TerraSolid Guide & Reference 3
Sensor Calibration using TerraMatch
LiDAR Systems & Processing,
Centre of Geographic Sciences, NSCC

Overview

Although very accurate GPS and IMU hardware is used to determine the position and orientation of each laser pulse during LiDAR collection, systematic and/or non-systematic errors are often still present in the data.

TerraSolid's TerraMatch product is designed to address these issues by determining the various error model parameters (e.g. boresight alignment between the IMU & laser) necessary to ensure that points in overlapping flightlines "match" each other (e.g. along the roof of a building).

Since TerraMatch is a highly complex and a very powerful package, this guide provides an introduction to only the core concepts and functionality of the software. These topics include:

Coordinate Systems

- Defining project coordinate systems
- Projecting points during import

Trajectory Management

- Importing flightline trajectory data
  - specifying coordinate system & trajectory folder
  - importing trajectory data from SBET
  - displaying trajectory data

- Creating flightlines from trajectory data
  - automatic clipping using point data
  - manual clipping using polygon elements

- Assigning points to flightlines (i.e. trajectory data)
  - removing unassigned points

Analyzing Flightline Misalignment

- Analyzing point data to determine extent of calibration errors
  - heading, pitch, roll, scale, etc.
  - along-track & across-track profiles, saving views, etc.

Calibration Data Selection & Preparation

- Creating a calibration dataset
  - selecting survey blocks
  - creating blocks from polygons

- Preparing calibration points
  - classifying points by flightline

(continued on next page)

Boresight & Sensor Calibration

- TerraMatch tools & settings
    - Measure Match, Find Match, Apply Results
    - iteration convergence, standard deviations

- Analyzing points
    - managing multiple “data” folders within a TerraScan project
    - measuring difference between flightlines (*Measure Match*)
    - determining systematic corrections (*Find Match*)
    - applying corrections to “calibration” data and verifying results (*Apply results*)

- Applying corrections
    - applying corrections to full project data (*Apply Corrections*)
    - measuring difference between flightlines of corrected data (*Measure Match*)
    - adjusting project blocks to account for corrected point locations (i.e. reimporting points)

Elevation Adjustments

- Analyze & correct points
    - determining individual flightline z-shift adjustments
    - applying adjustments to project blocks
    - measuring the difference between corrected flightlines

Note, that while the sections in this guide are partially organized in the order you may typically perform them, they are not intended as a rigorous step-by-step tutorial. Please **read through the entire guide before starting** as it will help your understand of the overall process.

Guide Dataset

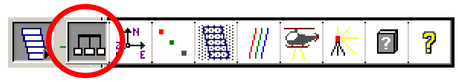
The LiDAR data used throughout this guide is a portion of an Applied Geomatics Research Group (AGRG) LiDAR survey of the Middleton area on Julian Day 332, 2007.

This subset was selected as it provides a clear example of boresight misalignment while still being relatively easy and quick to correct.

Coordinate Systems (for reference only)

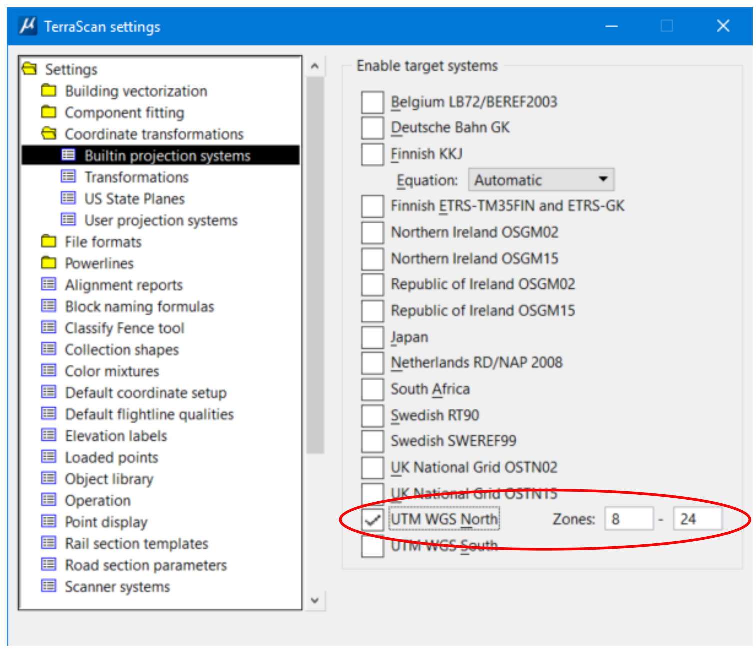
The flightline trajectory data that will be imported into the Middleton project has its coordinates in Geographic NAD83 (CSRS98) while the LiDAR points of the project are in UTM 20 NAD83 (CSRS98). In order to integrate these it will be necessary to project the trajectory positions into the UTM 20 coordinate system.

TerraScan offers a wide variety of coordinate systems used around the world, including Universal Transverse Mercator (UTM). However, before you can make use of this projection you must specify which zone(s) you are working within.



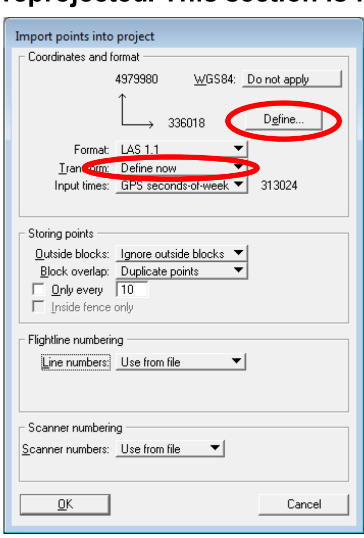
Begin by selecting the *Settings* tool (circled top right) from the main TerraScan toolbar. Within the **Settings** window (right) you will see a variety of configuration categories including f*ile formats* (e.g. importing and exporting ASCII data) and *Coordinate transformations*.

Select the latter and then select the *Built in projection systems* sub-category to open the **Enable target systems** panel (right). Click on the `UTM WGS North` coordinate system and enter the zone(s) you wish to use (e.g. 19 to 20). You can include more zones than those we will use for this lab if you wish.



Reprojecting Points (for reference only. not required.)

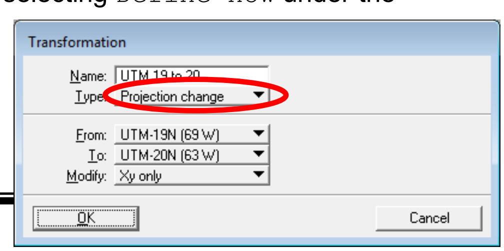
Recall from Lab 1 that the Lawrencetown LiDAR points were provided in UTM 19 (the zone the survey started in) but zone 20 may have been more appropriate for the site. **Note: the points for this lab do not have to be reprojected. This section is for reference only.**



After activating which coordinate systems (& zones) you wish to have available (see above), you are then able to reproject points when importing them into your TerraScan project.

Within the *Import points into project* dialog (in the *Project* window under *File*) you can request the points be transformed by selecting `Define now` under the *Transform* option (circled left).

This activates the *Define* button (circled above) which opens the *Transformation* dialog (right).



This tool allows you to apply a variety of spatial transformations (e.g. XYZ shift, rotation, etc.) to your points as they are imported into the project.

In the case of the Lawrencetown points we would set the *Type* of transformation to `Projection change` and then choose the appropriate *From* (UTM 19) and *To* (UTM 20) coordinate systems.
When working with the UTM projection you will want to only *Modify* the X & Y coordinates of the points, but other transformation models (e.g. "3D transform & rotate" for converting from one datum for another) may need to change the Z values of the points as well.

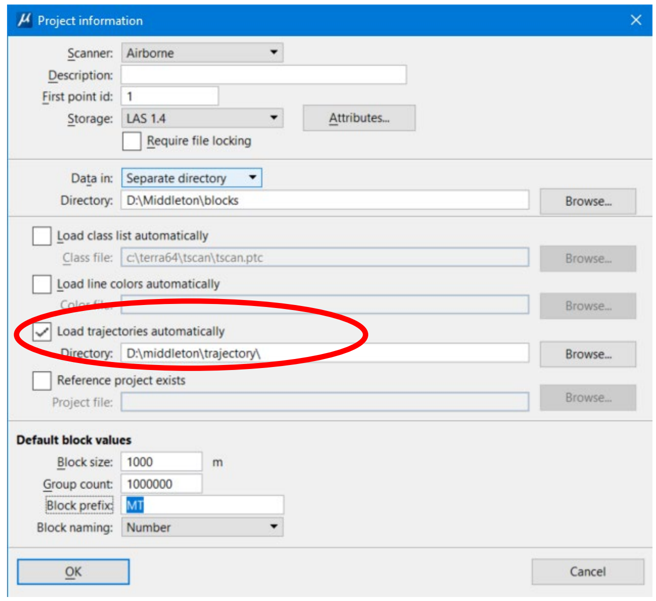
Importing Trajectory Data

The GPS and IMU data from the LiDAR system is typically processed into a smoothed best estimate of trajectory (SBET) that represents the path that the aircraft (more specifically, the position of the scanning mirror) flew during the LiDAR survey. While this data can be provided in a variety of formats, the most common is the "raw" SBET format typically output by the GPS/IMU processing software (e.g. POSPac).

Preparing the TerraScan Project

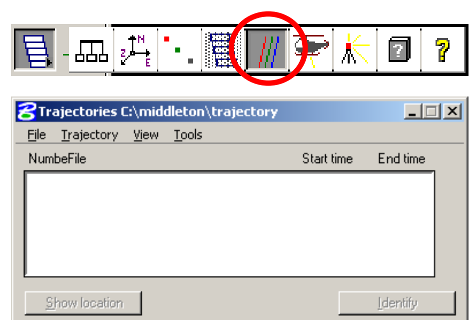
While TerraScan can read SBET files, it actually converts them into an internal format (`.trj`) and splits the SBET into individual files for each flightline. Before working with trajectories in your TerraScan project you must first specify where to place these files.

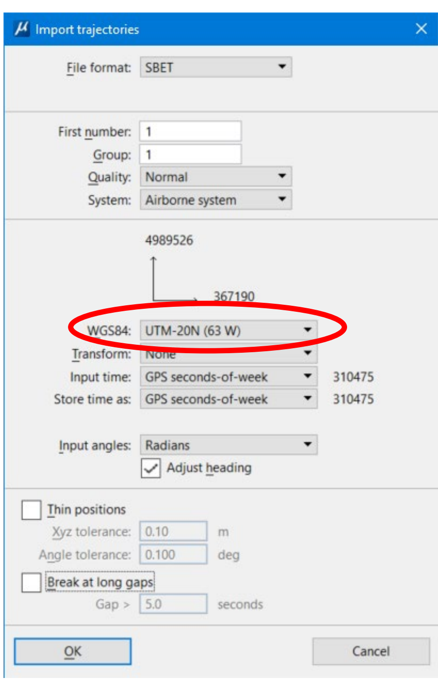
Open the *Project Information* window (*File / Edit project information*) and turn on the *Load trajectories automatically* option (circled right). Next, specify the directory (which must exist) to store (and retrieve) the `.trj` files.



Importing the SBET

To import the trajectory data into your TerraScan project, select the *Manage Trajectories* tool (circled top right) from the main TerraScan toolbar to open the **Trajectories** window (right).





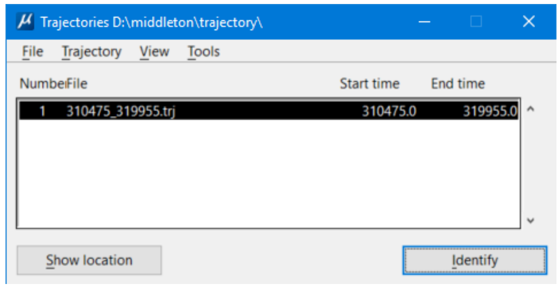
Next, select *Import files* from the *File* menu and select the SBET file for your LiDAR project (e.g. `sbet_01.out`). Upon selecting this file you will be presented with the **Import trajectories** window (left below) requesting additional information on how to import the data.

Since the SBET data is currently in Geographic coordinates, change the dropdown (circled left) to the `UTM 20N` coordinate system you setup earlier.

Next, specify that the *Input angles* are in *Radians* (typical for SBET) and tell TerraScan to *Adjust the heading* values to account for the IMU heading vs. Grid North (i.e. meridian convergence for UTM).

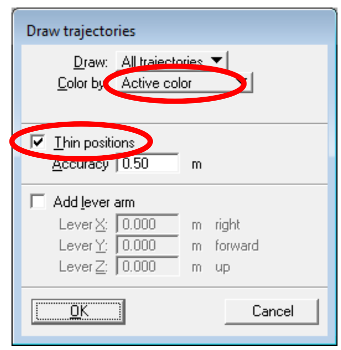
Since you will be using this data to correct the LiDAR points, be sure to turn off the *Thin positions* option (every trajectory position is needed).

Pressing OK, loads the trajectory data and displays it in the **Trajectories** window (below).



Displaying Trajectory Data

Once the SBET file has been imported, the flight path for the entire LiDAR survey is added as a single trajectory (right above). It can be helpful to view the data as a whole before breaking it into its separate flightlines (discussed in the next section).



Use the *Level Manager* (see *TerraScan Guide 1*) to create a new Microstation level called `SBET`, then make this level active. To add the entire flight path to your DGN project, select *Tools / Draw into design* (in the *Trajectories* window) to open the *Draw trajectories* window (right).
Set the tool to color the trajectory data using the currently selected color (`Active color`). To improve drawing speed when they are displayed, thin the trajectory points to a spacing of 0.5 m (this is for display only and does not affect processing).

The *Add lever arm* option allows you to adjust the trajectory positions to account for the spatial distance between those coordinates (e.g. IMU reference) and those of the laser scanner (e.g. scanning mirror).

In this case, the POSPac and DashMap software packages used for processing the AGRG LiDAR data accounted for these lever arms so this option can be disabled or set to 0's.

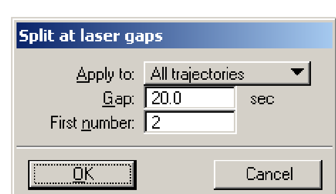
Creating Flightlines from Trajectories

In order for TerraMatch to identify and correct any alignment errors in the various LiDAR sub-systems (see *Flightline Mis-Alignment* later in this guide), the next task is to delineate the individual flightlines from the trajectory data loaded earlier.

Two different approaches are described in the sections below, the first uses the LiDAR points to automatically determine the flightlines; the second allows you to manually break the trajectory data using polygons.

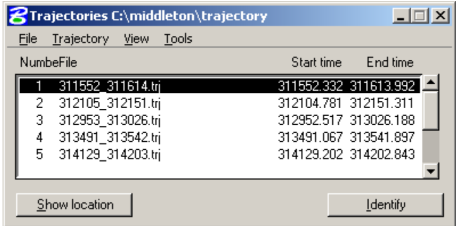
Splitting Trajectories Automatically (not recommended for this lab)

The first method makes use of the LiDAR points loaded within the TerraScan project blocks. Each LiDAR point has a time-stamp that can be related to those in the trajectory data.



The trajectory points can then be filtered according to the existence of LiDAR points; that is keep only trajectory points for where LiDAR data has been collected (i.e. the active swath of the system).

Select *Split at laser gaps* under the *Tools* menu in the *Trajectories* window to open the **Split at laser gaps** window (right).

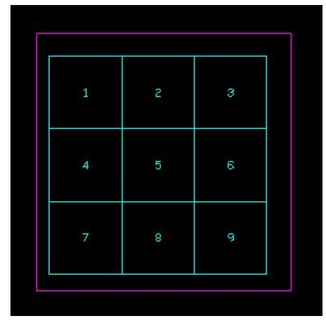


The start/end of each flightline is determined by discovering “gaps” (in time) within the LiDAR points. Generally this *Gap* value should be quite small; only a few 10’s of seconds (e.g. 20 in this example) to ensure that trajectory data is not unintentionally removed.

Upon completion, the **Trajectories** window will be updated to show the derived flightlines (left).

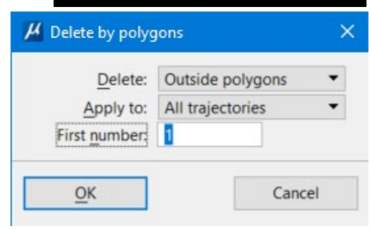
Splitting Trajectories Manually

The second method of delineating flightlines from the original trajectory data is to create one or more polygons (i.e. Microstation elements) and use them to “clip” the imported trajectory data into separate lines.



Generally you will want to draw the polygons (using the *Place Shape* tool) such that they clip away the “turn around” portions of the flight path (e.g. the loops at the end of each flightline). Depending on the complexity of the flightline plan this may be a single polygon or multiple polygons.

In the case of the example for this guide, a single polygon can be created to surround all of the project’s blocks (pink or outside polygon right).



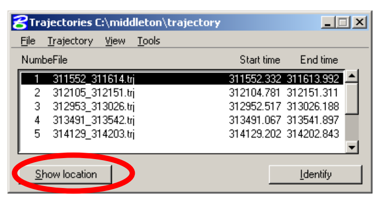
Once the clip polygon has been created, select it and run the *Delete outside polygons* tool from the *Tools* menu (right). This will remove all trajectory points outside of those polygons and then deduce individual flightlines from the remaining points.

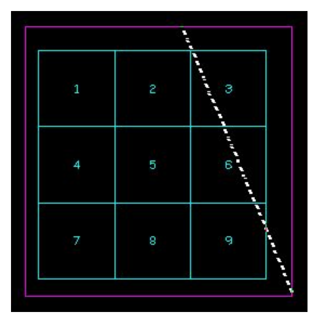
Viewing & Editing Trajectories

After splitting the initial trajectory data into individual flightlines, you are then able to manipulate them separately. The following sections describe ways to identify, display and delete these new trajectories.

Showing Flightlines

To display a particular a derived flightline, select its corresponding trajectory in the *Trajectories* window (e.g. #1 shown right) and then press the *Show location* button (circled right).

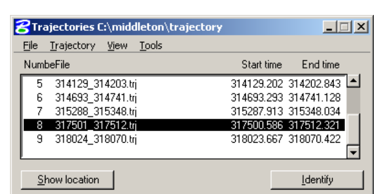




Once clicked, moving the cursor into *View 1* (containing your block outlines & trajectories) will display the selected flightline (right). Right-clicking in *View 1* ends this “show” mode.

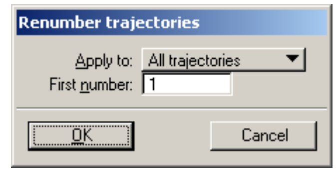
Deleting Flightlines

Situations may arise where we may wish to delete trajectories such as our full uncut trajectory. Or perhaps a calibration flight was performed over the project area and the calibration flightlines and data should not be included in our TerraScan project.



In this case, we may wish to remove the flightline(s) (i.e. its trajectory data) by selecting the trajectory, and using *Delete* under the *Trajectory* menu and allow the remaining flightlines to cover the study area.

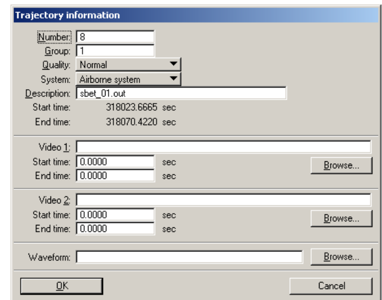
Renumbering Flightlines



When a flightline is deleted from the trajectory window, the remaining flightlines are not automatically renumbered and thus need to be manually adjusted.

The *Renumber trajectories* option (right) under the *Tools* menu allows you to quick renumber all or some (just those selected) of the trajectories. The new sequence begins at *First number* and increments for each following trajectory.

Flightline Information



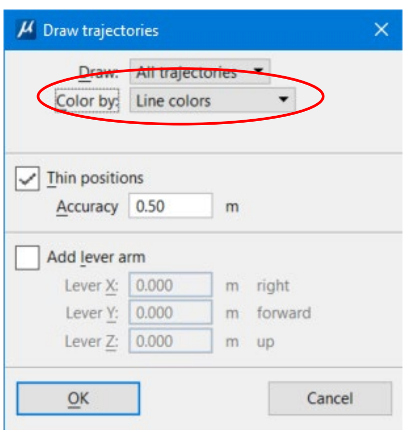
Detailed information on each flightline can be viewed & changed by selecting the trajectory of interest and then opening the *Trajectory Information* window (right) through *Edit information* (under the *Trajectory* menu).

The trajectory *Number* can be changed as well as the *Quality* of the trajectory data (`Excellent`, `Good`, `Normal`, `Bad`, etc.). This quality rating can be used while matching various flightlines to each other during the TerraMatch process (e.g. low quality trajectories are given less weight during calculations).

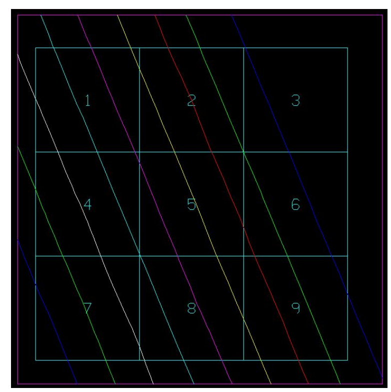
Displaying Flightlines

Once you've extracted the separate flightlines from the trajectory data (and performed any editing necessary) you may add them to your Microstation DGN using the *Draw into design* tool under the *Tools* menu in the *Trajectory* window (similar to you the adding entire trajectory data earlier).

Use the *Level Manager* to create a new Microstation level called `Trajectories`, then select this as the active level.



Unlike before when all of the trajectory points were added to the project, this time the points should be colored using `line colors`(circled left).



This will give each flightline a unique color (right).

As before, *Thin* the points (e.g. 0.5 m) to improve drawing speed (this is for display only and does not affect processing).

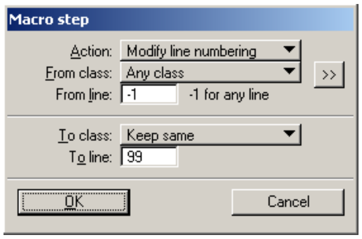
Assigning Points to Trajectories

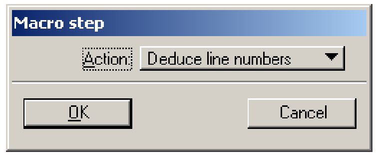
After individual flightlines have been derived from the full trajectory data, you may now apply this information to each LiDAR point. The *Deduce line numbers* tool within TerraScan uses the time-stamp of each LiDAR point in the project blocks and relates it with those from the individual flightline trajectory points; thus each LiDAR point is assigned which flightline it was collected in.

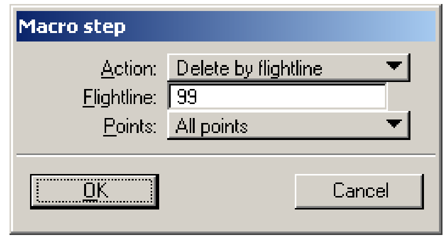
However, there may be points within the blocks that cannot be related to the trajectories you extracted earlier (e.g. flightline 8 that was deleted). In this case the *Delete by flightline* tool may be used to remove any "unassigned" points.

A macro can be created to perform the following tasks:

- Set the flightline number to 99 for all points (regardless of its current flightline (-1) or classification).
- Deduce flightlines using the trajectories and point time-stamps
- Delete any points that retain flightline 99 (i.e. they were not relabeled in the step above)

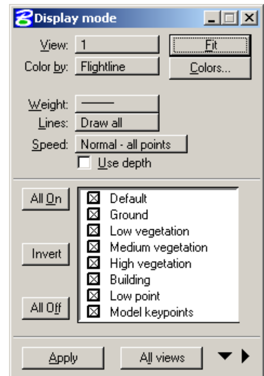


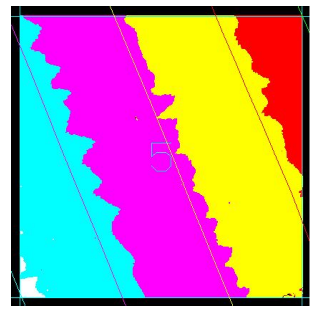




Save this macro as `deduce_flightlines.mac` and then use the *Run macro* tool (*Tools* menu in the *Project* window) to apply it to all the blocks in your TerraScan project. Once it has completed, save the resulting report as `deduce_flightlines.txt` for future reference.

Finally, verify that the flightlines have been assigned correctly by loading in one or more of your project blocks and displaying the points by flightline (right).





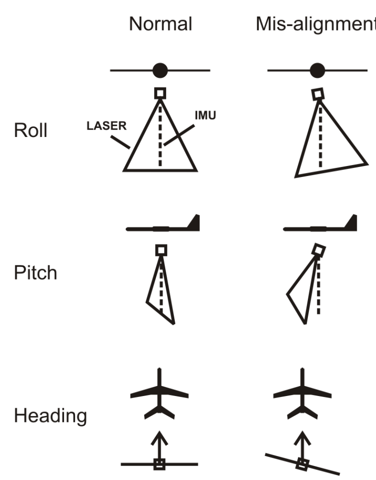
Analyzing Flightline Misalignment

Once the trajectory data has been loaded, extracted into individual flightlines and the LiDAR points assigned to their flightline numbers you are now ready to analyze your project data to determine which misalignment errors exist. There are several types of errors that TerraMatch can correct for, but only systematic roll, pitch and heading are discussed in this guide.

Laser Head vs. IMU

While the lever-arm offsets between the GPS & IMU reference frames can be directly measured and refined during post-processing (e.g. POSPac), the angular alignment between the IMU & the laser head is more difficult and prone to error.

Any amount of mis-alignment between those two sub-systems will present itself as spatial offsets between the LiDAR points of opposing flightline swaths. By analyzing the point cloud one can determine the broad trends of mis-alignment and then use tools such as TerraMatch to more precisely measure and remove that error.



Using Profiles to Determine Misalignment

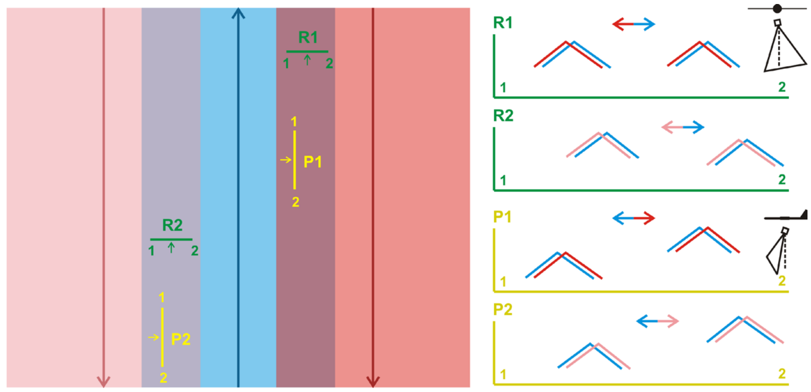
Profile views are an easy way to observe these issues, particularly pitch and roll errors.

When a profile is drawn *across-track* where two flightlines overlap (e.g. R1 and R2 below) then a *roll error* can be observed as a horizontal “shift” in the points between those flightlines.

When a profile is drawn *along-track* where two flightlines overlap (e.g. P1 and P2 right) then a *pitch error* can be observed as a horizontal shift.

Note that both roll and pitch appear to have similar patterns in the profile views, it is the **orientation of the profile** line that makes the distinction.

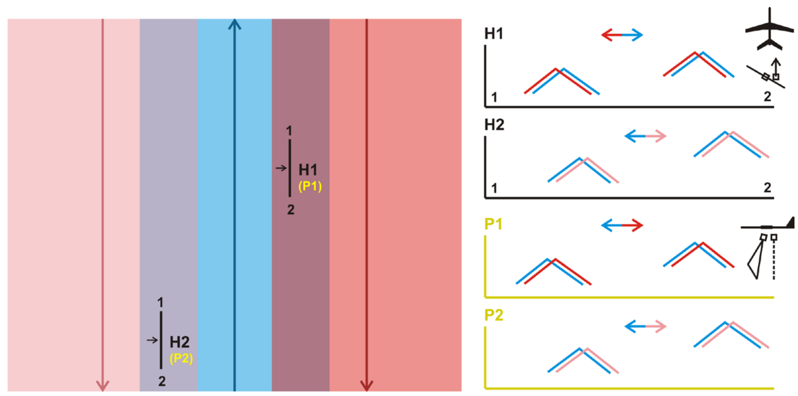
Unfortunately, heading errors are much more difficult to determine as they can appear to be pitch errors.



However, by comparing along-track profiles (i.e. looking for pitch errors), a heading error may appear as alternating "pitch" errors.

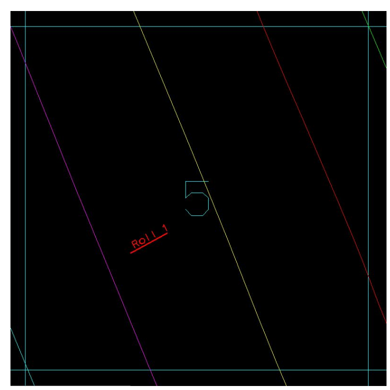
That is, with one pair of flightlines the pitch will appear to be forward (e.g. H2 vs. P2), while in the neighbouring pair it will appear to be afterward (e.g. H1).

Combinations of roll, pitch and heading can become very complex, but if a methodical approach to observing errors across multiple flightlines is used, these patterns may become more apparent.



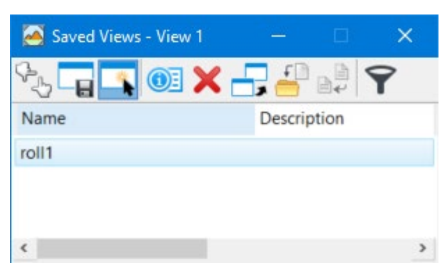
Analyzing Project Data

Along-track and across-track profiles should be made over several project blocks (not just within your "calibration" regions as discussed later) and your observations recorded (e.g. screenshots of profiles, possibly). There should be at least two along-track and two along-track profiles drawn into DGN.



Use Microstation elements and labels (e.g. `Roll 1`) to denote the locations viewed (right) as it will be helpful to reviewing these areas again once corrections have been determined and applied. You may also wish to make use of one or more Microstation levels (e.g. `Alignment Checks`) to help organize those various elements.

In addition, the *Saved Views* tool (right) in Microstation (under *View* menu, *Saved Views* pane) can be used to record a view's current settings (e.g. profile location, depth, etc.). This allows you to return to that exact location (e.g. roll error observed in sets of buildings) at a later time (e.g. after corrections have been applied).



This tool works very similar to "bookmarks" within ArcMap in that you choose a *View* to save and then provide a *Name* to store it under (e.g. `Roll 1`). To return to that location, choose which *View* to update (e.g. `3` for the profile), select which stored settings to restore and then click *Apply*.

Finally, consider using the *Measure Distance* tools (under the *Analyze* tab in the *Measure* pane) to quantify the amount of "shift" observed between overlapping flightlines.

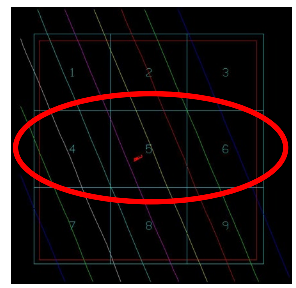
Preparing a Calibration Dataset

TerraMatch searches a provided set of LiDAR points and compares landscape patterns such as slopes and hard edges (e.g. buildings) across the various flightlines. Although the software is quite capable, it cannot effectively process large volumes of unclassified LiDAR points.

Since the goal is to identify and determine the boresight angles between the IMU and laser scanner you don't need to analyze the entire survey dataset but rather only a representative subset (i.e. **boresight is a systematic error, it is "the same everywhere" in the survey**).

The following sections describe the steps required to create and prepare a calibration dataset for analysis in TerraMatch. Many of the TerraMatch processing steps will be performed on this subset of the data (referred to as "calibration" blocks or project in this guide) rather than your original project blocks (referred to as "survey" blocks or project).

Calibration Blocks



One approach to creating a calibration dataset for your project is to simply choose a few representative blocks from the project.

These blocks should contain multiple flightlines (and crossing flightlines if available) as well as clearly defined & hard sloped surfaces such as hills and building rooftops. Try to avoid tiles that contain large sections of forested areas and flat terrain.
For example, blocks `4`, `5` and `6` (right) could be selected as they collectively span all eight of the project flightlines.

Create the calibration dataset by making a new folder under your project folder (e.g. `calibration`) and copying the LAS files of the selected blocks into this folder.

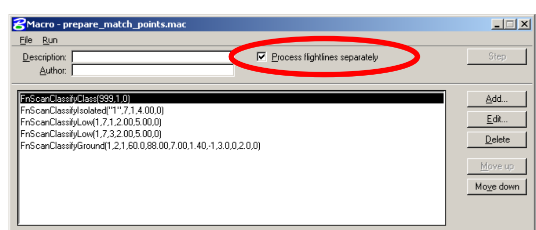
Once copied, you can either create a new TerraScan project (& related Microstation elements) to refer to these blocks or **make use of TerraScan's ability to dynamically switch the block folder it refers to during processing (see *Managing Multiple Data Folders* later in this guide).**

Classify Calibration Points

TerraMatch works best when there are few (if any) "random-like" points in the data it's attempting to match. This includes erroneous points (e.g. isolated) as well as most non-ground points such as forest canopy.

Buildings are an exception and if several are available throughout your project you may wish to classify those explicitly (often better done manually rather than through a macro) and provide them to TerraMatch when running the *FindMatch* tool (discussed later).

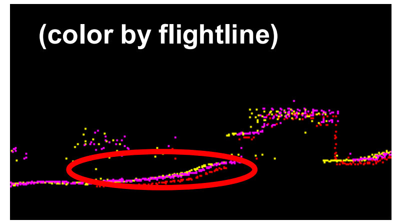
A relatively simple macro can be created to prepare the points within your calibration project. This macro should identify erroneous points, ground points and (optionally) buildings.



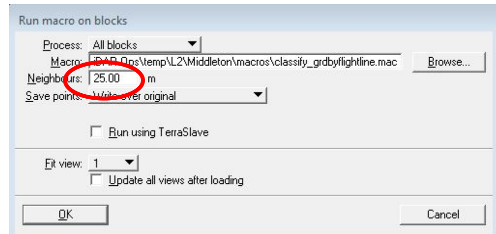
- Classify all points into the `Default` class
- Classify isolated points into the `Low point` class
- Classify single low points into the `Low point` class
- Classify groups of low points into the `Low point` class
- Classify ground points into the `Ground` class
- (optionally) Classify points defining buildings into the `Building` class

Although each survey and region may require slightly different parameters for each of these algorithms (or possibly entirely different algorithms), those described in your previous lab can be used as a starting point.

Before saving the macro, turn on the *Process flightlines separately* checkbox (circled above right).



This tells TerraScan to process each flightline independently allowing each ‘ground’ surface to exist independently (e.g. the ‘lowest’ points in one flightline may be higher than those of another). This is key to comparing the flightlines (i.e. vertical differences between surfaces) and determining the calibration parameters.
Save this macro in your `macros` folder as `classify_grdbyflightline.mac`. Before we run this macro, **remember to set the neighbors parameter to 25m** within the run macro window (circled right).



This will automatically load points from the neighboring blocks for each block the macro is working on. Run this macro on all of the blocks within the project and save the resulting report as `calibration_classify.txt`.

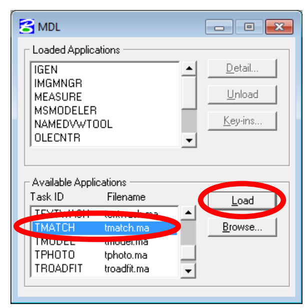
Finally, verify the results visually by loading and displaying points by class (right). If the classification was applied to flightlines separately then you should see multiple “layers” of ground points. If not, the ground classification would have focused on just the lowest points along this slope and TerraMatch would not recognize the others as belonging to different flightlines.

Analyzing a Calibration Dataset

Now that you’ve selected a subset of points (selected blocks or region polygons) to analyze and prepared them for comparison (e.g. per flightline classification) you can now start working with TerraMatch to determine the systematic corrections necessary to remove the errors observed earlier.

TerraMatch

Like TerraScan, TerraMatch is an MDL application that must be loaded from within Microstation (*Utilities / MDL Applications,* `TMATCH`; shown right).



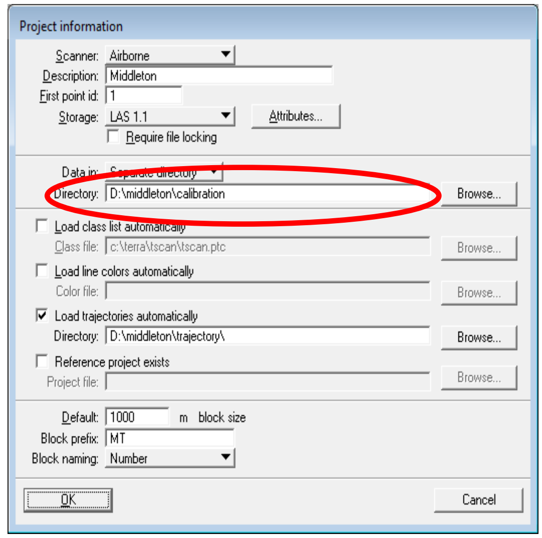
As discussed in class, TerraMatch offers two approaches for determining calibration parameters: surface matching, which compares TINs of overlapping flightlines; and tie-line matching, which compares “edge-like” features such as roof tops.

This guide focuses on the surface matching technique but many of the steps are similar for the tie-line approach. There are three TerraMatch tools that are commonly used during the surface matching workflow:

> *Measure Match*
> Compares the elevations (via TINs) across each flightline of your project to determine an overall amount of “error” (differences in Z). This can be used as a measure of how well the various error model parameters (i.e. derived boresight angles) perform. Upon completion of this process this value should be significantly lower than originally measured.
>
> *Find Match*
> Compares the various flightlines of the project to determine what corrections need to be applied (only systematic corrections are discussed in this guide) to improve the alignment.
>
> *Apply Corrections*
> Takes the results of the *Find Match* tool and applies it to your project blocks. This will typically result in points “falling outside” of their original block boundaries so additional processing (described later) is often required to address this.

Managing Multiple Data Folders

Normally a TerraScan project is set up to read its block data from a single folder (you state this when creating the project). However, this folder may be changed at any time to allow you to temporarily work with a slightly different variation of your data (e.g. a previous classification or various outputs from your calibration blocks).

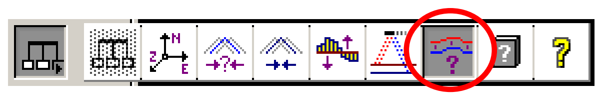


For example, if you had chosen to select a subset of survey blocks rather than creating custom region polygons for your calibration dataset you could simply use the existing TerraScan project (instead of creating a new one) by re-directing the project to the copied blocks (LAS files under the `calibration` folder).

Select the *Edit project information* option from the *File* menu in the TerraScan *Project* window to open the *Project Information* window (right). Here you may change the *Data in Directory* field (circled right) by clicking on *Browse* and selecting the new folder.

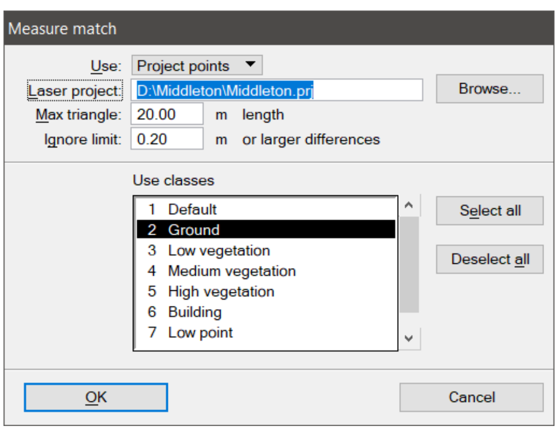
Note, whenever you change the data folder be sure to **immediately save your TerraScan project** after closing this window. Otherwise many of the TerraScan and TerraMatch routines will not recognize the change.

Measuring "Starting" Alignment



To provide a basis of comparison, begin by comparing the various flightlines to determine the overall amount of mismatch using the *Measure Match* tool (circled right).

Measure Match (below) creates Triangulated Irregular Networks (TINs) from the points of each flightline separately (which is why they needed to be classified individually). It then overlays the TINs and computes the differences in Z at multiple locations and summarizes the results.

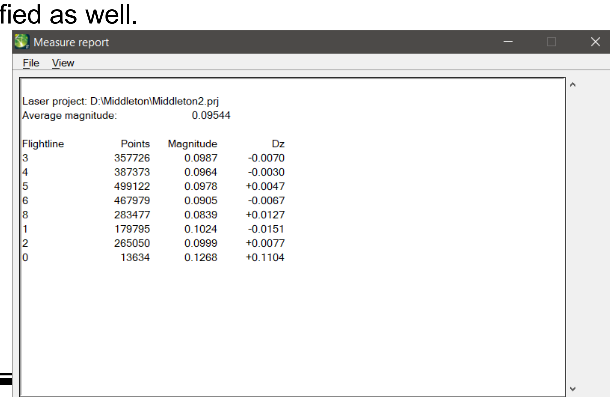


Set the *Laser project* to your TerraScan calibration project (referring to your selected blocks or region-based blocks).

The *Max triangle* setting tells *Measure Match* to not create a TIN triangle with an edge larger than that value; essentially to avoid creating triangles across large gaps in the data (e.g. forested areas).

The *Ignore limit* says to ignore any Z difference between flightlines larger than this value. This can be helpful in filtering out comparisons between buildings, remaining vegetation, etc. and "true ground" of another flightline.

Finally, select which input classes you wish to compare. This should be at least the ground points classified earlier and possibly any building points you may have classified as well.



Once *Measure Match* has completed (this will take a few minutes for the example dataset), review the report it generates (right) and save the report in your `reports` folder as `calibration_starting.txt`.

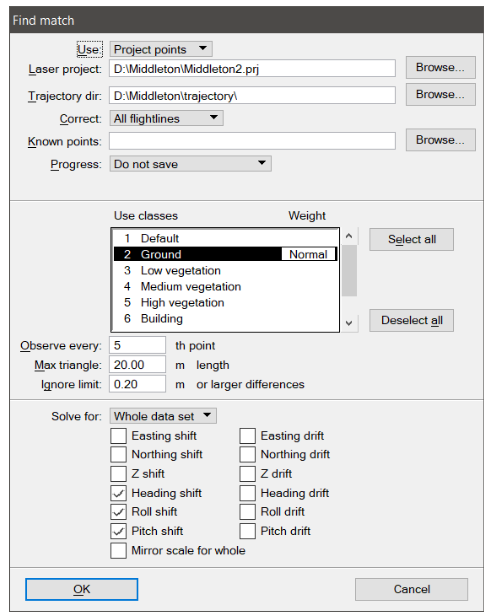
The summary report provides a magnitude and delta shift in Z for each flightline as well as an overall average magnitude. The *Magnitude* values represent the average absolute difference between that flightline and any other flightline that it overlaps.

The *delta shift* (Dz) represents an overall “bias” for that flightline; that is if the flightline was bulk shifted in the Z by this amount, it will tend to better align with its neighbours. For now, take note of the *Average magnitude* value as you will compare this to your final corrected points.

Determine Corrections

The primary purpose of TerraMatch is to determine the adjustments (systematic in this case) necessary to account for the IMU/laser mis-alignment (boresight angles).

The *Find Match* tool (circled right and shown below) is used to perform this task.



As with the *Measure Match* tool, set the *Laser project* to your TerraScan calibration project. The *Trajectory directory* should already be set to where you created your trajectories.

Similarly, select which input classes you wish to be included during the flightline analysis; in this case the ground class, but you may also include buildings (if classified carefully).

When you have multiple classes, you may wish to assign weightings to each class. This could be used to tell *Find Match* to pay more attention to the hard edges of buildings than the more uncertain ground surface. Click beside the class label under the *Weight* column to set/change these weights (`Low`, `Normal`, `High`).

Similar to *Measure Match*, during its analysis *Find Match* creates several TINs (one for each flightline) and uses these for its geometric comparisons and calculations. The *Max triangle* and *Ignore limit* parameters apply as with *Measure Match* described earlier.

The *Observe every* point setting allows *Find Match* to reduce the number of points it works with and generally can speed up the comparison process. However, be sure not to set this too high as it may begin to lose important details in the terrain and will be unable to compute adjustments as accurately.

Finally, tell *Find Match* to solve for the `Whole data set` (i.e. a single global adjustment - a systematic correction) and then check off which corrections you wish it to determine. This last step is critical as the more corrections you request the more complex the processing becomes. This can significantly increase processing time and may actually result in less effective corrections if they are not necessary. In this case, with a small dataset, it’s not an issue. Tell *Find Match* to solve for `heading, roll, and pitch shift`.

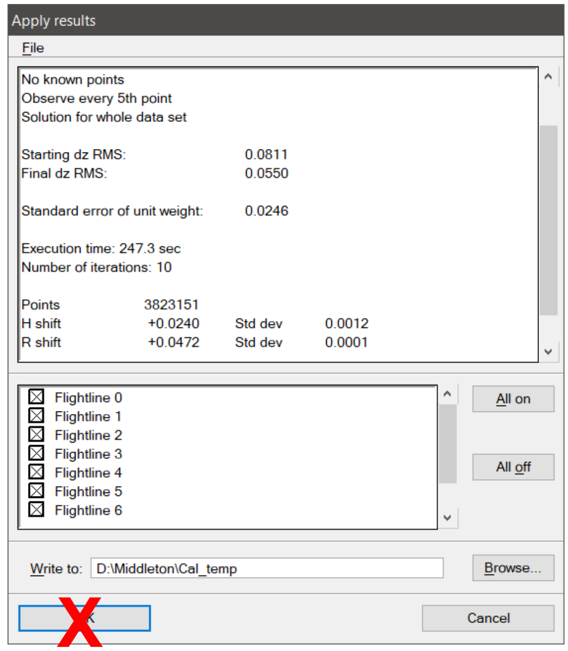
Reviewing Find Match Results

Once *Find Match* completes its analysis it will present a summary report of the corrections it determined and their effect on the dataset (e.g. average magnitude and standard deviation) within the **Apply results** window (right).

Take note of the *Starting dZ* and *Final dZ* values shown in the report. The final values should be notably less (several cm) as this indicates that the corrections determined effectively correct the data. If not, it may be necessary to re-select, process, etc. the calibration data and attempt another *Find Match* round.

It is **critical at this point** to save the corrections to your `reports` folder as `calibration_match.tms` (*File / Save corrections* under the *Apply results* window) as this needs to be applied to your entire project later. You should also save the summary report as `calibration_match.txt` (*File / Save report*).

Do not press the OK button yet, see the next section below (Applying Corrections).



Wait! (see next section)

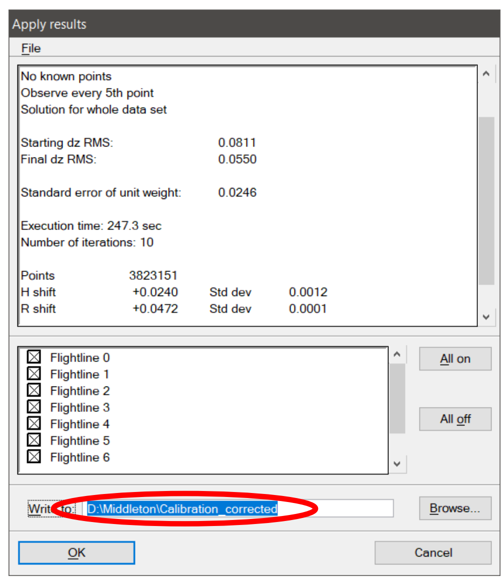
Applying Corrections

If you are satisfied with the *Find Match* results, you should next apply the corrections to your calibration blocks. This allows you to quickly view and assess the corrections visually (e.g. profiles) before they are applied to your entire survey project.

Rather than apply the corrections to your calibration blocks directly (in case the results are not as expected) instead change *Write to* field (circled right) to an output folder called `calibration_corrected` (you must create this folder).

This tells TerraMatch to apply the corrections and write the modified LAS files out to the specified folder rather than overwriting your original calibration blocks.

Note, no summary window will be presented when it is finished; instead TerraScan/Microstation will simply return control to the user.

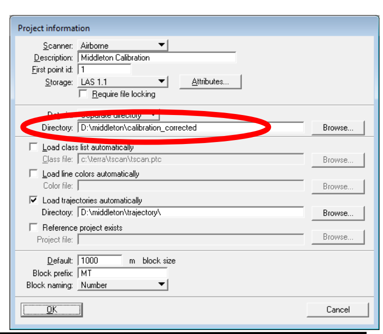


Review Corrections

Before applying the corrections to your entire survey project, you should first verify that the flightlines of the corrected calibration blocks do actually align better than before.

Begin by changing your TerraScan project folder to `calibration_corrected` (see *Managing Multiple Data Folders* earlier, be sure to save the project file). This will tell TerraScan to read points from that location rather than your original calibration blocks as before.

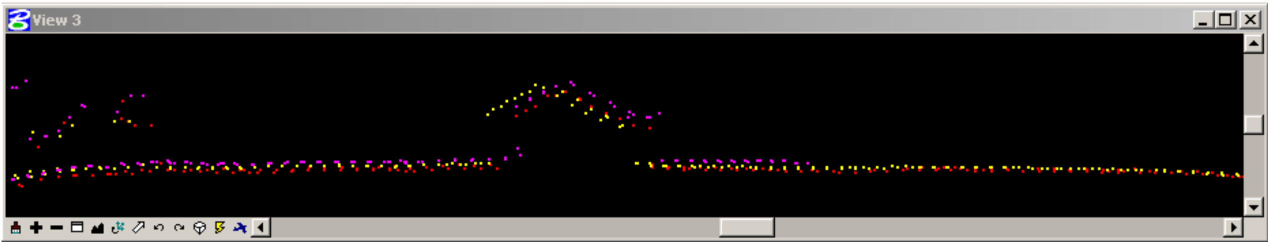
Next, run *Measure Match* again using your corrected calibration blocks and save the result in your reports folder as `calibration_final.txt`. Compare the average magnitude values from this report to those from the earlier report



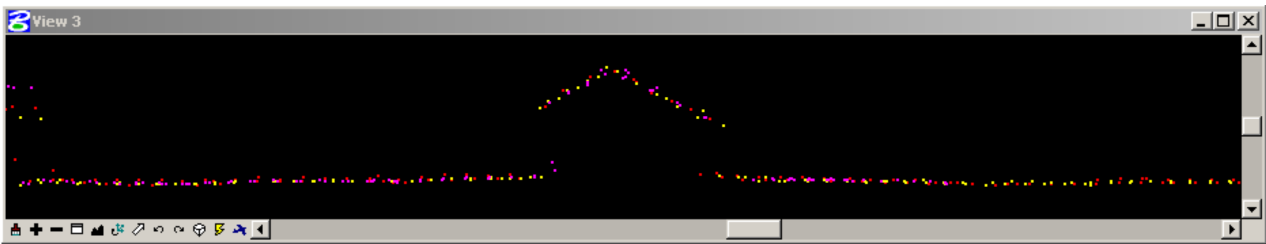
(`calibration_starting.txt`). The new values should be notable less (several cm) which indicate a good final adjustment.

Finally, return to your various "check" sites created earlier (see *Analyzing Flightline Misalignment*) and compare them with the newly corrected LiDAR points. For example:

Original



Corrected

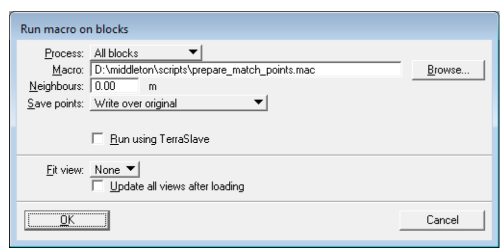


Applying Corrections to the Survey

After verifying that the *Find Match* adjustments do effectively correct the LiDAR points (at least for the calibration blocks) you are then ready to apply the corrections to your entire survey project.

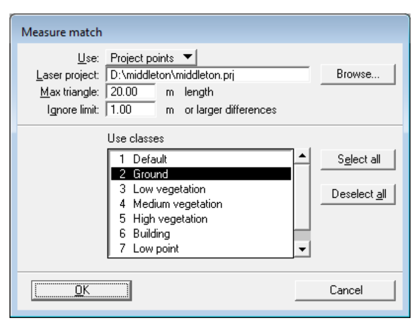
Analyzing Survey Blocks

As with the calibration blocks, we should create a basis of comparison by running the *Measure Match* tool on the original survey blocks. However, before *Measure Match* can be used, the survey blocks need to be classified by flightline as done with the calibration blocks (see the *Classify Calibration Points* section earlier).



Fortunately, you can simply run the `classify_grdbyflightline.mac` macro created while preparing the calibration dataset (remember that the *Process flightlines separately* option needs to be turned on) on your survey project.

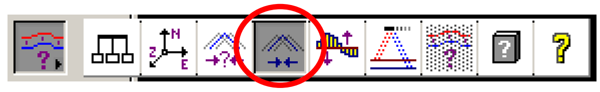
Make sure the survey project loaded in the TerraScan project window (and not the calibration project used earlier) before running the macro and save the classification report as `survey_classify.txt`.



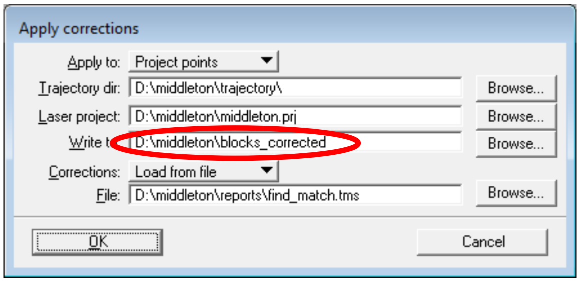
Once the survey blocks have been classified, run the *Measure Match* tool and select the survey project for analysis (right). As before, specify at least the ground class to be used during this comparison and any triangle size and Z difference limits you wish to apply.

Note that this will take more time as you are processing all of your survey blocks, not just your calibration blocks as before. Once completed, save the results into your `reports` folder as `survey_starting.txt`.

Correcting Survey Blocks



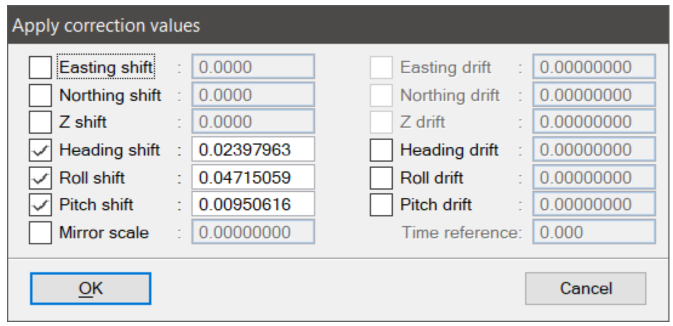
The corrections determined from *Find Match* can be applied to all of the survey blocks through the *Apply corrections* tool (circled right) in the TerraMatch toolbar.



Apply the corrections to your *Project points*, then set the *Laser project* to your survey project file. Rather than overwrite your original points, change the *Write to* parameter (using *Browse*) to use a new `blocks_corrected` folder.

Finally, set your *Corrections* to be `Load from file` and then select your `calibration_match.tms` file that you saved earlier from *Find Match*.

Upon clicking the OK button you will be presented a summary of the correction file you selected (right). Verify that you are about to apply the correct values and then press OK to begin the process.



Note: Applying corrections to your entire project is a long process. Although the example dataset for this guide this will only take ~5 minutes or less, this process can be expected to take several hours to complete for larger surveys.
Once completed, you will be presented with a summary report for the processing performed. Save this into your `reports` folder as `survey_corrections.txt`.

Finally, to verify that the corrections have applied correctly, run *Measure Match* on the newly created blocks (remember to change your TerraScan project folder to `blocks_corrected`). Save this report as `survey_final.txt` into your `reports` folder

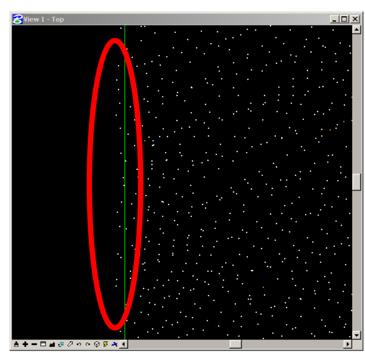
Compare these values to those from the original block data (e.g. `survey_starting.txt`). As before, the average magnitude should be notably smaller (several cm).

```
Laser project: D:\middleton\middleton.prj
Average magnitude:            0.27093

Laser project: D:\middleton\middleton.prj
Average magnitude:            0.08667
```

Similar to checking the calibration blocks, you should also revisit your various “check” sites to ensure that they appear improved (i.e. compare `blocks` to `blocks_corrected` by switching the TerraScan project folder).

Reorganizing Survey Blocks



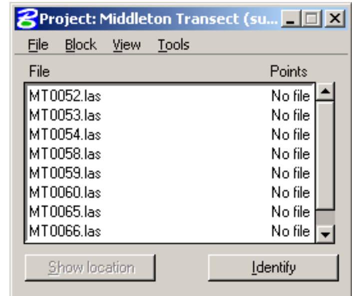
Take note that once you apply the TerraMatch corrections to your survey project blocks that several points now fall “outside” of their block boundaries (circled right).

This is expected as the various *Find Match* adjustments will move the points as necessary to address the sensor mis-alignments. To account for this, it will be necessary to reimport all of the project points to reorganize them into their newly respective block extents.

Reimport project points

Change your TerraScan survey project folder to back to your original `blocks` folder, save the project and then close the *Project* window. Then move all of the LAS files within the `blocks` folder into a new folder called `blocks_original`. This step is primarily meant as a backup in case your need to reapply any of your corrections.

Upon reopening your TerraScan project you should see all of the blocks be listed as “No file” (below). This is normal since the `blocks` folder no longer contains any LAS files.



Next, use the *Import points into project* tool (under the *File* menu of the TerraScan project window) to import all of the LAS files within your `blocks_corrected` folder into the (now empty) `blocks` folder (similar to importing flightlines into manually created overlapping blocks in *Guide 1*).

Note, since we deduced flight line numbers from the trajectory data earlier, select the `Use From File` option for *Flightline numbering* to preserve those values during importing. Finally, be sure to save the resulting summary report as `survey_reimport.txt` into your `reports` folder.

Elevation Adjustments

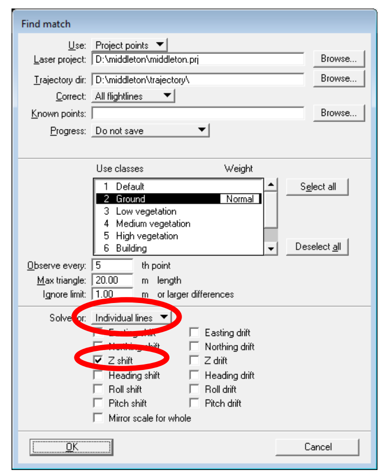
While the boresight calibration often removes a significant amount of error from the LiDAR points the individual flightline trajectories can still be affected by semi-random GPS inaccuracies.

This often leads to flightlines having a slight vertical offset (even if no pitch or roll is apparent). Fortunately, this can be corrected by using TerraMatch to identify and apply “z shift” values to each flightline.

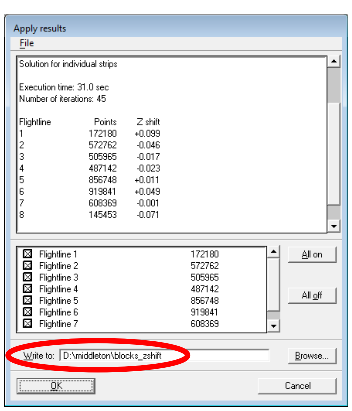
Determine & Apply Corrections

Similar to the boresight calibration process, the *Find Match* tool (right) is used to determine the z-shifts for each flightline in the survey project (not just the calibration blocks).

Since each flightline is to be adjusted separately solve for `Individual lines` and then check on the `Z shift` correction.



Find Match performs this task relatively quickly and presents the results as a list of z-shift adjustments (below). Review these to decide whether they are significant (e.g. more than 1 or 2 cm). Save this report as `survey_zshift.txt` into your `reports` folder.



Clicking *OK* in the *Apply Results* window will have the stated adjustments applied to each respective flightline.

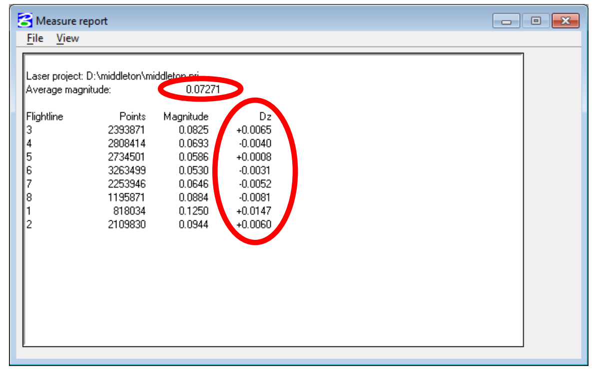
This process also runs quickly and once completed you should review the Z-shift values for each flightline to gain an appreciation of the corrections applied (typically only a few cm, much larger indicates a more serious sensor or GPS/IMU processing error that should be investigated).

You may choose to overwrite your survey blocks or output the z-shifted blocks to another folder through the *Write to* option (circled left). In that case, remember that you will need to change the data folder in the TerraScan project before viewing/working with those files.

Unlike when the boresight adjustments were applied, the z-shift values do not affect the horizontal position (Easting & Northing) of the points. This means that you will not need to re-import and remove duplicated points as done after applying those corrections.

Analyzing Survey Blocks

After applying the z-shift corrections, you should run *Measure Match* again to assess their effectiveness. Be sure to set the survey project folder to the correct location (i.e. you output the blocks to a new folder in the previous section).



Compare the final *Average magnitude* value to that from after applying the boresight calibration (e.g. 0.08667), it should improve slightly (often less than 2 cm).

Also review the new *Dz* values which indicate the remaining vertical mis-alignment between the flightlines. Ideally these should all be close to or less than 1 cm as the z-shift corrections you applied should account for most of their original differences.

Any large Dz values here indicate a more significant issue, likely due to GPS/IMU processing and should be investigated further. Save this report as `survey_zshift_final.txt`.