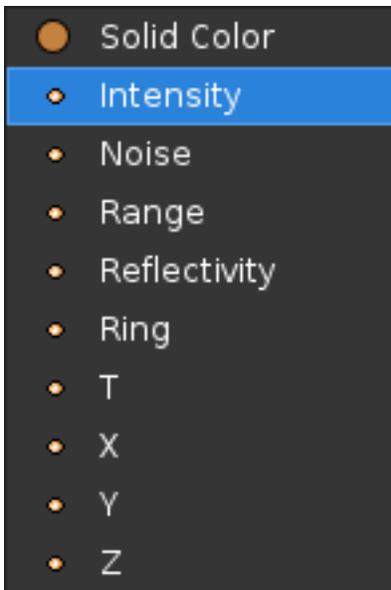


Figure II.D.3.2: available data information for color-coding



[Color legend:](#)

Display the color legend within the Main 3D View (see figure II.D.3.1).

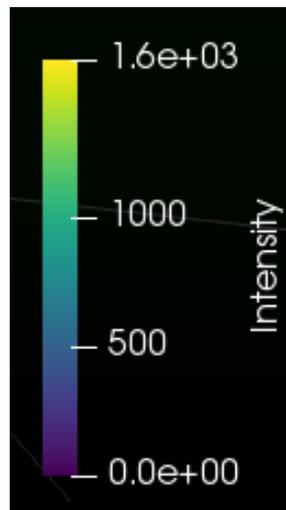


Figure II.D.3.3: color legend



[Edit color map:](#)

Edit and change the color mapping of the data.

When clicked, this button opens a new panel (see figure II.D.3.4)

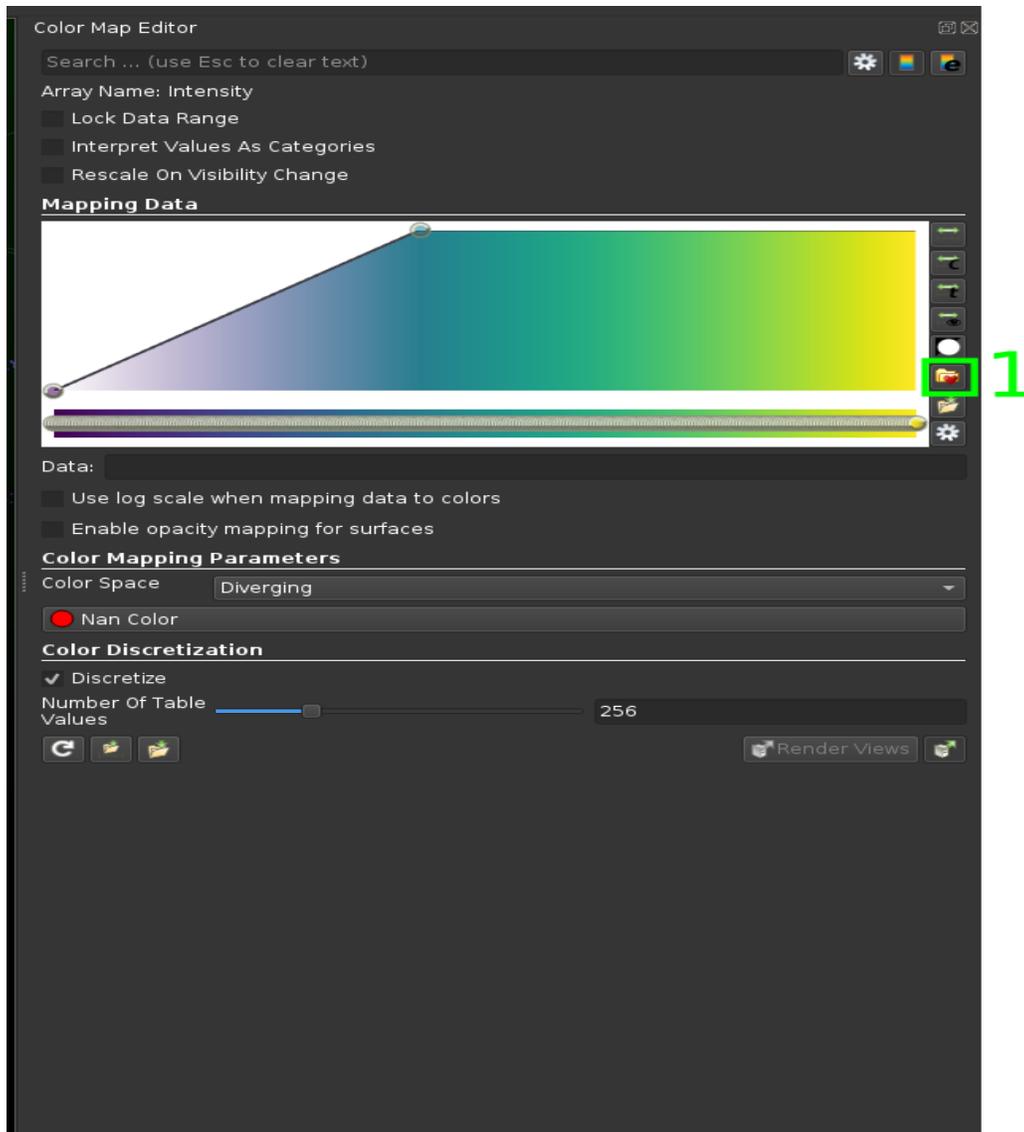


Figure II.D.3.4: The color map editor 1) clicking this button allows changing the colormap (see Figure II.D.3.5)

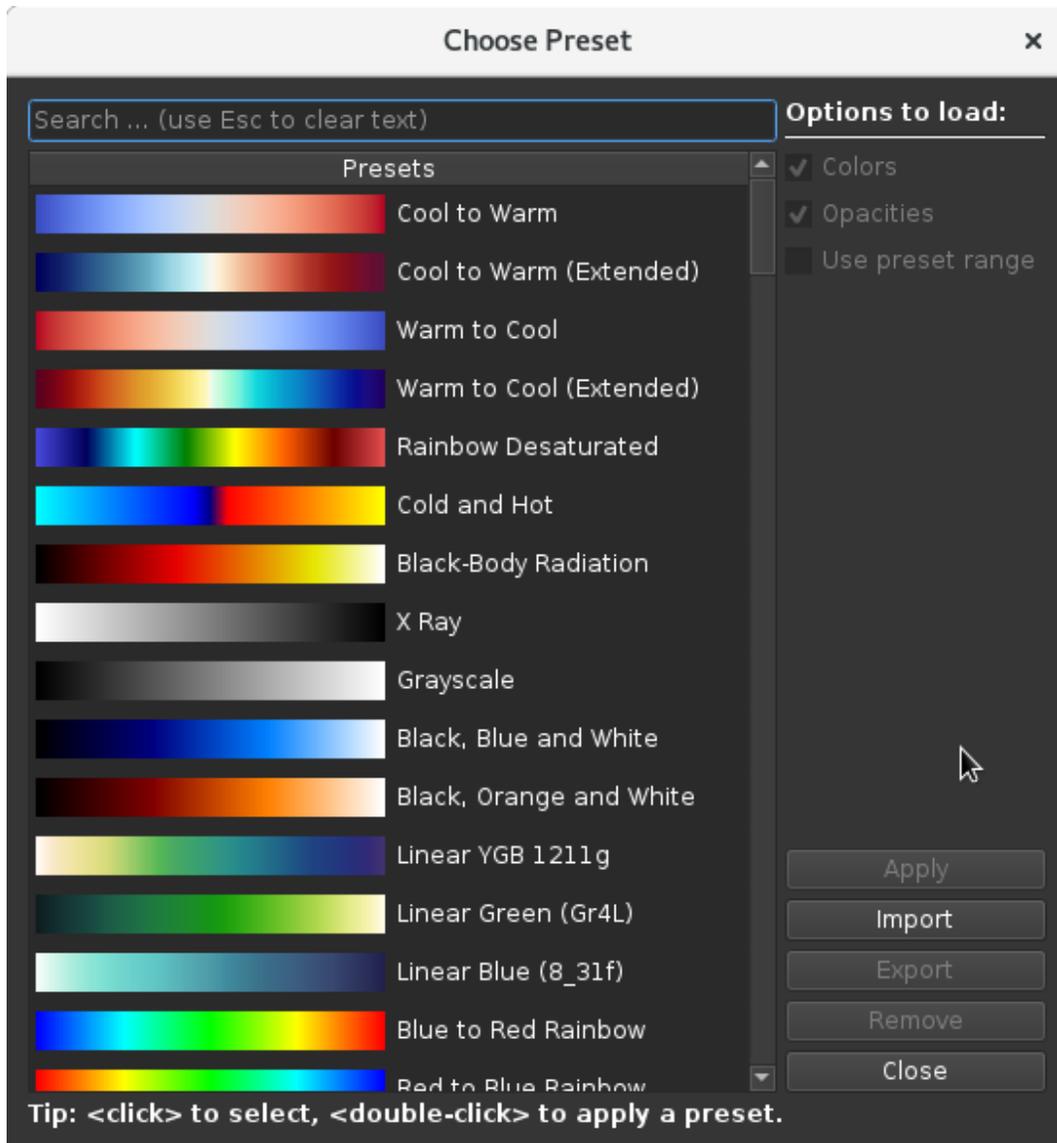


Figure II.D.3.5: A look at the available colormaps. Select on and click “Apply” to start using it.

Please take note that it is not currently possible to change the colormap used by the 2D 360° Image view shown in the main view.

Ouster Studio usage is similar to ParaView usage (especially if “Pipeline browser” and “Property” panels are enabled as well as “Advanced mode”, see sections II.A.4 and II.A.5) so we advise you to review the ParaView documentation (<https://www.paraview.org/documentation/>), especially the ParaView guide, section “10.2. Editing the transfer functions in ParaView”



**Rescale data:**

Rescale the color map so that the color extremities correspond to the data value extremities



#### Rescale custom:

Rescale the color map as you want. It is useful if you have data with a big range value. The major part of the information is located in a subrange values.

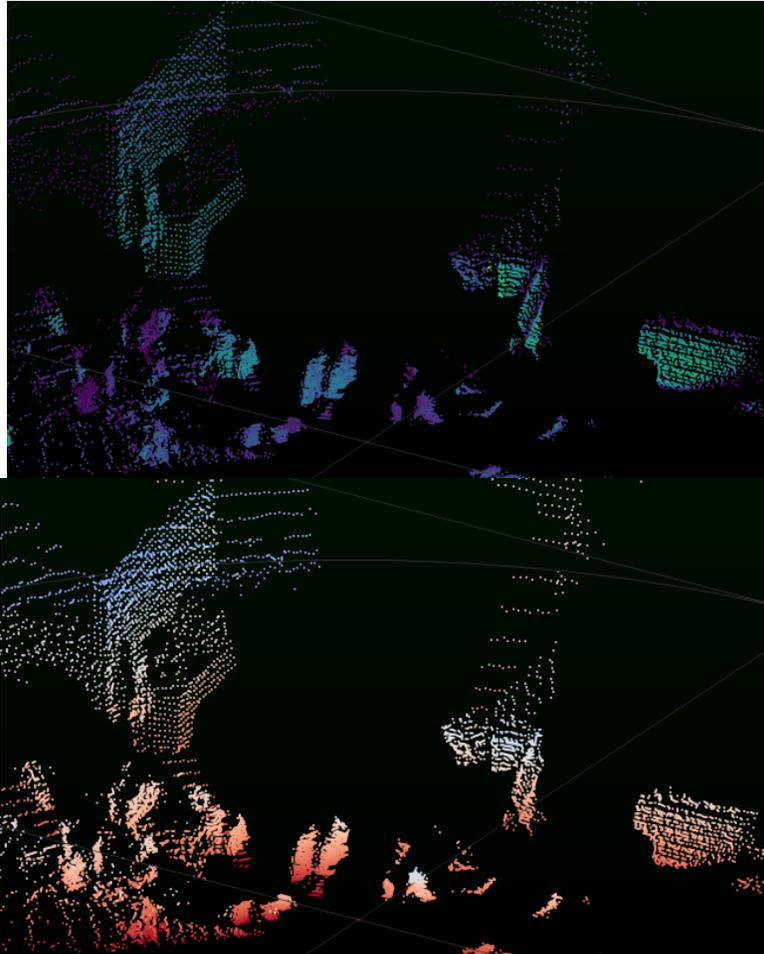


Figure II.D.3.6: same point cloud with coloring based on different attributes. Left: Intensity, Right: Ring

### D.4 View Controls toolbar

The view control (see figure II.D.4) contains tools to control the camera.

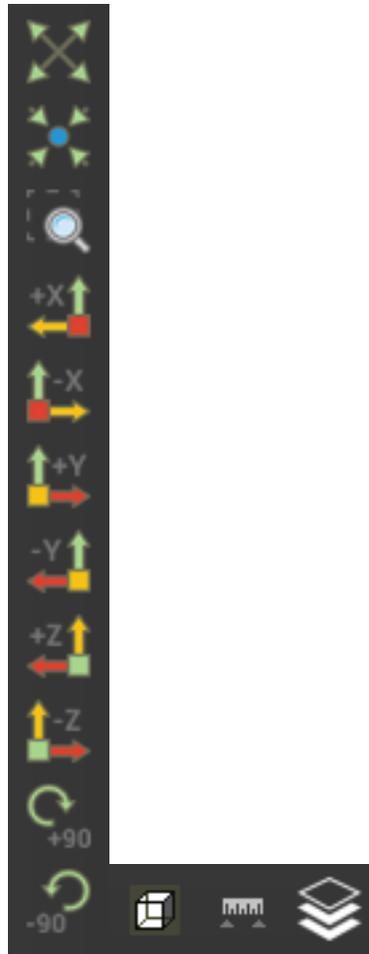


Figure II.D.4: Right: Camera control toolbar. From top to bottom: reset camera, zoom to data, zoom to selection, set view direction to +x, -x, +y, -z, +z, -y, rotate camera. Left: Orthogonal/Projective view, measure distances, plane fit selection,



Reset Camera:

Reset the camera at its initial position and orientation.



Orthogonal view:

Switch from a perspective to orthogonal view.



Measure distances:

While pressing the <ctrl> key, press and hold the left mouse button to start drawing a line inside the 3D view in order to measure its distance. Release the mouse button when the mouse is located at the desired position. This functionality only works when the orthogonal view is activated (see item above).



Plane fit tool:

Compute the plane which fits the currently selected points. The feature is not available yet.



Set view direction to +x, -x, +y, -y, +z, -z:

Set the camera viewing direction to the x, y or z axis.

## E. Data Export

### E.1 Save Current Frame as CSV

If no trailing frames are enabled () , it is possible to save the current frame of a .PCAP by using the keyboard shortcut <ctrl>+s when Ouster Studio is in focus. Alternatively, display the PipeLine Browser as described in section II.A-4, select the “Frame” node, and press <ctrl>+s to save the data. Select “CSV File(\*.csv)” in the filetype dropdown menu (see figure II.E.1) and an output path then press OK.

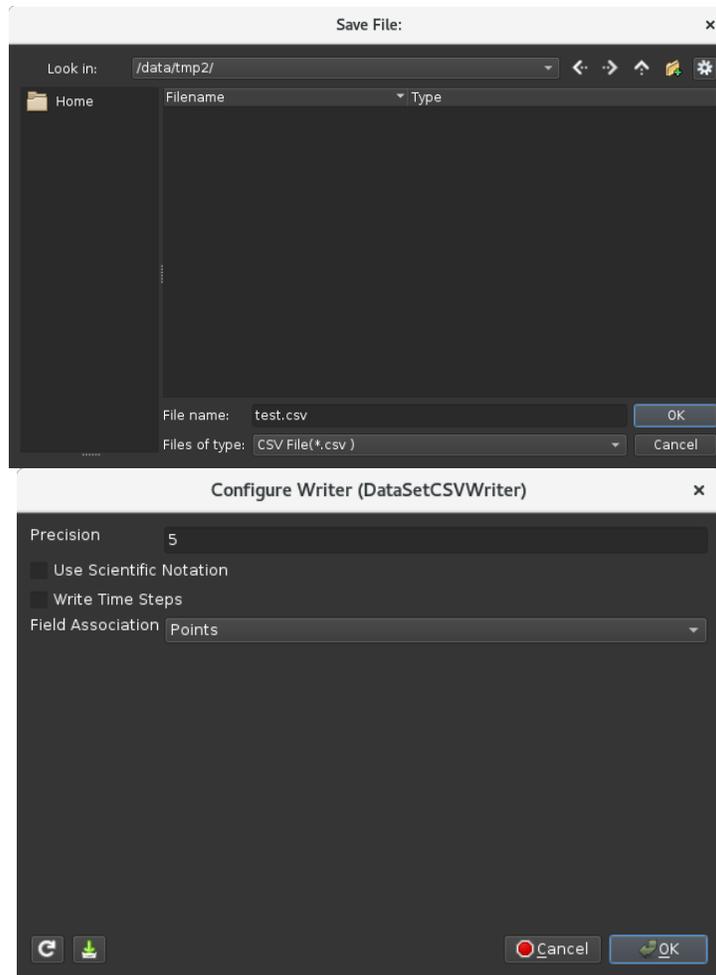


Figure II.E.1: Top: first window allows choosing a name and a location for the resulting CSV,

Bottom: second window allows changing the precision. Field Association should be "Points". If "Write Time Steps" is selected, all frames will be written in CSV names: test0.<frame number>.csv. Be careful as writing all frames as a CSV takes about 3 times more disk space than the PCAP from which you are reading.

## E.2 Save Current 2D 360° View as Image

Enable the pipeline browser (see section II.A.4) and select the LidarAsImage node (figure II.E.2). Press <ctrl> + s (<alt> + s on Mac) to bring up the file selection dialog. Choose an output path and name, select PNG, TIFF or JPEG from the dropdown menu, and click OK to export the image.

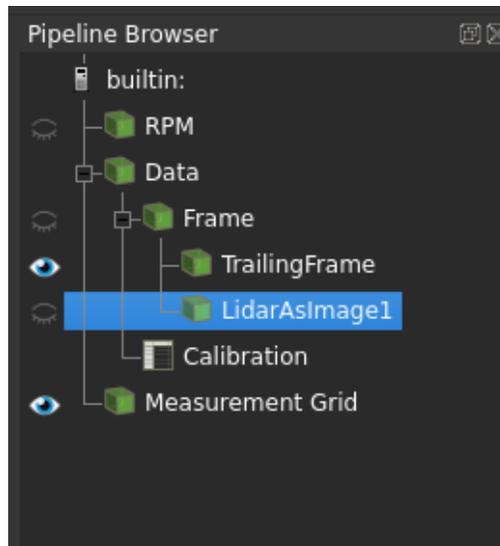


Figure II.E.2: The LidarAsImage node in the Pipeline Browser.

## F. ParaView Pipeline Examples

### F.1 Example of a Filter: Threshold (Select Values In A Range)

Because Ouster Studio is built on top of ParaView, it is possible to use some of ParaView's visualization features. The first step is to open the "Pipeline browser" and "Properties" panel: (please refer to section I.A.4 for instructions). Then when a PCAP is open, it is possible to plug a filter after the "Frame" source in the Pipeline. To do that, select the "Frame" source inside the "Pipeline browser" then hit the keyboard shortcut `<ctrl> + space` (`<alt> + space` on Mac). This will open a window letting you type the name of a filter to instantiate. An alternative if Advanced mode is enabled (as shown in section II.A.5) is to go to the menu "Filter"

In this example we will show how to instantiate a "Threshold" filter to restrict the display to points with an intensity value within a given range.

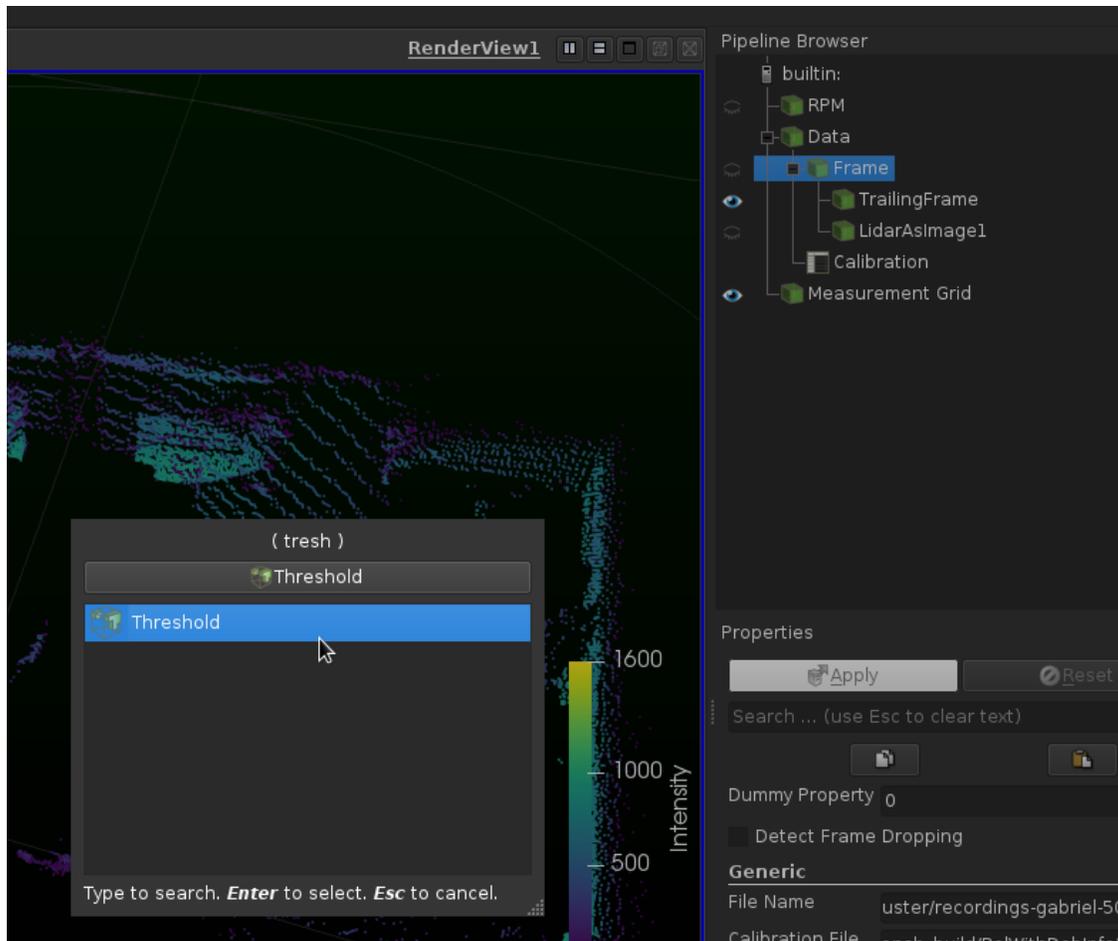


Figure II.F.1: while “Frame” is selected in the pipeline browser (blue background behind the word “Frame”), hit <ctrl> + space (<alt> + space on Mac), then type “thresh”, then press enter. >

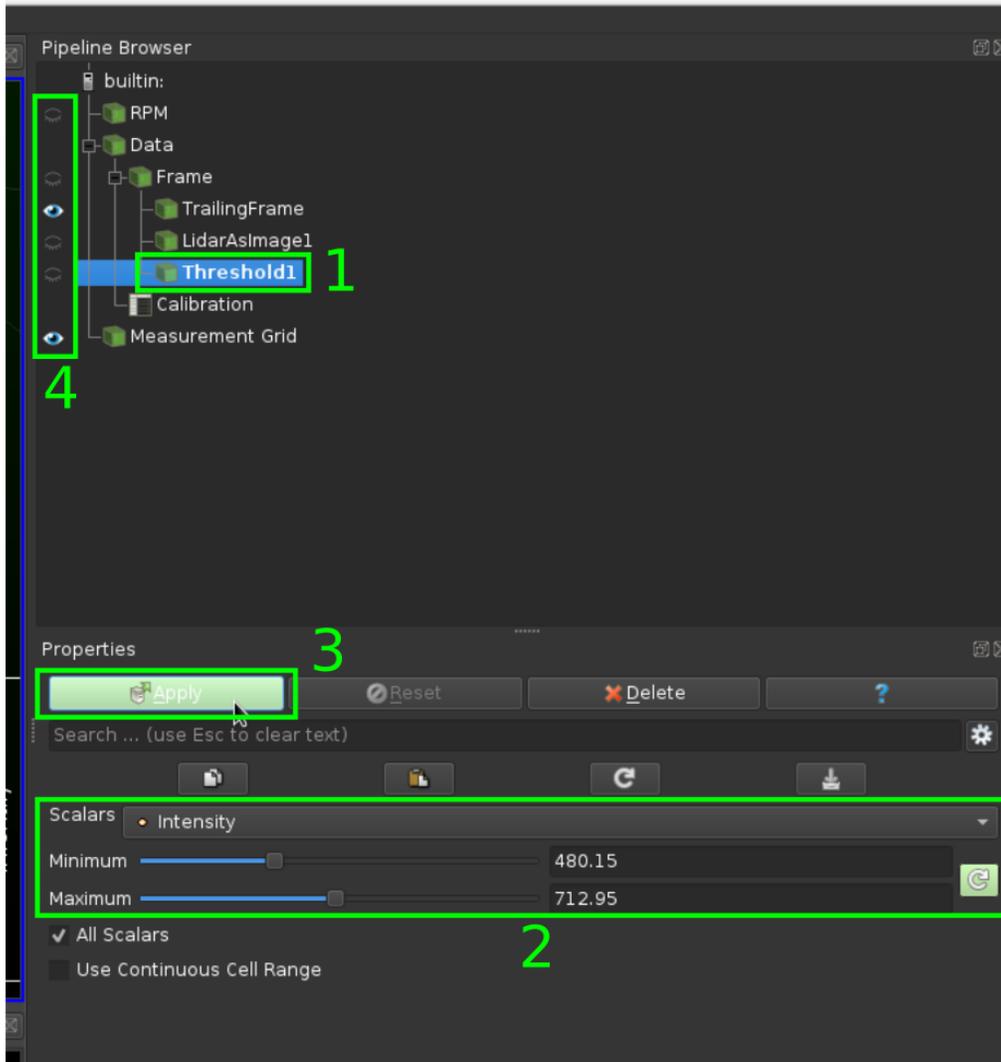


Figure II.F.2: 1) select the “Threshold” filter to show its properties in the properties panel 2) Choose an array to select from and a range of values to keep. 3) Apply these parameters 4) click the eyes so that among the sources that produce 3D points, only the “Threshold” filter is visible. When this is done you should have an output resembling Figure II.F.3

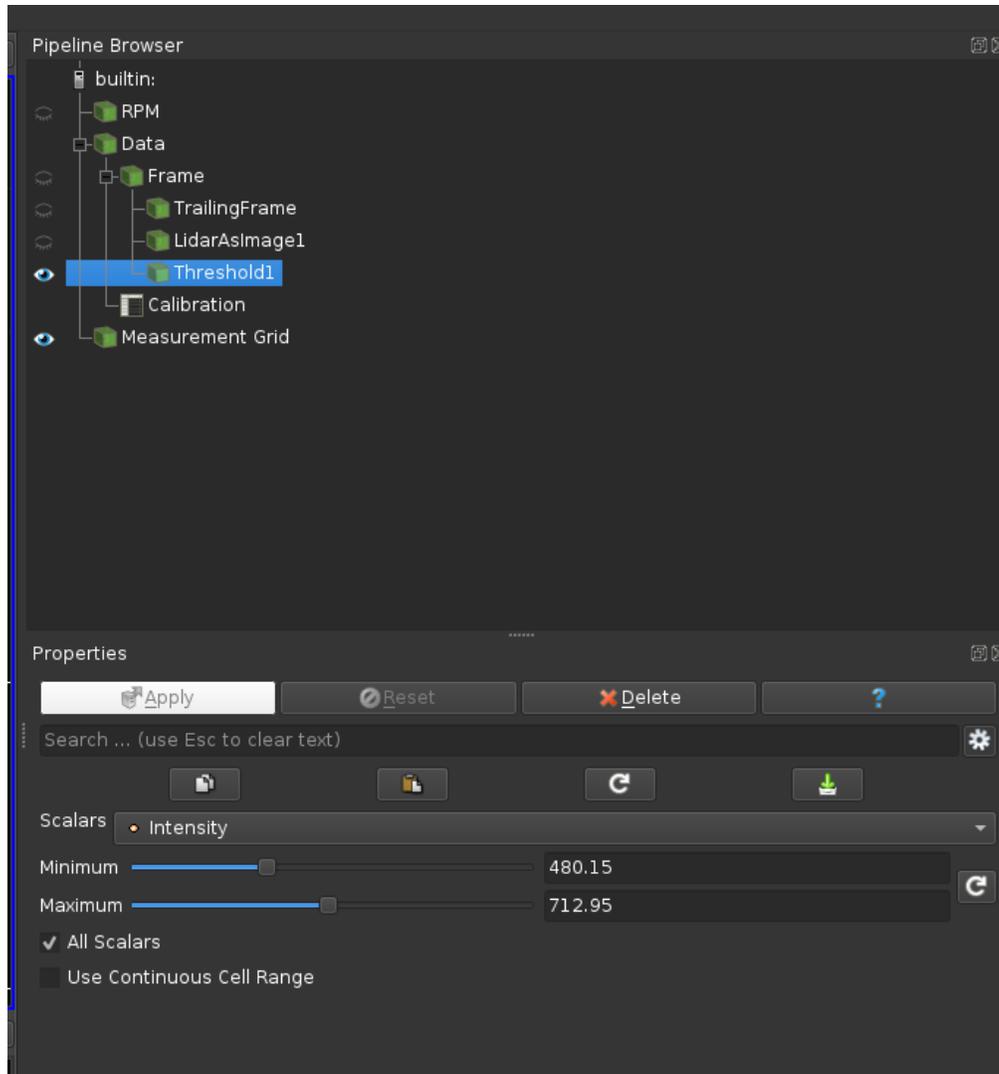


Figure II.F.3: Resulting pipeline when the “Threshold” filter was validated (i.e. the “Apply” button clicked) and it is the only pipeline element producing points that is shown (the blue eye is visible on “Threshold” but not on “Frame” or “Trailing frame”). The last step is to apply a coloring if needs be (by using the color map panel shown in Figure I.A.1).

It is possible to delete the “Threshold” filter by clicking the “Delete button” in the “Properties” panel when the “Threshold” filter is selected in the “Pipeline browser”.

## F.2 Example of a Filter: Pass Arrays (Select Fields to Export to CSV)

The Pass Arrays filter can be used to remove fields from the visualized data and reduce the size of saved data files. The procedure to insert a Pass Arrays filter is the same as for the threshold filter described in the previous section. Go to the pipeline browser and select the “Frame” node (figure II.F.1) again, then press <ctrl> + space (<alt> + space on Mac) to bring up the filter search dialog. Type “pass” and the Pass Array filter will appear in the list. Select it and press Enter to continue.

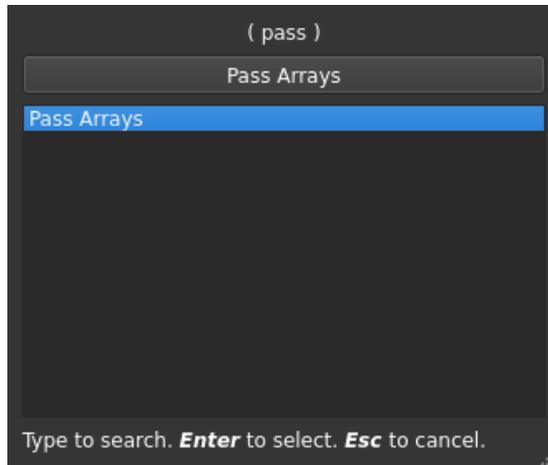


Figure II.F.4: Selecting the Pass Arrays filter from the filter selection dialog.

Once the “Pass Arrays” filter has been added to the pipeline, select it to display its properties in the properties section (figure II.F.5). Uncheck fields in the list of sensor data to remove them from the output and then click on Apply. Once the filter has been applied, make sure that it is still selected in the pipeline browser and then press <ctrl>+s to save the filtered data. Select the CSV output format and a file path, then click OK.

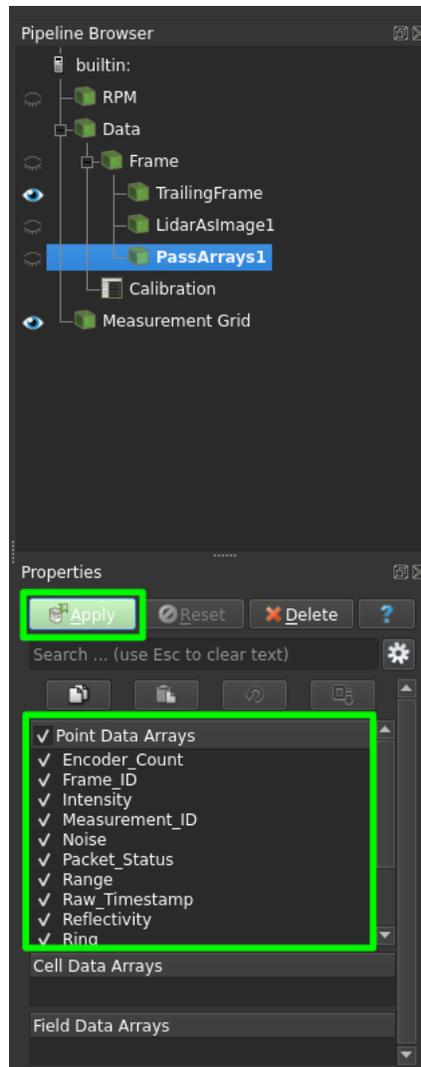
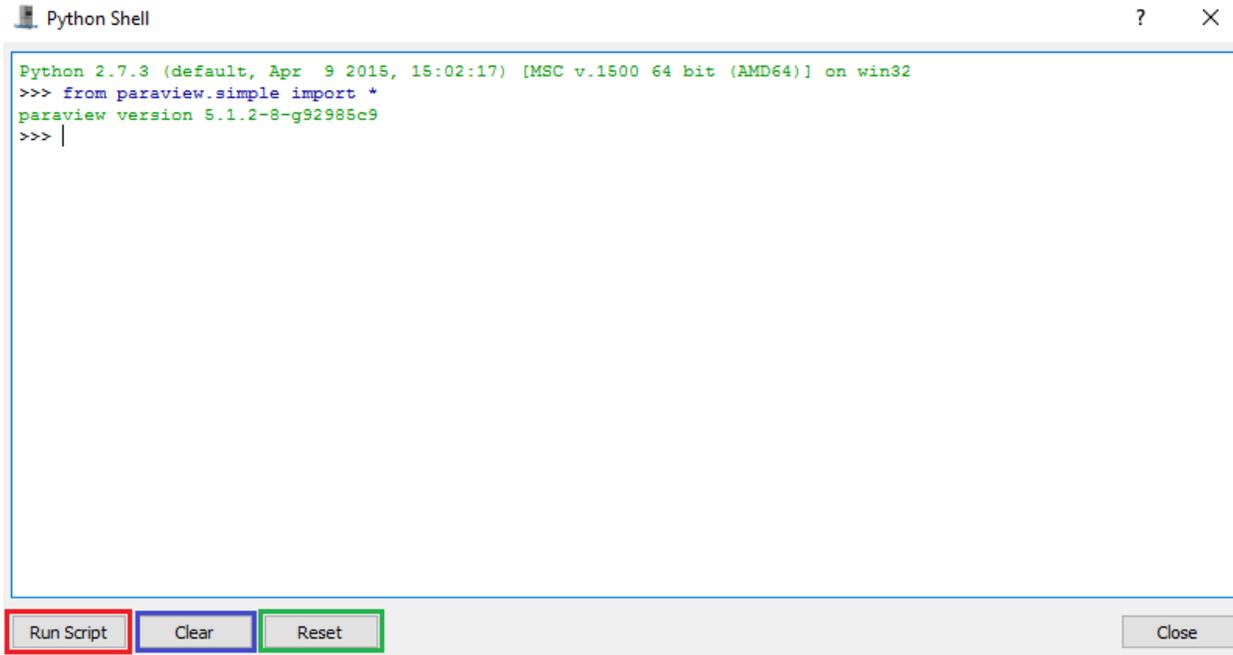


Figure II.F.5: The properties of the Pass Array filter.

## III. Python Console

### A. Introduction

Ouster Studio is based on ParaView which allows to create some task automation pipelines or small filters using the Python Shell console. Another convenient tool is the “Ouster Studio --script” directly derived from ParaView. It is possible to launch Ouster Studio with the following argument “Ouster Studio --script path\_to\_my\_script” to directly run a python script when Ouster Studio is launched. However, the automation is limited by functions which will require a user interaction such as choosing a sensor calibration. This manual goes through accessing data and programming into the python console. Then, we will show some examples of python programming using Ouster Studio specific functions. Finally, we will expose a non-exhaustive list of Ouster Studio features accessible from Python.



## B. Presentation of the Python Shell

### B.1 Introduction to Python-Ouster Studio features

The python console is a Python environment available in Ouster Studio which gives you access to some python, paraview, Qt and Ouster Studio features. The Ouster Studio features give access to the data provided by Ouster lidar sensors and gives some basic manipulation and analysis functions. Paraview provides lots of built-in filters to be applied on the data. Qt is in charge of the graphical user interface (GUI) and can be controlled with python too. Finally, you get all the usual Python scripting functionalities for task automation.

### B.2 Opening the Python Console

Once an Ouster Studio instance is running, the "Python Console" can be opened in three different ways:

1. Clicking on the python icon in the "Basic Controls" bar (see Figure I.A.1)
2. Clicking on the python icon in "Tools > Python Console" in the Menu Bar
3. Pressing F8 on the keyboard

### B.3 Presentation of the Python shell

The python console / shell presents 4 different options:

1. Run Script: Allow you to load and run a Python script.
2. Clear: Clear the console. The information of the console are cleared but the state remains. All your variables and data will not be removed.

3. Reset: Reset the console. All the variables and data will be removed. Nethertheless, the console won't be cleared. If you want to restart with a fresh new console you can Clear and Reset the console.
4. Tab: After pressing the "Tab" key, the Python Shell shows you the available autocompletion.

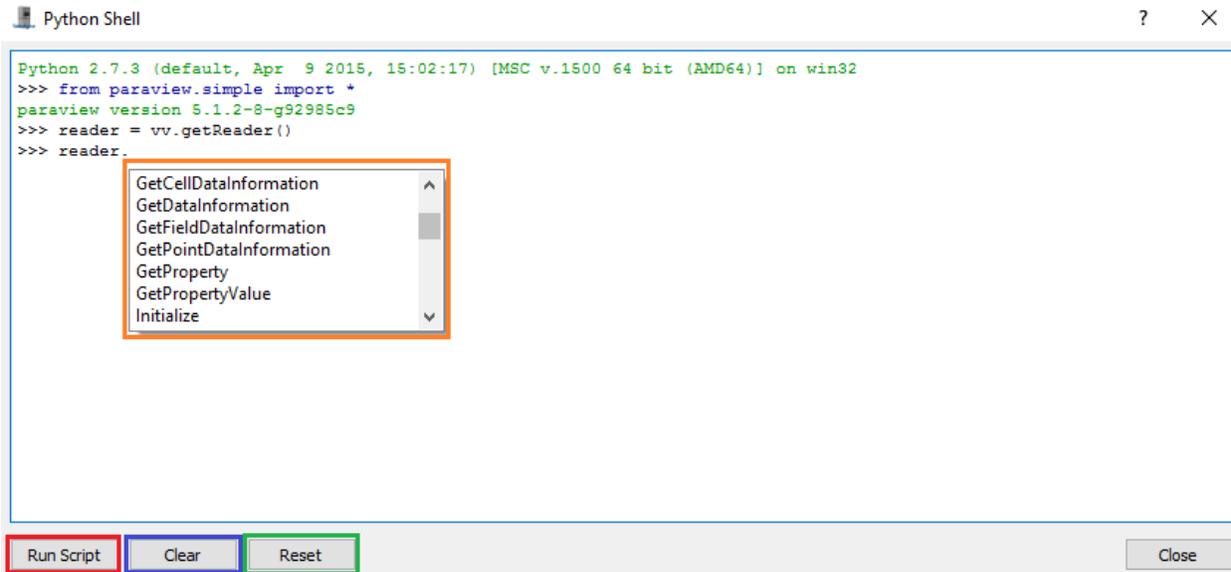


Figure 22: The Python Shell, Red: Run script option, Blue: Clear option, Green: Reset option, Orange: Autocompletion option

## C. Ouster Studio and VTK Glossary and Datamodel

In Ouster Studio the data is stored using VTK "vtkPolyData" objects. It is a dataset containing two things: the 3D geometry (the list of 3D points provided by the Ouster sensor) and the extra data associated to these points (intensity, timestamp, azimuth, laser ID, ...). The association of the points and their corresponding data is called the polydata or dataset.

The readers available in Ouster Studio read the data from the network (streaming reader) or from .pcap files (pcap reader) and provides the vtkPolyData corresponding to the data. Hence, when we get the output of a reader in the Python Shell we get a vtkPolyData object. The structure of a vtkPolyData can be found here: <http://www.vtk.org/doc/release/5.4/html/a01260.html>.

What should be retained is that the geometric points are accessible through the method ".GetPoints()" and the extra data are stored in arrays accessible through the method ".GetPointData()".

On the "GetClientSideObject()":

TL;DR: You don't need to understand the client/server paradigm: just base yourself on examples and auto-completion.

Long story:

Usually the Ouster Studio functions will provide a servermanager containing the object you want. It is because Ouster Studio uses the Paraview Client / Server architecture. When you access a reader with the function "lv.getReader" for instance, you will access a servermanager instance on the reader. If you want to get the output or have access to the method of the object you must use "GetClientSideObject()"

function. For instance, if you want to get the vtkPolyData output of the reader you must write “getReader().GetClientSideObject().GetOutput()”.

## D. Examples

### D.1 Get Data Information

The first example will show you how to get the point cloud generated by the Ouster sensor.

Step 1: Open the provided .pcap example: “roundabout.pcap”

Step 2: Open the Python Shell

Step 3: Write these python code lines:

```
reader = lv.getReader() # Get the .pcap reader
cloudInfo = reader.GetClientSideObject().GetOutput() # Get the output of the reader -> dataset stored in a vtkPolyData class
points = cloudInfo.GetPoints() # Get the 3D points list
times = cloudInfo.GetPointData().GetArray("T") # Get the time information of the points
times.GetValue(10) # Value of the time information for the 10th point
points.GetPoint(10) # Coordinates of the 10th point
```

Explanation: In this example we get the point cloud associated to the first frame of the “roundabout.pcap” data.

1. As you can see, all the functions available in Ouster Studio’s Python module are stored in the module named “lv”. For instance, if you want to access to the function stored in the module vtk, you must write: lv.vtk.FunctionWanted.
2. getReader is the Ouster Studio function which returns the .pcap reader. The .pcap reader is a vtk filter whose output is the dataset of the current frame. The outputted dataset is stored in a vtkPolyData class. It contains the 3D points list and the extra data associated to these points
3. The method GetPointData() of the vtkPolyData class give you access to the data arrays stored in the dataset. Using the method GetArray(“ArrayName”) you will have access to the data of the corresponding array. The exact names of the arrays available in the point cloud are visible in the spreadsheet view. The arrays available depend on the data (Type of sensor, if GPS is available or not, IMU available or not, extra data, ...).

## E. List of Functionalities

Here we will give a non-exhaustive list of all functionalities available in Python-Ouster Studio

### E.1 Simple Paraview: “lv.smp.”

All functionalities provided by the module `paraview.simple` are available using `"lv.smp."`. You can find the description and documentation of the functionalities here:

<https://www.paraview.org/ParaView3/Doc/Nightly/www/py-doc/paraview.simple.html>

## E.2 Qt "lv.QtGui"

All functionalities provided by the module `QtGui` are available using `"lv.QtGui."`. You can find the description and documentation of the functionalities:here:

<http://pyqt.sourceforge.net/Docs/PyQt4/qtgui.html>

## E.3 VTK Python

All functionalities provided by the module `vtk` are available using `"lv.vtk."`. You can find the description and documentation of the functionalities here: <http://www.vtk.org/Wiki/VTK/Examples/Python>

# Questions, support and requests

For questions and troubleshooting, please contact:

[ouster-studio-support@ouster.io](mailto:ouster-studio-support@ouster.io)

If you have questions about your Lidar sensor configuration or usage, please contact your Ouster lidar sales representative.

If you would like to hire Kitware to develop new features or a specific version of Ouster Studio tailored to your requirements, please contact:

Kitware: <https://www.kitware.com/contact-us/>

or directly:

Bastien Jacquet: [bastien.jacquet@kitware.com](mailto:bastien.jacquet@kitware.com)