GPR data processing automation and 3D visualization

Gabriele Benedetti

1. Introduction

This brief report has the main objective to illustrate an open-source compatible processing pipeline to automatically elaborate Ground Penetrating Radar (**GPR**) data to output a GPR volume (in VTK format). Once ready, the volume then can be used to intuitively localize, identify and analyze in 3D space buried objects in the top most layer of soil (up to a depth of 1m). The results of this analysis can be useful for many different applications ranging from geotechnics and archaeology to more peculiar subjects such as forensic sciences (Kelly et al. 2021).

1.1. GPR data

Ground Penetrating Radar (**GPR**) is a relatively new indirect non-invasive geophysical exploration method that can be used to acquire subsurface data up to a depth of 50m[a] . This technique is based on the capture of reflected high frequency electromagnetic signals emitted from a controlled source. The contrast of electrical properties in the soil given by presence of anomalous objects, cavities, sedimentary structures, change in lithology or in water contents induce a variation in the signal's propagation velocity thus producing a return pulse that is recorded (Neal et al. 2000). By visualizing the acquired data it is then possible to define the presence of perturbations across the medium and carry out different interpretations.

GPR data is recorded in traces grouped and organized in segments saved as SEGY files, every trace represents a point in space where the pulse has been emitted and the return signals recorded. A segment can be comprised of multiple sub-segments where the start and end (markers) can be defined during the recording or manually in the preprocessing phase. Markers can also be used to highlight points of interest encountered during the acquisition. These and any many other types of information are stored in the SEGY file header of both the segment and the single trace (see [**Tab.A.1**] for an example).

The data analyzed in this report was acquired in Piazza della Scienza (MI) and the acquisition area extends for 10.7m along the X axis, for 11.73m along the Y axis and for about 2.25m in depth [**Fig.B.1**].

It is possible to obtain the maximum depth reached by the emitted signal by defining a conversion parameter (**c**) and calculating the time range (**tr**). The depth will be calculated as follows

$$d[m] = \frac{\mathrm{c\ [m/ns]} \cdot \mathrm{tr\ [ns]}}{2}$$

The time range can be obtained using the sample interval (**dt**) and the number of samples per trace (**s**) both recorded in the header.

$$tr[ns] = \frac{\mathrm{dt\ [ps]}}{1000} \cdot \mathrm{s\ [\text{-}]}$$

[a] Best case scenario

2. Processing pipeline

The processing pipline adopted to elaborate and visualize the data is composed by a pre-processing and processing phase. Once the data is downloaded the user needs to take in to account a series of characteristics. Firstly it is necessary to read the header to define the geometry of the survey using the trace position information and the presence of markers. If these types of information are present or not required, the pre-processing step can be skipped and the data directly used. In this case the GPS information was not present and because of this synthetic positional data were used. To recover the lost information a constant spacing method was adopted considering the distance between two traces along the same segment constant (=0.96 cm).

For every segment/sub-segment the returning signal data needs to be read and every trace has to be processed to a certain degree. It is better not to use the raw data or heavily processed data since the interpretation of both can result difficult. Signal noise can hide the objects or lead to wrong object geometry reconstruction and on the other hand too much processing/filtering can mask some objects or create artifacts also causing interpretation problems (Neal et al. 2000).

In this report the processing steps are divided in two main points:

- Mean trace removal
- First break correction

The mean trace removal procedure is a signal detrending method that removes the mean signal of the given segment from every trace in the segment. This procedure removes the source signal present in the different traces.
The second processing step considers that, since the antenna is not directly in contact with the ground, the first return signal is given by the change in medium passing from air to the soil. To remove the air stratum a first break correction (**FB**) is applied by defining a sample number before which the data is not considered or visualized. This step is very important if the position and depth of the target needs to be known with high precision. In this case an automatic picking procedure was adopted using the Baer et al. 1987 algorithm applied to the mean signal of the given segment. The function used (*obspy.trigger.pk_baer*) is provided by the **obspy** Python package (Beyreuther et al. 2010). The parameters used follow as a reference the example in the documentation. Only the *thr1*, *preset_len* and *p_dur* where changed to accomodate the short arrival time and low noise (see [**Tab.A.2**]). Finally a filter can also be applied to remove the remaining noise. In this case it was not necessary and so the filtering step was skipped.

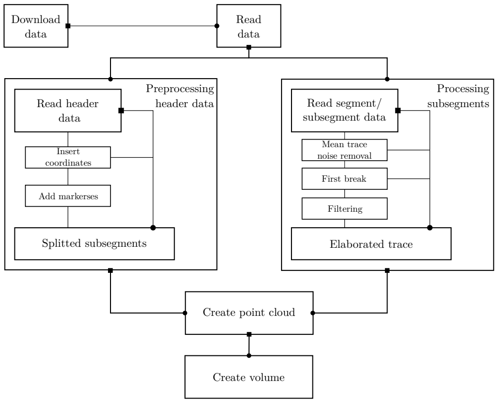


Figure 1: Adopted processing pipline

Once the header information is gathered and the signal data processed it is possible to create a uniform 3D point cloud. This is achieved by dividing the traces in multiple intervals (integration depth) and linearly interpolating on a 2D depth slice the mean trace intensities in the given depth interval[b] . In this case an integration depth of 1cm[c] was used. Although this approach effectively reduces the number of samples along the Z axis, thus lowering the vertical resolution of the model, it improves the overall quality of the data and highlights the presence of anomalies.

[b] The depth slice is an "image" of (nx;ny) pixels where for each pixel the respective calculated value is assigned. The absolute interpolated value was used, but squaring too is a viable option

[c] about 5 samples per trace

The intensity value of each point is then assigned to a cell (voxel) creating a GPR volume. The final computed GPR volume is 1m deep with a resolution of 600x600 cells giving a final grid resolution of 600x600x100. The grid, volume creation and final analysis were achieved using Pyvista (Sullivan et al. 2019).

3. Results

The automatic first break values were successfully determined using the Baer algorithm. The advantages gained by using this method are several related to both the speeding up of the overall processing time and the precision in the picking. By using other methods such as manual first break picking or assuming a constant FB, errors could be introduced since it is likely that distinct segments have different FB values even if the topography is apparently flat (such as in this case study) [**Fig.2**]. Moreover an automatic FB picking method could prove most useful in drone based GPR acquisition since the altitude of the drone or the topography below can rapidly change.

The GPR volume too was successfully constructed but the pipeline needs to be used with other datasets to confirm it's validity. There are some expected flaws present in the final volume such as an interpolation "hole" in the top right quadrant and a persistent smearing effect parallel to the X axis[d] . Both of these flaws are related to the acquisition geometry given by the presence of obstacles such as trees and the absence of cross-line data (see [**Fig.B.1**]).

Nonetheless, using the newly defined volume three main objects where analyzed and measured. Starting at Y= 7.6m and X = 4m in the first 0.1-0.3 cm a 5 m long and 0.4 m wide object was defined and classified as an underground pipe (verified by on site inspection). At Y=9.3m, a second object 0.3cm to 0.7m deep and 0.7m wide was defined running along the whole volume. By comparing satellite images with the reconstructed geometry, service manholes approximately align along this feature leading to hypothesize that could be an underground wire canal running parallel to the lightrail track.

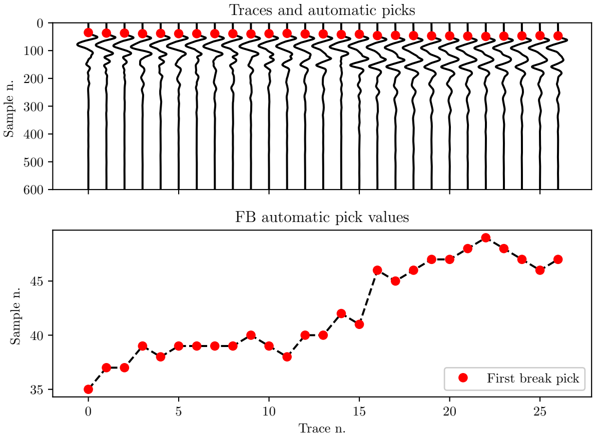
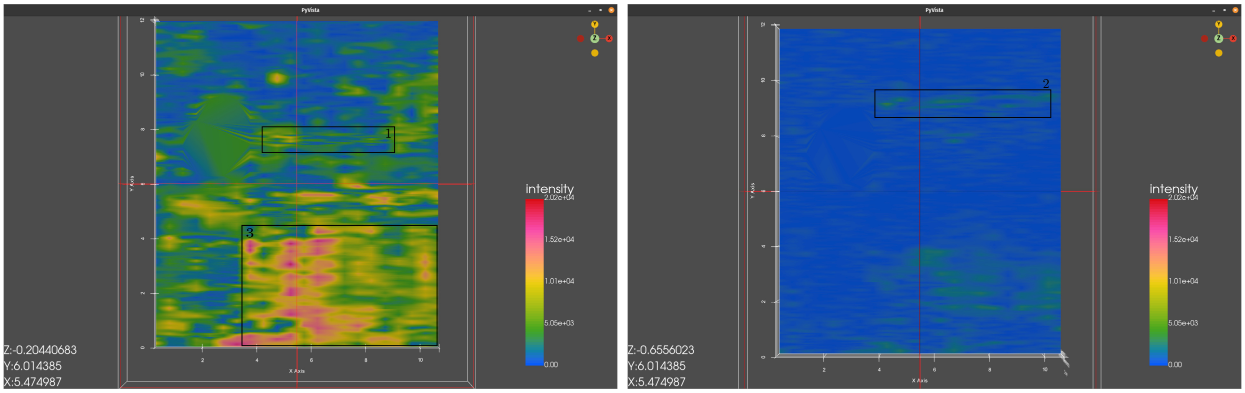


Figure 2: Automatic first break determination using the Baer algorithm. In the shown graphs is visible an increasing trend of the FB values .

Finally occupying the bottom right corner of the volume an extensive reflector 0.2m to 0.8m deep was identified. This last object could be a structural part related to the presence of the university's underground complexes (such as walkways, rooms and study areas).

[d] Visible in the crossline and planar sections



(a) **(b)**

Figure 3: Resulting depth slices with highlighted the different objects found.

4. Conclusions and final remarks

The validity of the proposed pipline was demonstrated by automatically picking first break values and creating a functional volume then used to localize, identify and measure different objects present in the first meter of soil. Even if the final interpolated volume presents smearing artifacts, probably caused by the missing cross-line data, it is still possible to quickly define and interpret the presence of objects. With GPS information and a proper acquisition geometry a precise geolocated volume could be obtained and used with other types of data. Paring, for example, the resulting volume with Digital Elevation Models could provide a quick and intuitive way to define the location and geometry of underground objects.

This method could be a precise, rapid and cheap subsurface exploration alternative thanks to the relative ease of the acquisition, elaboration speed and the flexibility provided by the Python packages.

References

[1] Baer M. and Kradolfer U. An Automatic Phase Picker for Local and Teleseismic Events. In: *Bulletin of the Seismological Society of America* 77.4 (Aug. 1987), pp. 1437–1445. ISSN: 1943-3573, 0037-1106. DOI: 10.1785/BSSA0770041437.

[2] Beyreuther M., Barsch R., Krischer L., Megies T., Behr Y., and Wassermann J. ObsPy: A Python Toolbox for Seismology. In: *Seismological Research Letters* 81.3 (May 2010), pp. 530–533. ISSN: 0895-0695. DOI: 10.1785/gssrl.81.3.530.

[3] Kelly T.B., Angel M.N., O'Connor D.E., Huff C.C., Morris L.E., and Wach G.D. A Novel Approach to 3D Modelling Ground-Penetrating Radar (GPR) Data – A Case Study of a Cemetery and Applications for Criminal Investigation. In: *Forensic Science International* 325 (Aug. 2021), p. 110882. ISSN: 03790738. DOI: 10.1016/j.forsciint.2021.110882.

[4] Neal Adrian and Roberts Clive L. Applications of Ground-Penetrating Radar (GPR) to Sedimentological, Geomorphological and Geoarchaeological Studies in Coastal Environments. In: *Geological Society, London, Special Publications* 175.1 (2000), pp. 139–171. ISSN: 0305-8719, 2041-4927. DOI: 10.1144/GSL.SP.2000.175.01.12.

[5] Sullivan C. and Kaszynski Alexander. PyVista: 3D Plotting and Mesh Analysis through a Streamlined Interface for the Visualization Toolkit (VTK). In: *Journal of Open Source Software* 4.37 (May 2019), p. 1450. ISSN: 2475-9066. DOI: 10.21105/joss.01450.

A. Tables

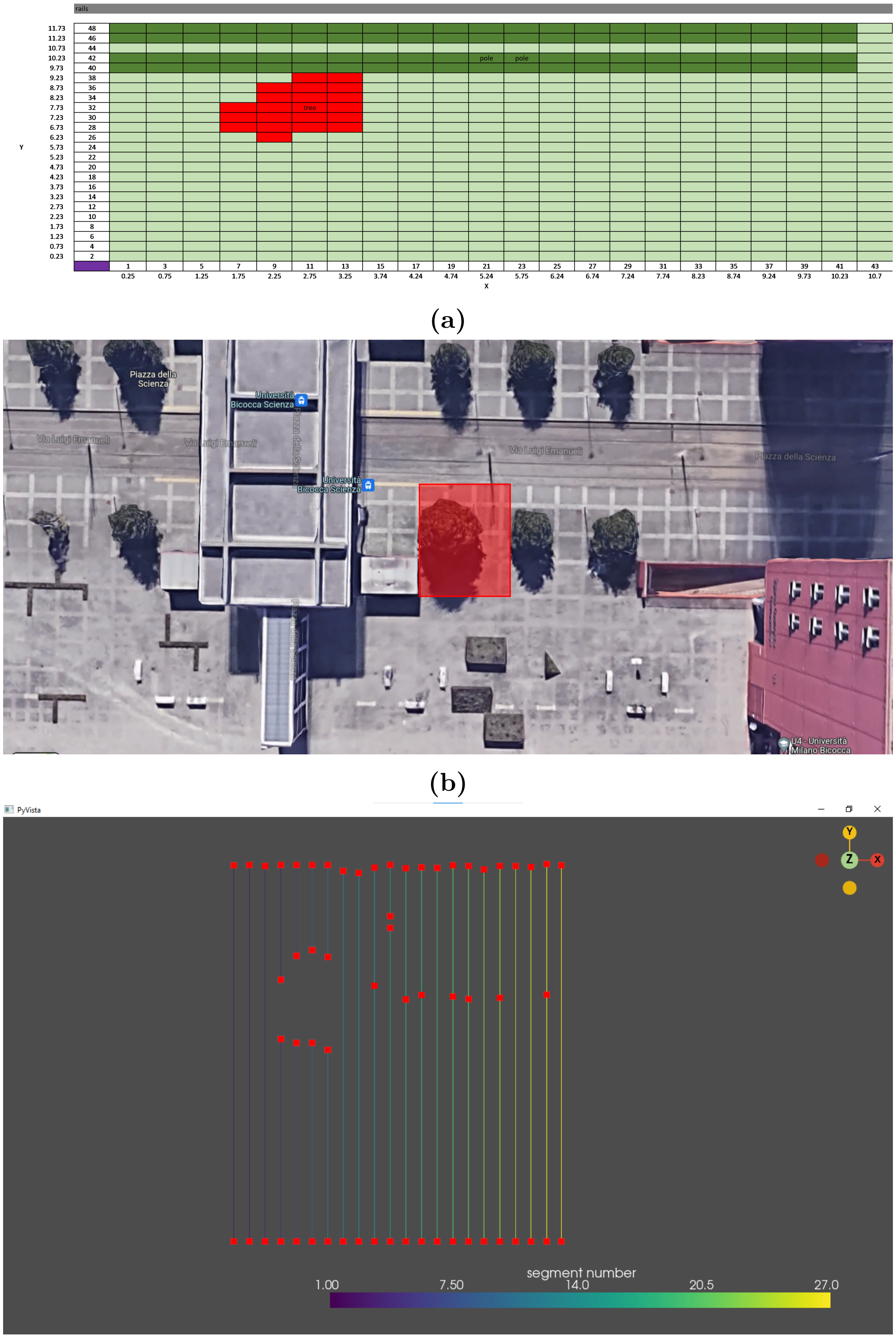
Parameter	**Value**
n. of channels	2
n. of files (segments)	27
n. of subsegment per file	1
GPS data	NO
Antenna frequency used [Hz]	20833.3
Conversion parameter [m/ns]	0.09
Time range [ns]	50
Sample interval [ps]	0.049
number of samples per tr.	1024

Table A.1: Summary dataset description table.

Parameter	**Value**
reltrc	segment mean
samp_int [Hz]	2.08333x10^{10}
tdownmax	20
tupevent	60
thr1	50
thr2	12
preset_len	15
p_dur	15

Table A.2: Used parameters for the Baer algorithm requested from the *obspy.trigger.pk_baer* function. Highlighet in bold the parameters changed.

B. Additional images



(a)

(b)

(c)

Figure B.1: Acquistion geometry (a), reference satellite image (b) and segment order (with markers) (c).

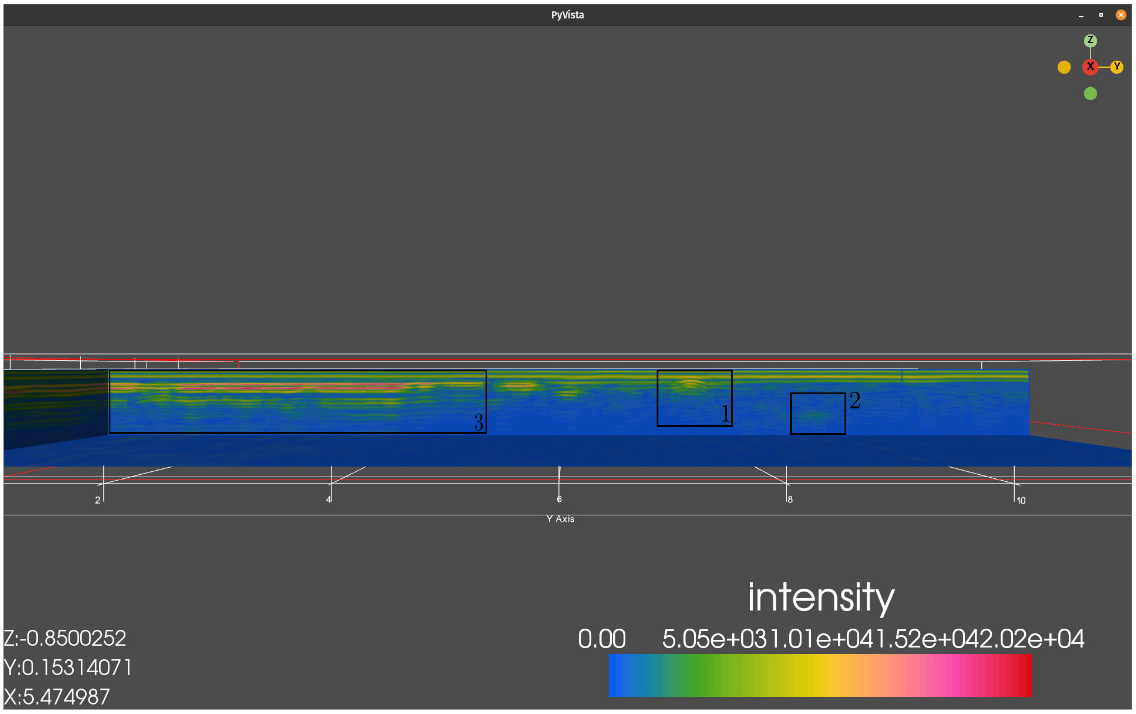
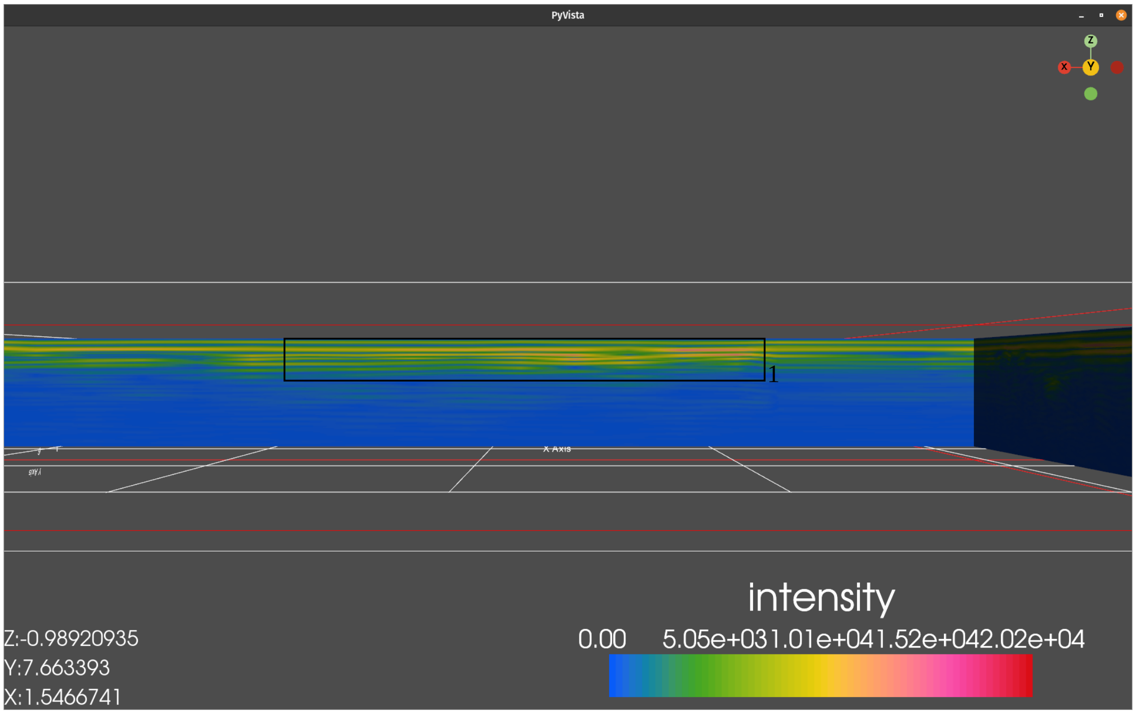
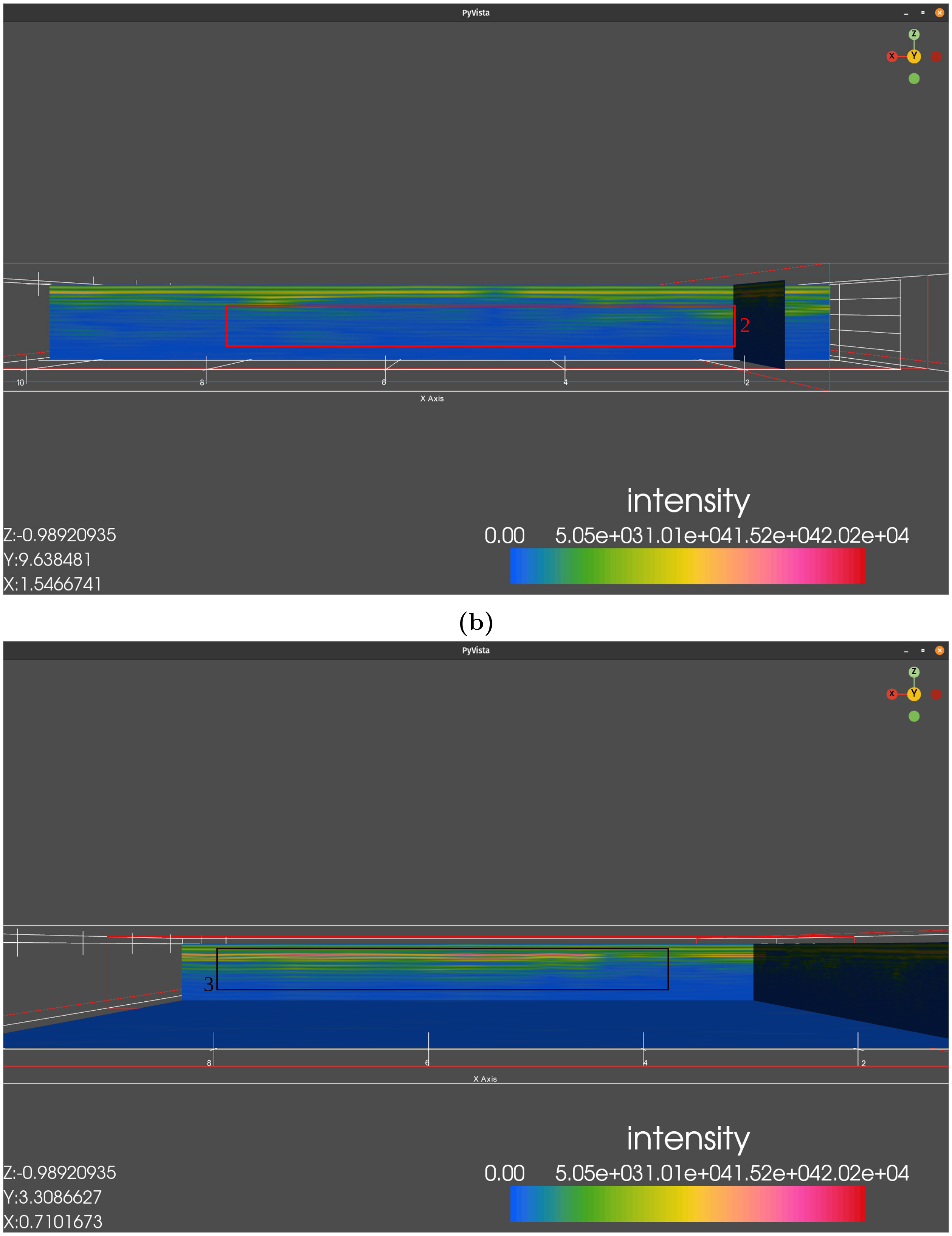


Figure B.2: Inline section (X=5.5) intersecting the three object analyzed in the main text



(a)

Figure B.3: Crosslines of the different objects



Z:-0.98920935
Y:9.638481
X:1.5466741

(b)

Z:-0.98920935
Y:3.3086627
X:0.7101673

(c)

Figure B.3: Crosslines of the different obejcts