

## Geometry Clean Up and QA/QC of Raptor Data

The introductory note in this series offered advice on the collection of GPR array data. The key takeaway of which was that a real-world project could contain data that is less than optimal. Therefore, this note focuses on the topic of data QA/ QC and will discuss useful tools for the selection and management of data imports into processing software. The primary objective being the import of quality data and efficient workflows.

Figure 1 shows a project containing 2800 individual GPR-profiles, which combine to form 175 input files for easier management, although, still a substantial number to handle.

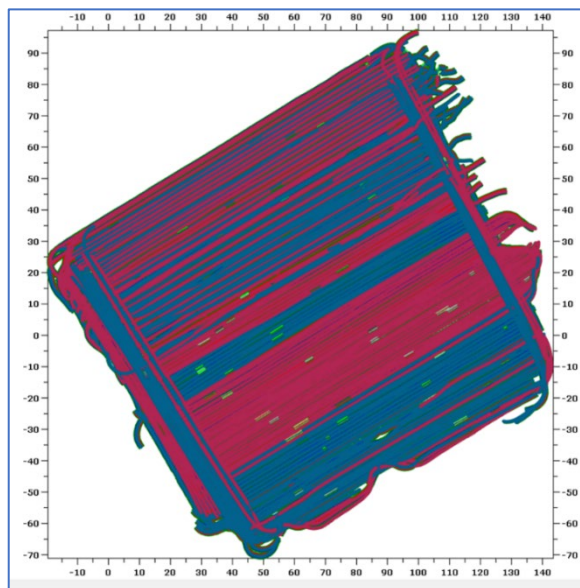


Figure 1, A project, with 175 input files, containing 2800 individual radar profiles

Further, the original data contains over 70000 positioning points, a large percentage of which are problematic — consequently, data sets such as these require practical tools to sort out problems early on before processing.

Figure 2 shows a close up of one part of this project, where the zoom function reveals clear positioning errors (self-intersecting swaths) as well as data swaths that don't make sense.

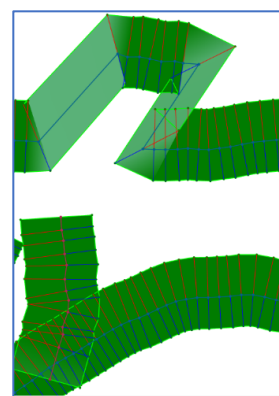


Figure 2, Zoomed picture revealing erroneous data

### Swath statistics

Figure 3 shows the swath statistics tool, which is a useful first step to identify essential data readings outside the project norms. In the example shown, the average position density is approx. 3-4/m, but some read as low as 0.06/m. It is safe to assume that these swaths will cause a problem if they import, so uncheck to ignore.

	Swath	Points/m	Traces/m	Start time	End time	Total length, m
<input checked="" type="checkbox"/>	58 ALDERSHOT_...	3.78	20.78	11:21	11:21	105.002
<input checked="" type="checkbox"/>	59 ALDERSHOT_...	0.06	0.16	11:39	11:40	88.172
<input type="checkbox"/>	60 ALDERSHOT_...	0.06	0.08	11:48	11:48	84.096
<input type="checkbox"/>	61 ALDERSHOT_...	0.06	0.08	11:50	11:50	95.826
<input checked="" type="checkbox"/>	62 ALDERSHOT_...	4.43	20.72	08:31	08:32	131.700
<input checked="" type="checkbox"/>	63 ALDERSHOT_...	3.20	21.33	08:32	08:33	121.351

Figure 3, Statistics showing positioning and data density and total length of profiles

Another noticeable variation is in swath length, where some files indicate only a few meters versus an average of 125 m; again, a simple uncheck of the problem files will omit them from import.

### Colour coding statistics

Colour coding is a simple way to highlight swaths with problematic positioning density. Easy to identify visually, a simple cursor mouse-over reveals specific information concerning the swath file name and position, as per the example in Figure 4. The density of radar data may be treated in a similar way to highlight problems with the odometer values and wheel slip.

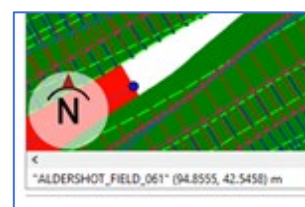


Figure 4, Identifying outliers by colour coding

## Geometry Clean Up and QA/QC of Raptor Data

### Removal of positioning data

Continuing with the same project example, a lot of data on the perimeter will not process well in 3D, so removing such points will speed up the data processing and reduce the amount of PC storage space required. Figure 5 shows an example of a simple tool to mark and remove such positioning points.

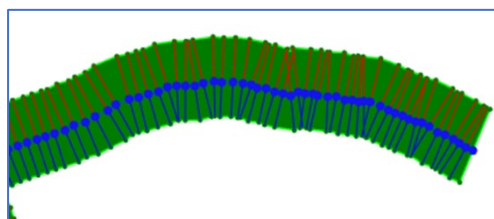


Figure 5, means of effectively mark and delete positioning points, may significantly reduce workload in later stages

### Reducing the positioning density

As indicated earlier, swaths with a very high density of positioning will be problematic upon import, which is due to the self-intersection of swaths, an effect typically caused by the GPS antenna swaying side-to-side. Reducing the positioning density by half from 3-4/m to 1-2/m will decrease this problem. Although it does mean removing some data from the project, it will simplify and speed up the processing time.

### Final clean-up and radar data import

Even after observing the preceding steps, there may still be some cause for errors in the data. Modern processing software should be able to warn the user of this and guide them on where to search for such errors, as per the example shown in Figure 6.

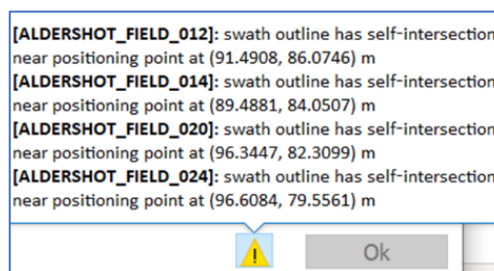


Figure 6, Additional statistics showing self-intersections in data

Figures 7 and 8 illustrate how this project will look after following the methodology described above, both as a

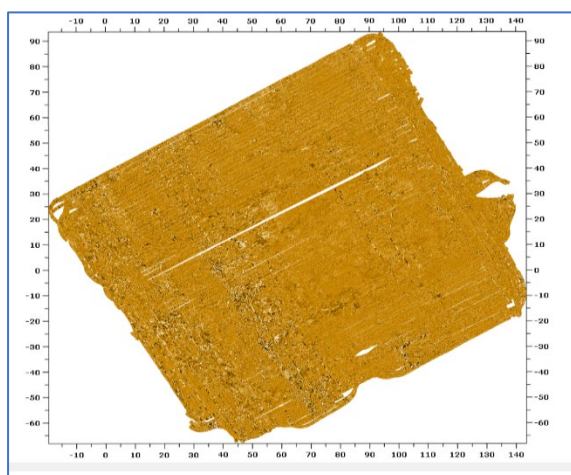


Figure 8, whole project, radar data shown

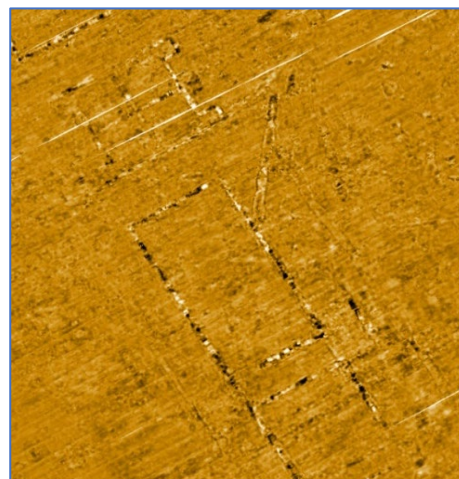


Figure 7, Zoomed section

whole project and a section close up using zoom.

An important takeaway from this exercise is that the geometry requires cleaning as much as possible before the import of radar data. Processing radar data takes a lot of computer memory and can slow operations down. In this instance, none of the geometry was moved as that would be very difficult since we have no references, before radar data import. The next note will show how we may use visible object in the radar data to correct for some positioning errors, besides some other hints.