

Tools, Tips, and Workflows

Automatic Ground Classification of Dense Point Clouds in LP360



LP360, versions 2014.1 and above

Lewis Graham

November 5, 2017

Revision 1.0

The density of point clouds (usually expressed as points per square meter) is increasing on what seems to be a monthly basis. This is particularly true of point clouds derived from Dense Image Matching (DIM), the clouds produced from software such as PhotoScan, Pix4D, DroneDeploy and so forth.

You can quickly check the density of a point cloud in LP360 using the Point Cloud Task (PCT), Point Cloud Statistics Extractor. Figure 1 shows this PCT set up to collect Point Count and Point Density.

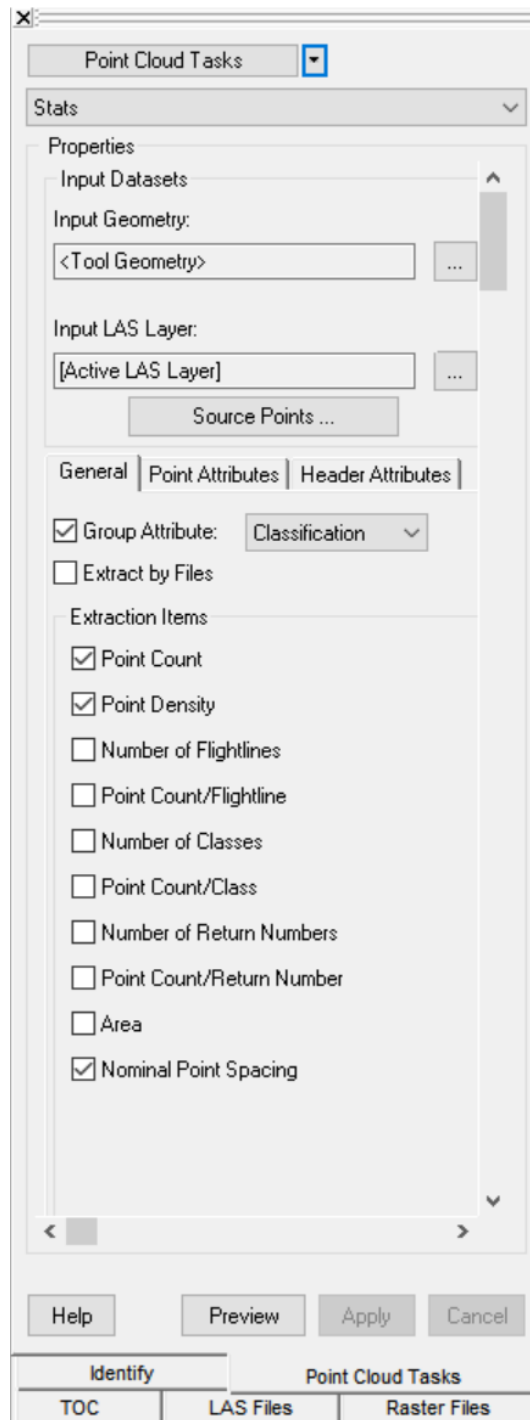
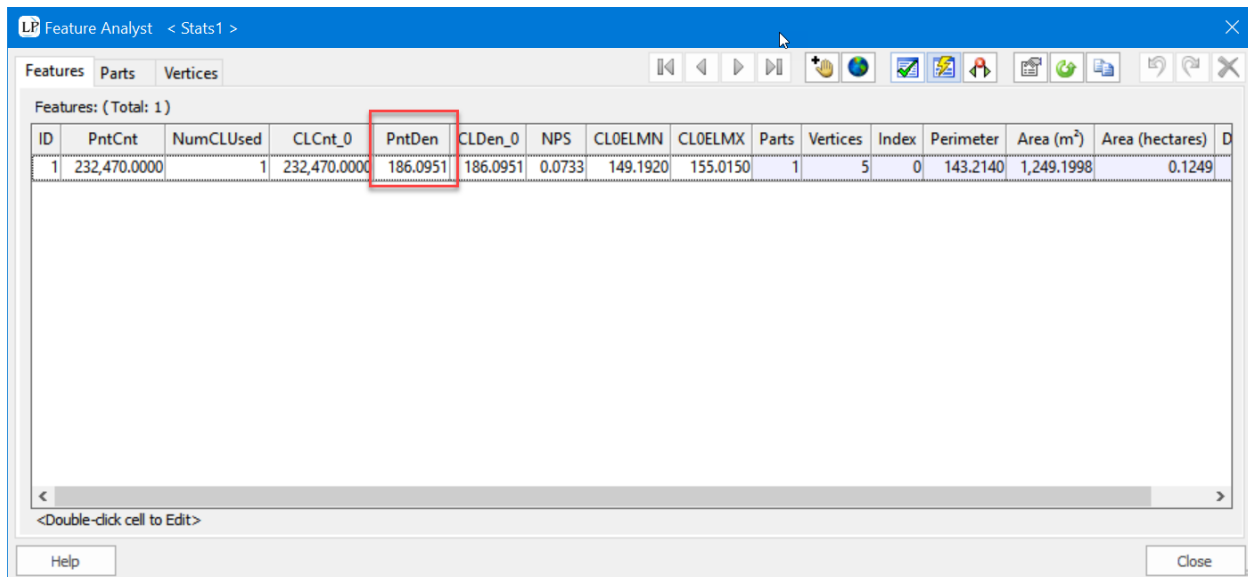


Figure 1: The Planar Stats PCT

When I run this on a demonstration mine site we collected using PhotoScan Pro to generate the dense point cloud, I get about 186 points per meter squared (pts/m²) -see Figure 2. Note that the Point Cloud Statistics Extractor conveniently also provides the nominal point spacing (NPS); for this case, about 7.3 cm.



ID	PntCnt	NumCLUUsed	CLCnt_0	PntDen	CLDen_0	NPS	CLOELMN	CLOELMX	Parts	Vertices	Index	Perimeter	Area (m ²)	Area (hectares)	D
1	232,470.0000	1	232,470.0000	186.0951	186.0951	0.0733	149.1920	155.0150	1	5	0	143.2140	1,249.1998	0.1249	

Figure 2: Point density from the Point Cloud Statistics Extractor

Point clouds with densities exceeding about 25 pts/m² (an NPS of 20 cm or smaller) are a challenge to process with triangulation-based ground classifiers due to the large memory footprint required of the triangulation code. This is true of any software (which includes LP360) that uses variations of the adaptive TIN algorithm. When these ground classification algorithms were designed, an 8 pts/m² point cloud was considered very high density!

The adaptive TIN ground classifier in LP360 is an iterative algorithm that works basically as follows:

1. The data are divided into a virtual grid. The size of each grid cell is a user set *seed* value. This seed value must be chosen as small as possible but not so small that a non-ground structure (e.g. a building) is completely contained within a cell (we'll see why in the next step).
2. Within each cell, the lowest point in the cell that meets a user defined point filter criteria (this filter is used to filter out points already classified as non-ground such as

noise) is marked as a Ground point. These are the so-called seed points. Note that when the data are from a multiple return LIDAR system, we also always set the filter to exclude all but last return points since only last returns can be ground points. You can see that if a non-ground object such as a building completely fills a cell, the lowest point will not be on the ground. This will cause an error in subsequent processing steps.

3. Using the current ground points, a triangulated irregular network (TIN) is constructed. Note that on the first pass, this will be just the seed points from the previous step.
4. Points that pass the user defined input filter are tested for inclusion in the current TIN. If a point does not *bend* the TIN too much, is not too far from a current TIN facet and a few other tests, it is marked as a ground point.
5. The TIN is regenerated using the expanded set of ground points and the inclusion test is repeated until the process meets a stopping criteria such as a maximum number of iterations or a minimum number of added points in the last iteration.

The problem with dense data occurs when you need to select a large seed distance. You might have to do this if large buildings occur in the data set. For example, if you have square buildings that measure 100m on edge, your seed size would have to be a bit over 100m to ensure that the first step of the algorithm did not select a building roof point as a seed (the ground must be *seen* in every cell). The algorithm requires that an area of approximately 3 cells by 3 cells be processed as a unit in order to generate a representative TIN. As you can see, with large buildings and high-density clouds, this is a lot of points!

We are working on efficient ways to automatically adapt our ground classifier to these dense clouds, but, in the meantime, the data thinning algorithms in LP360 can be used to tackle these situations.

I will give an example here based on a dense image matching point cloud created from drone imagery collected over a mine site. Unlike LIDAR, these point clouds contain significant amounts of noise. These noise points are caused by correlation errors (technically, the wrong conjugate points have been chosen for the test point, driving the resultant elevation to a false value above or below the true surface). Our algorithm will proceed as follows:

1. Classify low noise points so they are not misinterpreted as ground points
2. Temporarily thin the point cloud by moving select points into an unused class
3. Run the ground classifier on this thinned set, using the proper setting for the seed size
4. Integrate the remaining points left in the thin cloud back into the full cloud

5. Set the seed size to a small value (to control the amount of memory needed) and tell the algorithm to use existing ground points as the seed rather than finding new seed points
6. Rerun the ground classifier on this dense data

This sounds a bit complicated because it is! Fortunately, LP360 contains a macro facility for combining point cloud tasks. This allows you to create the above steps a single time as a macro and then just use this on your dense clouds.

The first step is noise removal. LP360 has a great automated tool (a PCT, of course) for filtering both low noise and *isolated* points. I do not like to combine this into the ground filter macro because a manual quality check (QC) step is in order. If you allow any low noise points to remain in the source point class for ground classification, that point (if it is the lowest in a cell) will be marked as a seed point, corrupting the rest of the classification process. If the adaptive TIN filter has an Achilles heel, this is it! An example of low noise in a dense image matching (DIM) point cloud is depicted in Figure 3. If you are manually cleaning these (by reclassifying to Low Noise in the profile view), be aware that they generally only occur under areas of “confused” surface such as overhead structures and vegetation.

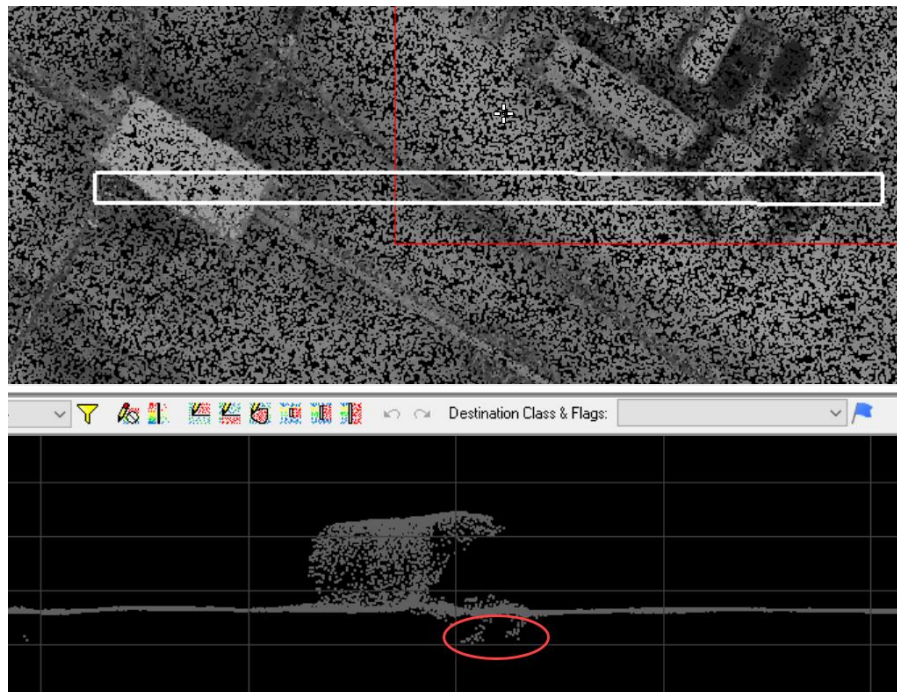


Figure 3: Low Noise in a DIM point cloud

Figure 4 depicts the same project area with the low points classified to Low Noise. You can make quick work of finding low points by simply scrolling through the project data in the profile view. I typically run the Low/Isolated Points PCT and then do inspection/cleanup with the profile tools.

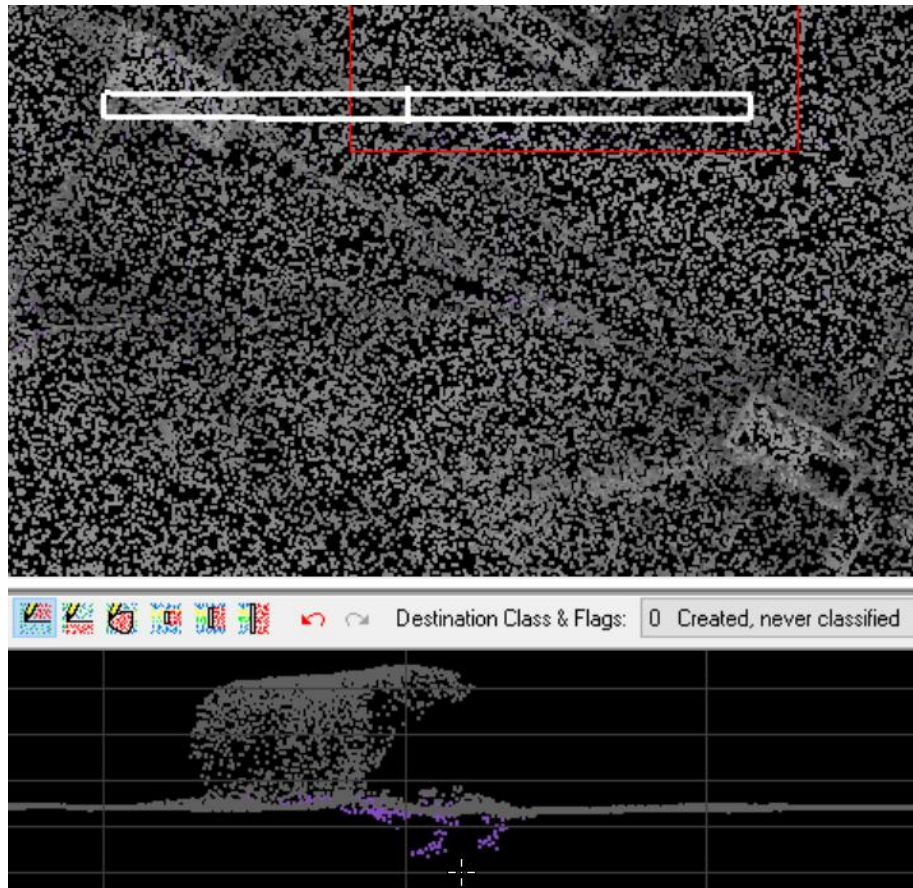


Figure 4: Classified Low Noise (purple points)

So let's now create the macro for ground classifying these points. The first step is to create the empty macro Point Cloud Task (PCT) by choosing "Macro" from the list of available Point Cloud Task templates (Figure 5). Give it a meaningful name, of course. I like to start the name with the principle thing it does since we list user created PCTs in alphabetical order. If you start all various ground filters with Ground, they will group together.

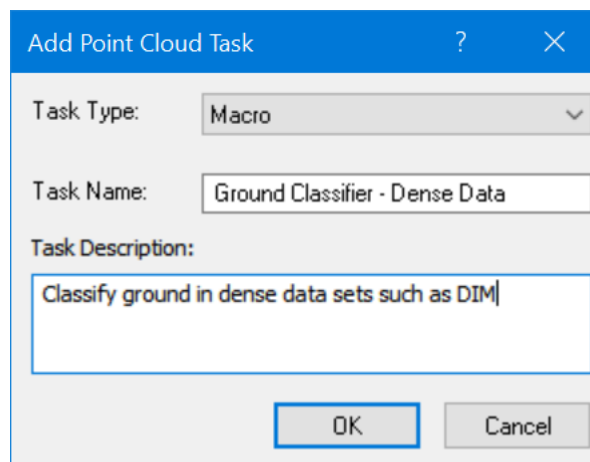


Figure 5: Creating the Macro shell

The first step to add to the macro is the PCT “Classify by Statistics.” This will be used to make a thinned dataset of the area that will ultimately be ground classified. What we will do is reclassify a subset of all the target points (target points being those eligible to be included in our ground class). I suggest the following:

Set the grid size to 1 meter. Set the source to class 0 (assuming your DIM generation or LIDAR processor puts all “never classified” points in class 0). Set the destination class to 25 or any unused class that is valid for the LAS Point Data Record Format (PDRF) that you are using - be careful not to use a class above 31 in a PDRF below 6 since these have only classes 0 through 31. Eight pts/m² is a pretty good density for ground classification. We have to ensure that we get the lowest point so set the thinning criteria as shown in Figure 6. Notice that I am selecting the minimum point in the 1 m² cell and then 7 additional random points. Also shown is the destination class, 25.

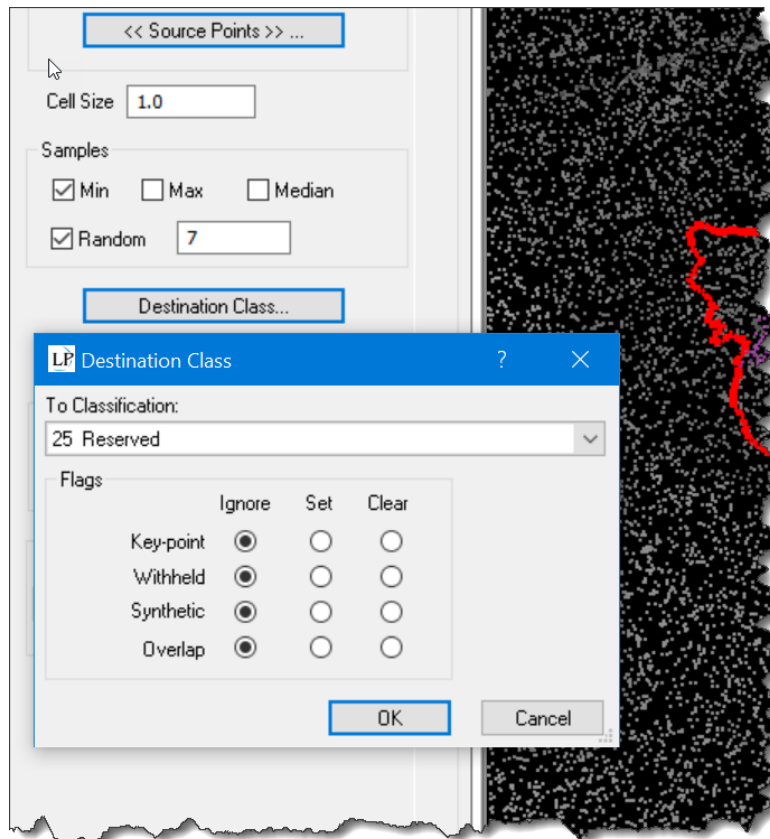


Figure 6: Statistics Filter settings

This will give you 8 pts/m² in the thinned class and ensure that you get the low point that is not low noise. If you want to work with denser data, just increase the number of random selected points. Make sure you do not ask for a grid to be generated since you will not need to visualize the virtual grid and it just adds unnecessary processing time.

The second step in the macro is to ground classify this thinned cloud. Thus, add the Adaptive TIN Ground Filter as the second step in the macro. Make sure you set the source filter to Class 25 (our thinned cloud) and the destination class to 2, the class normally used for ground. If you are using DIM data make sure to set the source Returns filter to All since different DIM software set different values for the return (DIM generated point clouds do not have multiple returns. Vendors should be setting all points to Return 1 of 1). Choose the “Generate Seed Points” option and set the seed spacing as normal (that is, to avoid classifying buildings). You can choose a low number of iterations since, at this step, we are just trying to flesh out our thin class 25 data with some ground points. My settings are shown in Figure 7. Note that I have

chosen a seed size of 15 meters. Thus, I expect all of my buildings to have a short edge of about 14 meters or less.

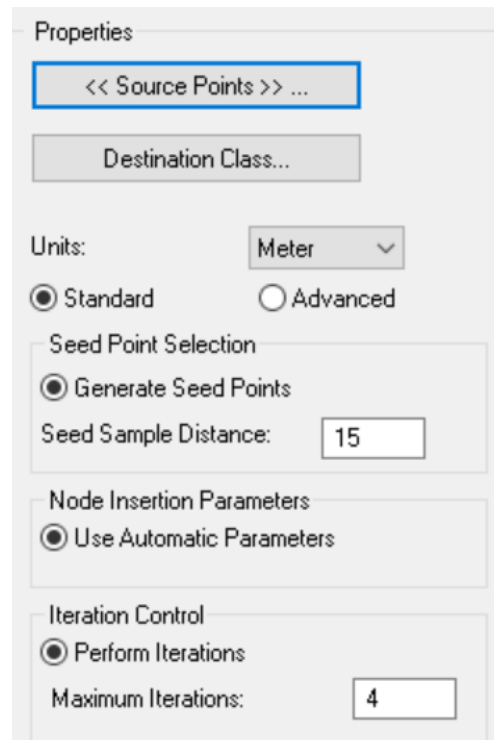


Figure 7: Thin cloud ground classification settings

After this step has run, you will have created a rarified ground class. You will still have a lot of points in Class 25 (or whatever temporary class you chose). These are points that were not considered because they failed the ground test or because they were not encountered in the 4 iterations of the algorithm. You will then add the Basic Filter PCT to the macro to move these Class 25 points back to the originating class (usually Class 0).

The final step of the macro is a ground classification run at full density but using the thin ground points created in the prior step as the “seed” points. To accomplish this, add the Adaptive TIN Ground Classifier as the fourth step in the macro. Set up the ground filter with class 0 as the source and “Use Existing Seed Points Only” as the seed strategy. Set the Seed spacing to something “large” that your machine can handle (on my 32 GB machine, 300 meters in LP360 (standalone 64-bit Windows version) uses up about 8 GB of system memory. You will have to

temporarily set the dialog to “Generate Seed Points” to enable the seed sample distance input. After keying in the desired value, toggle the option back to “Use Existing Seed Points Only.” Set the rest of the ground macro parameters according to your ground strategy. I usually stop at 4 iterations just for the sake of time since I do not see many more points entering ground class beyond this. These settings are shown in Figure 8.

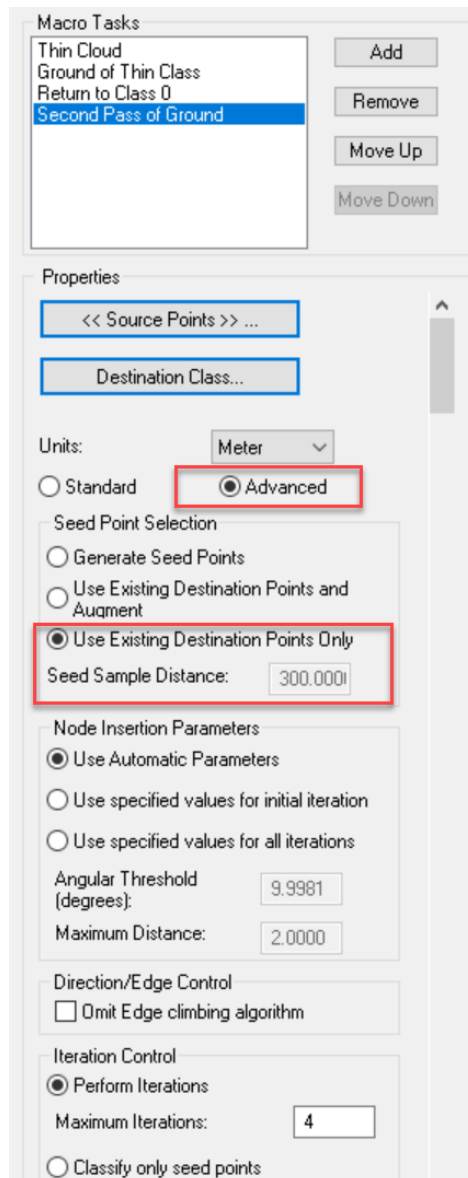


Figure 8: Second pass Ground filter

Note the setting of the seed span in step 4 is critical due to way LP360 chooses its processing tile size for the ground processing algorithm. Regardless of whether or not you select “use existing ground points,” LP360 still uses the seed span as its choice for processing tile size. We will change this in the 2017.2 build so that the seed size, in this case, will be ignored. The result of the macro on our test mine site is shown in Figure 9. Note that the macro has done quite a nice job on this very dense *SfM* data!

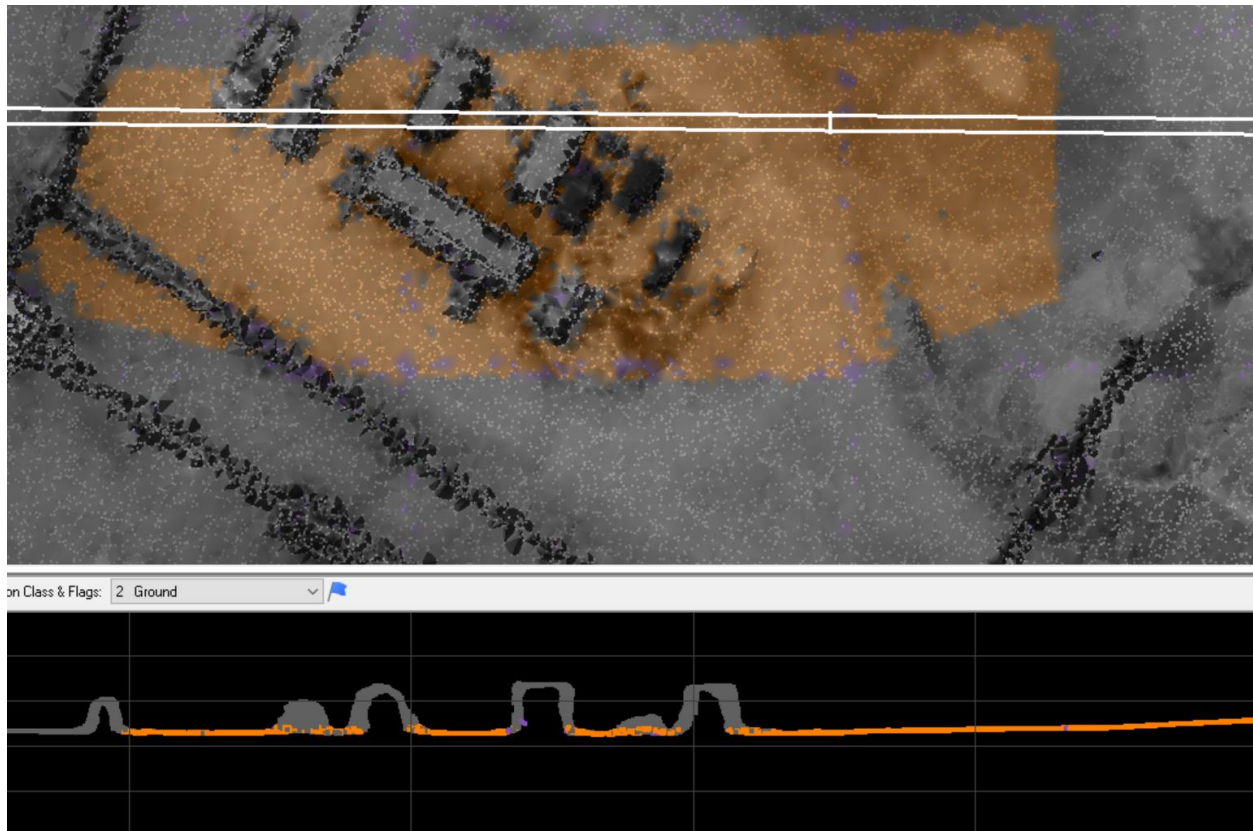


Figure 9: The resultant ground class

All of the above is for situations where, for some reason, you need a dense ground class.

If all you are doing is creating contours, doing volumes and so forth where the ground cloud is not subject to scrutiny, just do steps 1 – 3 and you are done. It actually makes a better product to have fewer points in the ground class (within limits, of course).