

# BayesStripAlign

*Align your data. Fast.*

Fast, robust, adaptive airborne LiDAR strip adjustment  
and sensor calibration with integrated quality control

---

## QUICK DOCUMENTATION AS OF 10 / 6 / 2016

---

### BayesStripAlign 1.3 (latest build) with LASlib 950605\*

(c) 2015-2016 BayesMap Solutions, LLC - [bayesmap.com](http://bayesmap.com)  
Questions and bug reports: email [support@bayesmap.com](mailto:support@bayesmap.com)

#### The use of this software subject to licensing:

- New users: visit [bayesmap.com/licensing](http://bayesmap.com/licensing) for information and to request a demo license.
- Existing users: please check your EULA for conditions and restrictions.

\* Based on LASLib V 150605 - LASlib with LASzip (c) 2007-2015 martin.isenburg@rapidlasso.com  
Latest version: <http://www.lastools.org/download/LAStools.zip>

---

## What is BayesStripAlign and what is it for?

BayesStripAlign is an automatic strip alignment tool: it registers overlapping LiDAR swaths and corrects both relative and absolute geometric errors. It allows to effectively reduce discrepancies between strips due to IMU attitude and position errors, and to combine corrections optimally in order to minimize the absolute error. It uses a spatially adaptive approach to address time-dependent effects such as drifts and oscillations, which cannot be corrected with classical sensor calibration.

The algorithms it is based on take advantage of all the last returns, classify and filter out outliers (including vegetation, mismatched building walls, missing and erroneous data); LAS ground classification can be used if available. A roughness threshold can also be used to exclude vegetation and speed up the computation. No planes, or geometric objects of any type required. No control data required either, but if available it can be used.

Swaths to be aligned need to be approximately parallel, but can be flown in any direction, and with any spacing, as long as there is some overlap. The flight lines need to be straight, or comply with selectable tolerance levels. The software will only process the largest group of parallel lines and warn the user and log all rejected lines, which do not comply with the parallel and straight requirements. It is possible to specify a reference flight line to select a group of parallel lines, which is useful when handling multiple groups having different orientations.

Trajectory and orientation data are needed to achieve maximum accuracy, however trajectory segments can also be used if the full position/orientation data are not available. This way, archived data, as long as files are flight lines, can also be corrected.

Industry-standard LAS, LAZ and ASCII files are accepted for input/output. Even shuffled files can be used (where points are not in time order, but reordered by the software used to generate them). Position/orientation files can be provided in Applanix SBET, Riegl POF, Leica SOL, Terrascan TRJ, or text formats. GPS time can be in week seconds, adjusted GPS time, or a mixture of both, and the software will convert to adjusted whenever possible.

Systematic effects, such as internal geometry miscalibration, attitude drifts, and leverarms are automatically computed, before treating all the random effects. Therefore, sensor calibration is not a prerequisite; if done, the software will correct for imperfect calibration.

Classical sensor calibration is also possible, with a cross flight, or at least three lines. There is no constraint on flight line orientation, as intersecting lines at any angle can be used. This allows the user to estimate the boresight angles, the y-leverarm, and the scan angle scale factor. Forward/backward looking scanners can be used (e.g. Riegl Q-1560) and each channel can be calibrated separately, to achieve optimal accuracy during corrections.

We did not forget the users of multichannel instruments, and designed a relative calibration module for inter-channel alignment, as well as a relative correction command. The latter saves time as the full alignment procedure only needs to be run on a single channel, and relative calibration results are then used to correct the other channels. The relative calibration procedure is restricted to across track-looking instruments. Other scanners require separate channel calibration followed by uniform correction of all swaths, before alignment.

It is possible to set some of the strips as references to keep them fixed, when they are part of an already aligned dataset, or to address special cases and circumvent some of the limitations.

Finally, quality control tools are also provided for convenience. Relative accuracy can be checked before and after registration, and after final corrections are applied. It can also be checked on arbitrary sets of parallel swaths without running any alignment. Grayscale TIFF or color JPEG z-difference maps, relative and global, can be easily exported. Shaded-relief is also available as an option. Histograms can be generated as well, and are saved as simple text files which can be plotted in gnuplot or R. Various text files can also be generated if desired, e.g. swath footprints, bounding boxes, simplified trajectories and segments and imported in GIS software.

Absolute control is also possible using ground control point (GCP) files. A user-readable file is generated summarizing the GCP statistics. There is an option to cancel the global Z bias between LiDAR and control data through a uniform vertical shift. The main reason for this capability is to address a fundamental limitation in boresight calibration (which cannot provide the z-leverarm). GCPs are never used to derive corrections otherwise, due to the fair number of parameters involved in spatially adaptive corrections.

Most parameters are optional, and have relevant default values, so don't panic when you see the full list of options! Minimal command line examples are provided with each task type, and are often sufficient. Expert options are labeled as such, and should be used with care, when well-understood, or unpredictable results may occur despite all safeguards.

Long options are defined for clarity, and short versions (single letter) sometimes made available for simplicity. Persistent options can be stored in a configuration file if used systematically.

Please refer to the end of this document for common error messages (appendix A1), and current software requirements and limitations (appendix A2).

Should anything unexpected happen, please re-run with option -oall to write all text outputs, send bug reports along with the text files, including the latest log files, to [support@bayesmap.com](mailto:support@bayesmap.com). Log files are in the output directory (or current directory if not specified); in case of crash they are still in the executable directory, with the archived ones.

---

*Please note that this is not a manual, but a documentation listing all commands, options, inputs and outputs systematically. It might also be incomplete, we are doing our best to finish it up and keep it up to date.*

*There will be a manual and tutorials, please check back on bayesmap.com!*

## CHANGES: V1.2 to V1.3

This section summarizes the changes in option names, syntax, behavior, major capabilities and output files.

Release notes (all changes including bug fixes and improvements) can be found on the BayesMap Google Group: <http://bayesmap.com/products/software-users-google-group/?place=topic%2Fbayesmapsoft%2FKXU2V77NFME%2Fdiscussion>

### Generic internal geometry model

- A new generic internal geometry model allows to handle sensor non-linearity more effectively and provides significantly more accurate corrections for certain LiDAR scanners. It has two parameters, called eps and mu, that replace the scan angle scale used in the previous versions. Eps corresponds to the z contribution of the scan angle scale and mu to the across-track scale factor. They are estimated during drift calibration and used for optimal corrections
- To maintain compatibility, absolute calibration estimates the scan angle scale and saves it as eps, however this will be upgraded in the next version to estimate both parameters
- The uniform correction module still accepts traditional angle and range scale/offset corrections through options only (no files). The new options for eps and mu are **-eps** and **-mu** for absolute values and **-eps\_diff** and **-mu\_diff** for corrections.
- The internal geometry file format has changed and has now only 2 parameters (eps and mu)

### Color JPEG z-difference map output

- 24-bit JPEG color images can be generated (instead or along grayscale TIFF images) with option **-J** or **-ojpeg**
- JPEG compression quality can be set with **-Q** or **-quality**
- 5 predefined color maps are available and can be selected with **-K** or **-color** (grayscale, gradients, solid colors)
- TIFF output is now controlled by **-M** or **-otiff** (instead of -odzmap)
- The color map application is now controlled by option **-dz\_range** (instead of -dz\_step) in all cases
- World file output is now set with **-owf** for all image formats

### Inputs: file lists

- Explicit file names can be replaced by (or combined with) file list names; multiple lists can be used
- Regular inputs: use option **-I** or **-input\_list** for each group (each option -i or -I will start a new group)
- Reference data: use option **-ref\_list** for each group
- Excluded files can also be specified with lists with option **-exclude\_list**
- Trajectory files can also be entered as file lists with options **-sbt\_list -pof\_list -sol\_list -trj\_list -potxt\_list**

### Point filtering

- Define a custom dropped class list with option **-D** or **-drop\_class** (specifies the LAS classes to be filtered out)
- Default ground filtering: using option **-f** sets dropped class list as {3,4,5,7,18}
- For LAS 1.4 files, specific channels can be selected with **-ch**
- If a reference dataset is used (alignment or relative calibration) the reference channel is set with **-ch\_ref**
- Selecting a channel with **-ch** will make sure output files have all other channel information unchanged
- By default, all non-withheld last return points are used, regardless of class or channel

### Absolute Quality Control

- A human-readable report is also generated **name\_GCP.txt** and includes file name information and summary statistics

### SSRP, SSCP, int, temporary files

- Registration parameter files, correction files, internal geometry parameter files, and various temporary files are no longer compatible with previous versions due to the addition of custom dropped class lists and channel numbers, and the change in the internal geometry parametrization
- Tiling, or point cloud sorting (in the mean flight line direction) is now enabled by default, as most software seem to produce point clouds that are poorly sorted in time order. It is the recommended setting (point statistics are also more reliable than when using original files)

# 1. INTRODUCTION - THE BASICS

BayesStripAlign is a **command-line tool** that requires a terminal (e.g. cmd or ConEmu on Windows, Terminal on Mac OS) or an external interface to be run. Currently it has no graphical user interface, so *double-clicking on the icon will not do anything*, as the software requires the user to select the input dataset and enter some optional parameters. Complex tasks can be achieved automatically with a simple command line, specifying the task, the input files, and various options. Good results can be achieved with default (missing) options in most cases so the command can be really very short. Commands can also be used in sequences in batch files, and invoked through high-level software interfaces.

## OPTIONS AND ARGUMENTS

<i>-n</i>   <i>name</i> <arg>	option: short form n, long form name; argument <arg> (if any); options are case-sensitive
<i>Conventions</i>	[] means optional,   means mutually exclusive, ... means multiple
<i>Argument syntax</i>	-n<arg>, -n <arg>, -n=<arg>, -name <arg>, -name=<arg> equivalent
<i>Spaces</i>	Separate multiple arguments, or must be in "" if part of argument (file names etc.)
<i>Missing boolean arg.</i>	Equivalent to 1 (-n same as -n1)
<i>Units</i>	Meters and decimal degrees, unless specified, see option -feet
<i>Persistent option file</i>	Text file stripalign.opt in exec. dir.; command-line options override persistent options.
<i>Persistent option syntax</i>	name=<arg> (only form accepted), one per line, "" not necessary, see example below.

## GENERAL OPTIONS

### Directory

-I   idir <dir>	Set input directory (unless explicit path is given) [default: current]
-O   odir <dir>	Set output directory [default: current]
-T   tmpdir <dir>	Set temporary directory [default: current]

### Processing

-mem <MB>	Memory made available to the program in MB [default: 1024]
-N   nthreads <n>	Max number of simultaneous threads [default: depends on processor]

### Behavior

-q   quiet <0 1>	Quiet mode, no text output, no progress bar [default: 0]
-progress <0 1>	Display progress bar [default: 1]
-c   cleanup <0 1>	Delete all temporary files from temporary directory, after processing [default: 0]
-cleanup_dir <dir>	Delete all temporary files from specified directory [default: current]

### Logs and text outputs

-oname <name>	Base name for output text files (except uniform corrections) [default: out]
-log <0 1>	Logs high-level operations, keeps history in exec. dir., latest in output dir. [default: 1]
-olog <0 1>	Leaves a copy of the latest log files in the output directory [default: 1]
-clear_log_history	Clear log history from exe directory
-ocomp_log <0 1>	When applicable, write a log file of computation results [default: 1]
-obb <0 1>	When applicable, write bounding boxes (text files) [default: 0]
-ofp <0 1>	When applicable, write trajectory and segment footprints (text files) [default: 0]
-osfp <0 1>	Write swath footprints (text files) during swath indexing [default: 0]
-oall <0 1>	Write all possible debug outputs (log files, footprints, bounding boxes etc.) [default: 0]
-owf <0 1>	When applicable, write TIFF/JPEG world files (.tfw/.jgw) along with images [default: 0]

### Units

-feet	LiDAR data is in feet (for xy and z) [default: 0, meters]
-------	---

### Information

-h   help	Display brief documentation about general options
-?   options	Display the list of available options
-v   version	Display the version and license information

## PROCESSING COMMANDS

-align	Register, compute local optimal corrections, apply corrections
-corr [-multi] [-rel]	Apply uniform or local or multichannel corrections
-calib [-rel]	Perform standard absolute calibration or relative calibration
-qc	Perform relative/absolute quality control (can be added to -align, -corr, -calib)

## EXAMPLE USAGE

```
> stripalign -corr -i *.las -o*_corr.laz -potxt file.txt -shift 0 0 0.5
```

corrects all las files from current directory, writes results in laz with \_corr suffix, using the trajectory file file.txt, the correction is a global vertical shift of 50 cm

```
> stripalign -align -i *.laz -pof file.pof -I datadir -O corrdir
```

aligns all laz files in directory datadir and puts the results in directory corrdir, same format, uses the POF trajectory file.pof - running first registration, then drift calibration and correction propagation, and finally correcting the files using all corrections (internal, calibration, biases and local).

```
> stripalign -align -i *.laz -e 1234.laz -pof file.pof -I datadir -O corrdir -oall -qc -J
```

performs the same task excluding input file 1234.laz; also writes all possible text outputs (for plotting or debugging) such as bounding boxes and footprints, and generates relative quality control outputs, such as histograms and z-difference maps (JPEG format, default colormap) before, after registration, and after correction.

```
> stripalign -qc -I \\share -i file?.laz -pof file.pof -gcp gcp.txt -t
```

performs relative quality control on a series of parallel swaths from \\share, using filter file?.laz, POF trajectory file.pof, and absolute quality control with the GCP file gcp.txt; the data are time-ordered and will not be sorted (tiled)

```
> stripalign -qc -l data_c1 -sbet_list sbet_c1 -M -J -X -dzmap_range 0.4
```

performs relative quality control on a series of parallel swaths listed in text file data\_c1, using SBET trajectory files listed in file sbet\_c1, computes pair z-difference maps (grayscale TIFF and color JPEG, default colormap) and global maximum z-difference map, with a specified range [-0.4 m,+0.4 m].

```
> stripalign -calib -i *.txt -sbet file.sbet -mnt file.mnt -oname cal
```

performs absolute calibration using all \*.txt LiDAR files, SBET trajectory file.sbet, and mount file file.mnt (boresight, leverarm used for georeferencing the data), all output file names start with cal.

```
> stripalign -corr -i file.las -pof file.pof -mnt_diff cal.mnt -int_diff cal.int
```

corrects strip file.las and puts the result in the current directory, same format, will not overwrite it (uses c\_ prefix by default), uses the POF trajectory file.pof, and files cal.mnt and cal.int with boresight/leverarm and interior parameter differences computed during a previous calibration (see example above).

```
> stripalign -clear_log_history -cleanup_dir C:\SA\tmp
```

deletes all log files from the executable directory, and all the temporary files from C:\SA\tmp

```
> stripalign -?
```

displays a compact list of options.

```
> stripalign -v
```

displays the version and license information.

```
> stripalign -h
```

displays a basic documentation (general options and processing commands above).

## PERSISTENT OPTION FILE EXAMPLE (stripalign.opt)

```
mem=4096  
tmpdir=C:\SA\tmp  
cleanup=1  
dzmap_range=0.32  
color=4  
jpeg=1
```

defines the memory size, the temporary file directory, makes sure temporary files are deleted at the end;  
defines z-difference map parameters (range +/- 0.32m, color scheme 4, color JPEG image output).

*We recommend using full name options for clarity.*

*Note: unused options may trigger warnings but no errors.*

## 2. SIMPLE UNIFORM CORRECTIONS (-corr)

### Goals:

- Apply global boresight, leverarm, scanner scale/offset, and shift corrections to one or multiple swaths.
- To apply different sets of parameters, use separate commands.

### Usage:

```
stripalign -corr
({-i <files...>}... | {-l <lists...>}...)
[{-e <files...> | -exclude_list <lists...>}] [-iparse <format>] [-ch <n>]
(
  ( {-pof <files...>}... | {-pof_list <lists...>}... )
  ( {-sol <files...>}... | {-sol_list <lists...>}... )
  ( {-trj <files...>}... | {-trj_list <lists...>}... )
  ( {-sbet <files...>}... | {-sbet_list <lists...>}... )
  ( {-potxt <files...>}... | {-potxt_list <lists...>}... )
  [-poseg | [-pparse <format>] )
  [-gps_date <YYYY-MM-DD>...] [-G<0|1>] [-W<0|1>]
  [-R <a b c>] [-o<[pre]*[suf].[ext]>] [-oparse <format>]
  [-mnt_diff <file>] [-bore_diff <w p k>] [-lever_diff <x y z>]
  [-int_diff <file>] ([-eps_diff <s o>] [-mu_diff <s o>] |
  [-theta_so_diff <s o>] [-range_so_diff <s o>]) [-shift <x y z>]
```

### Options:

-i   input <files...>	Input LiDAR files (.las, .laz, .txt)
-l   input_list <lists...>	Text files specifying file lists (one file name per row), same purpose <b>repeat options for each separate group -i &lt;files1...&gt; -i &lt;files2...&gt; -l &lt;lists3...&gt;...</b> (one GPS week per group) (pitch constant within each group for calibration)
-e   exclude <files...>	Exclude one or more LiDAR files from all input (options -i or -r)
-exclude_list <lists...>	Use text files containing file lists (one per row), same purpose
-iparse <format>	Parse string for input text LiDAR files [default: xyzt]
-ch <n>	Read/write only specified channel [0,3] (LAS 1.4 only) [default: 4 (all)]
-pof <files...>	Riegli POF trajectory files, in principle in geodetic coordinates
-pof_list <lists...>	Text files specifying file lists (one file name per row), same purpose
-sol <files...>	Leica SOL trajectory files, in principle in geodetic coordinates
-sol_list <lists...>	Text files specifying file lists (one file name per row), same purpose
-trj <files...>	Terrascan TRJ projected trajectory files (option -G will be ignored)
-trj_list <lists...>	Text files specifying file lists (one file name per row), same purpose
-sbet <files...>	Applanix SBET trajectory files, in principle in geodetic coordinates
-sbet_list <lists...>	Text files specifying file lists (one file name per row), same purpose
-potxt <files...>	Text trajectory files, can be projected or in geo. coordinates
-potxt_list <lists...>	Text files specifying file lists (one file name per row), same purpose <b>repeat these options for each separate group (one GPS week per group)</b>
-poseg	Use segment trajectory file .seg for each input (.seg appended to each LiDAR file) Format: {x y z (first) x y z (last)}, can be geodetic or projected (use option -G)
-pparse <format>	Parse string for input text trajectory files [default: txyzwpk]
-gps_date <YYYY-MM-DD>	GPS date (ISO 8601) to convert GPS week seconds to adjusted GPS time [default: auto] <b>may repeat option for each group (one GPS date per group, or one for all groups)</b>
-G   po_geo <0 1>	Position is in geo. coordinates (vs. projected as LiDAR data) [default: 1]
-W   po_weeksec <0 1>	Timestamps are GPS week seconds (vs adjusted GPS) if format allows both [default: 1]
-R   imu_body_rot <a b c>	IMU to BODY rotation, angles a b c about axes x,y,z (degrees) [default 0 0 0]
-o<[pre]*[suf].[ext]>	Output file name format: define prefix, suffix and extension [default: same name/format]
-oparse <format>	Parse string for input text LiDAR files [default: xyzt]
-mnt_diff <file>	Mount difference text file: x y z (leverarm) w p k (boresight)
-bore_diff <w p k>	Boresight angle difference, overrides file values [default: 0 0 0]
-lever_diff <x y z>	Leverarm difference, overrides file values [default: 0 0 0]
-int_diff <file>	Internal scale ratio & offset difference text file: scale offset (angle) scale offset (range)
-eps_diff <s o>	Eps parameter difference (internal generic model), overrides file value [default: 0]
-mu_diff <s o>	Mu parameter difference (internal generic model), overrides file value [default: 0]
-theta_so_diff <s o>	Scan angle scale ratio and offset difference, force scale/offset correction [default: 1 0]
-range_so_diff <s o>	Range scale ratio and offset difference, force scale/offset correction [default: 1 0]

-shift <x y z>                    Global shift vector [default: 0 0 0]

**Examples:**

```
> stripalign -corr -i file.laz -pof file.pof -mnt_diff cal.mnt -int_diff cal.int
```

corrects file file.laz using POF trajectory file.pof and a file cal.mnt with boresight/leverarm differences and a file cal.int with internal parameter eps and mu differences (from a previous calibration).

```
> stripalign -corr -i file.laz -pof file.pof -mnt_diff cal.mnt -int_diff cal.int -ch 2 -oname file_2
```

same as previous command but only corrects channel 2 (must be LAS 1.4) preserving data from other channels; other subsequent commands are needed to correct other channels, sequentially.

```
> stripalign -corr -i file.laz -sol file.sol -bore 0 0 14 -mnt_diff cal.mnt -int_diff cal.int
```

same purpose as first example, but using an explicit boresight (here corresponding to channel 1 of the Riegl Q1560 scanner), mandatory for non across-track looking instruments.

```
> stripalign -corr -i *.las -sbet file.out -R 0 0 180 -ocorr_*.laz -mnt_diff cal.mnt  
-theta_so_diff 1.001 0.15
```

corrects all \*.las files in the current directory using the SBET trajectory file file.out and a 180 deg IMU/BODY rotation around the z axis, outputs laz files with corr\_ prefix, and uses a file cal.mnt with boresight/leverarm differences and a scan angle scale factor of 0.001 and a 15 cm offset, thus selecting the scale/offset internal geometry model.

```
> stripalign -corr -i *.las -sbet file.out -R 0 0 180 -ocorr_*.laz -mnt_diff cal.mnt  
-eps_diff 0.001 -mu_diff 0.02
```

same as previous command with the generic internal geometry model and related parameters eps and mu.

```
> stripalign -corr -i *.txt -iparse txyz -potxt file.txt -pparse t##yx#zwpk -G0 -olaz -shift 0.5 0.7 0.9
```

corrects all \*.txt files in the current directory formatted as t x y z (first 4 columns), using the text trajectory file file.txt where t is the 1st column, y the 4th, x the 5th, z and z roll pitch yaw the 7th to 10th, the trajectory is already projected, the output is laz, and the correction consists of a global shift vector (0.5,0.7,0.9) m.

```
> stripalign -corr -i *.laz -trj *.TRJ -W0 -bore 0 10 0 -bore_diff 0.1 0.2 0.05 -lever_diff 0.15 0.25 -0.1 -  
theta_so_diff 0.9995 0 -O outdir
```

corrects all laz files in the current directory using TRJ trajectory files (one per swath) and explicit corrections, and a non-nadir boresight (pitch 10 deg), puts results in directory outdir; TRJ files have GPS week seconds, and will be converted to adjusted GPS time or not depending on laz file timestamps, automatically

```
> stripalign -corr -i *.laz -trj *.TRJ -W0 -gps_date 2015-10-03 -bore 0 10 0 -bore_diff 0.1 0.2 0.05 -  
lever_diff 0.15 0.25 -0.1 -theta_so_diff 0.9995 0 -O outdir
```

same as above, specifying the acquisition date, so any week seconds timestamps are converted to adjusted GPS time (the gps week is computed from the given date)

```
> stripalign -corr -i group1/*.laz -i group2/*.laz -trj group1/*.TRJ -trj group2/*.TRJ -W0 -bore 0 10 0 -  
bore_diff 0.1 0.2 0.05 -lever_diff 0.15 0.25 -0.1 -theta_so_diff 0.9995 0 -O outdir
```

same as above, with two groups of inputs and related PO files, with automatic conversion to adjusted GPS time for each group; the GPS week and GPS time type can be different between groups

*Note 1: the use of segments is not supported for uniform corrections in the current version.*

*Note 2: time ordering in point cloud files does not matter; points are read and corrected sequentially and ordering is preserved.*

### 3. SINGLE SWATH ABSOLUTE QUALITY CONTROL (-qc)

#### Goal:

- Check the absolute quality of a swath using a ground control point (GCP) text file.

#### Usage:

```
stripalign -qc -i <file> [-iparse <format>] ([-f<0|1>] | [-D<i...>]) [-t<0|1>]
{PO options} -gcp <file> [-gparse <format>]
(expert)
[-gsd <v>] [-mul_gsd <v>] [-tile_size <v>]
```

#### Options:

-i   input <file>	Input LiDAR file (.las, .laz, .txt)
-iparse <format>	Parse string for input text LiDAR file [default: xyzt]
-f   filter_ground <0 1>	Filter out points with LAS classes {3,4,5,7,18} [default: 0]
-D   drop_class <i...>	Filter out points with specified LAS classes [default: empty]
-t   time_order <0 1>	Points are not time-ordered, indexing used instead of sorting & tiling [default: 0]
{PO options}	<a href="#">See section 2 for position/orientation options</a>
-gcp <file>	Input text GCP file <b>Format:</b> space, tab or comma separated numbers, only numbers before/between data Index and x y z coordinates required, same projection and unit as LiDAR input Column parse string for input GCP file, use # to ignore column [default: ixyz]
-gparse <format>	
(expert)	
-gsd <v>	Ground sampling distance (GSD) for gridding algorithm [default: auto]
-mul_gsd <v>	Automatic GSD multiplier [default: 1.0]
-tile_size <v>	Tile size for I/O optimization [default: auto]

#### Output files:

<name>_GCP.aqc	Absolute quality report, text file Columns: {index 0 x y z_gcp z_data x_CS y_CS}* *CS = swath coordinate frame, rotated such that x = mean flight line direction
<name>_GCP.txt	Absolute quality report, user-readable text file: - Input GCP file name and output QC file name (see above) - Summary statistics: GCP found, Mean_DZ, Sigma_DZ, CI_95** - Swath file list: {swath# file name} - GCP list: {GCP# swath# X Y Z_GCP Z_LiDAR Z_LiDAR-Z_GCP IN_CI95**} **CI95 = 95 % confidence interval, assuming a Gaussian distribution

\*CS = swath coordinate frame, rotated such that x = mean flight line direction

\*\*CI95 = 95 % confidence interval, assuming a Gaussian distribution

#### Examples:

```
> stripalign -qc -i file.laz -sol file.sol -gcp gcpfile -t0 -c -oname QC1
```

checks the absolute quality of strip file.laz with SOL trajectory file.sol against the GCP from file gcpfile (formatted as index x y z), the points are not time ordered so caches will be created, and deleted after use, creates an absolute quality report text file QC1\_GCP.aqc and a user-readable QC1\_GCP.txt file.

```
> stripalign -qc -feet -i file.las -sbet file.out -gcp gcpfeet -gparse xyzi -f
```

checks the absolute quality of strip file.las (horizontal and vertical unit for data and GCP is feet), with SBET trajectory file.out, against the GCP from file gcpfeet (formatted as x y z index, also in feet) and filters out non-ground points (based on LAS information), creates an absolute quality report file out\_GCP.aqc and the more readable out\_GCP.txt file.

```
> stripalign -qc -feet -i file.las -sbet file.out -gcp gcpfeet -gparse #ixyz -f
```

same, with different GCP file format: first column ignored (number only!), then index, y, x, z (rest of line ignored).

```
> stripalign -qc -i file.laz -sol file.sol -gcp gcpfile -t0 -c -oname QC1 -D 4 5 7
```

same as first example, with explicit filtering of LAS classes 4 5 and 7.

## 4. MULTIPLE SWATH RELATIVE/ABSOLUTE QUALITY CONTROL (-qc)

### Goals:

- Check the relative quality of a series of swaths: compute z-difference statistics, histograms and optional z-difference or hill-shaded maps in grayscale TIFF and/or color JPEG.
- Check the absolute quality of one or multiple swaths if a ground control point (GCP) text file is provided.

### Notes:

1. Relative QC (and related options) also available during alignment, local corrections and calibration.
2. Absolute QC (and related options) also available during local corrections.
3. Only approx. parallel and straight swaths will be used. Others will be ignored and logged to swaths\_rej.log

### Usage:

```
stripalign -qc {input options} [-ref_dir <file>] ([-f<0|1>] | [-D<i...>]) [-t<0|1>]
{PO options} {GCP options}
[-H<0|1>] [-histo_step <v>] [-histo_range <v>]
[-M<0|1>] [-J<0|1>] [-K<n>] [-Q<v>] [-X<0|1>] [-dzmap_range <v>]
(expert)
[-shade<0|1>] [-shade_ai <az inc>] [-checker <size>]
[-tile_size <v>] [-angle_tol <a>]
[-gsd <v>] [-mul_gsd <v>] [-rough_max <v>] [-density_min <v>]
[-stddev_niter <n>] [-seg_traj_tol <v>]
```

### Options:

{input options}	See section 2 for LiDAR file inputs and related options
-ref_dir <file>	Reference direction given by file (force common swath direction using specific file)
-f   filter_ground <0 1>	Filter out points with LAS classes {3,4,5,7,18} [default: 0]
-D   drop_class <i...>	Filter out points with specified LAS classes [default: empty]
-t   time_order <0 1>	Points are time-ordered, indexing used instead of sorting & caching [default: 0]
{PO options}	See section 2 for position/orientation options
{GCP options}	See section 3 for GCP options
-H   ohisto <0 1>	Computes pair, global and sum z-diff histograms as text files [default: 1]
-histo_step <v>	Sets the z-diff step size for histograms [default: 0.002]
-histo_range <v>	Sets the z-diff range for histograms [default: 2.0]
-M   otiff <0 1>	Computes pair and global z-diff maps in TIFF (+.tifw with option -owf) [default: 0]
-J   ojpeg <0 1>	Computes pair and global z-diff maps in JPEG (+.jgw with option -owf) [default: 0]
-K   color <n>	For JPEG, set color scheme [0-4] (predefined color map)*** [default: 0]
-Q   quality <n>	For JPEG, set compression quality [50-100] [default: 80]
-X   maxdzmap <0 1>	Uses maximum absolute z-diff instead of signed z-diff for global map [default: 0]
-dzmap_range <v>	Sets the z-difference range [-v,v] for images [default: 0.5]
(expert)	
-E   shade <0 1>	Computes shaded-relief maps instead of z-differences [default: 0]
-shade_ai <az inc>	Sets the azimuth and inclination for shading [default: 45 15]
-checker <size>	For shading, uses a checkerboard pattern of size GSD units to alternate between swaths
-tile_size <v>	Tile size for I/O optimization [default: auto]
-angle_tol <a>	Angle tolerance: max. deviation from dominant angle (parallel swath check) [default: 5.0]
-gsd <v>	Ground sampling distance (GSD) for gridding algorithm [default: auto]
-mul_gsd <v>	Automatic GSD multiplier [default: 1.0]
-rough_max <v>	Maximum allowed roughness (local z standard deviation) for data filtering [default: 1.0]
-density_min <v>	Maximum allowed point density for data filtering, in pts/GSD^2 [default: 1.0]
-stddev_niter <n>	Number of iterations for robust standard deviation estimation [default: 30]
-seg_traj_tol <v>	Segment tolerance or maximum error btw. original trajectory and segment [default: 25.0]

### Output files:

{AQC output}	See section 3 for absolute QC output, if GCP are used
<name>.sp	Swath parameter file: #S swaths - columns: {swath# direction(cs) group file_name}
	#P pairs - columns: {pair# swath1# swath2# x0(cs) y0(cs) dx(cs) dy(cs)}*
<name>.rqc	Relative quality report, text file

<name>_DZ_i_j.ext	Columns: {pair# z_bias z_stddev outlier_rate}
<name>_DZ.ext	Images**: z-difference map for swaths i,j (z_j-z_i)
<name>_cmap.jpg	Image**: global z-difference map (maximum absolute value if -X, middle cut otherwise) (generated if 2 pairs or more)
<name>_HDZ_i_j	Color box (z-difference range used to generate color images), bottom is min, top is max (generated if JPEG output requested) - size 16x256 pixels
<name>_HDZ	z-difference histogram for swaths i,j (z_j-z_i)
<name>_HDZ_sum	Columns: {dz n pdf} where n is the count in bin centered on dz, pdf = probability density
swaths_rej.log	z-difference histograms for all swaths in a single file
	Columns: {dz n pdf pair#}
	z-difference histogram sum for all swaths (generated if 2 pairs or more)
	Columns: {dz n pdf}
	List of rejected swaths (due to trajectory requirements)

\* *CS = common swath coordinate frame, rotated such that x = mean flight line direction coordinates are in the same projection as the data, and same vertical and horizontal units (e.g. feet)*

\*\* *Image format: 24-bit color JPEG (ext=jpg) or 8-bit grayscale TIFF (ext=tif), both can be used simultaneously Color images are generated from 256 gray level z-difference or shading maps by applying a color map*

\*\*\* *Predefined color maps: (option -K)*  
*0 black-white (gradient)*  
*1 blue-black-red (gradient)*  
*2 green-blue-back-red-white (gradient)*  
*3 green-cyan-blue-back-red-yellow-white (gradient)*  
*4 green, yellow, orange, red (constant over interval)*

## Examples:

```
> stripalign -qc -i file?.laz -pof file.pof -gcp gcpfile -c
```

takes all files named file?.laz (current directory) with POF trajectory file.pof, uses control points from file gcpfile with default formatting (columns: index x y z), tiling will be used and temporary files will be deleted; generates histograms, relative and absolute quality report files with base name=out by default, and lists used strips in .sp text file, and unused strips in swaths\_rej.log.

```
> stripalign -qc -i *.laz -sbt file.out -gcp gcpfeet -feet -M -shade
```

takes all laz files from current directory, uses SBET trajectory file.out and control points from file gcpfeet, data (vertical and horizontal units) as well as GCP are in feet, generates TIFF maps with hill-shading instead of z-differences for each pair of swaths, combining swath elevations so that misalignments are visible thanks to the shading; also writes histograms and text file reports and logs similar to the previous command.

```
> stripalign -qc -i *.txt -ref_dir file1.txt -potxt PO.txt -feet -gsd 5
```

takes all \*.txt files (current directory) and uses the group of swaths containing swath file1.txt (useful when it may be rejected by default, due to large deviation from the dominant swath direction), uses trajectory information in text from PO.txt (default column format: time x y z roll pitch yaw), uses feet as data unit, forces the gsd to 5 m; same type of output as previous example without absolute quality reporting.

```
> stripalign -qc -i abc*.las -sol file.sol -H0 -J -dzmap_range 1 -oname QC2
```

takes all files named abc\*.las (current directory) with trajectory in SOL format file.sol, names file outputs QC2, does not generate histograms, writes JPEG z-difference maps with color map applied to range [-1m,1m].

```
> stripalign -qc -i abc*.las -sol file.sol -H0 -J -X -K3 -dzmap_range 1 -oname QC2
```

same but uses the max to combine pairwise z-difference maps into a global absolute z-difference map, and uses color scheme 3 (only the upper half for the global map).

```
> stripalign -qc -i abc*.las -sol file.sol -H0 -J -K4 -dzmap_range 0.32 -oname QC2
```

same but uses constant colors over 8 cm ranges to produce accuracy level maps.

```
> stripalign -qc -i g1_file?.laz -pof g1.pof -i g2_file?.laz -pof g2.pof -gcp gcpfile
```

uses two groups (or separate collects): first group with all files named g1\_file?.laz (current directory) with POF trajectory g1.pof, second group with files g2\_file?.laz and trajectory g2.pof, uses control points from file gcpfile with default formatting (columns: index x y z); this allows a correct determination of adjusted GPS time for each group when PO or LiDAR data are in week seconds.

```
> stripalign -qc -i g1_file?.laz -pof g1.pof -gps_date 2015-09-23 -i g2_file?.laz -pof g2.pof -gps_date 2016-01-15 -gcp gcpfile
```

same but with all data in week seconds, GPS dates for each collect are known and provided to allow an internal conversion to adjusted GPS time (no date input = automatic determination).

```
> stripalign -qc -i file?.las -pof file.pof -gcp gcpfile -c -ch 1 -f
```

same as first example with point filtering: only channel 1 is used (assuming extended channel information available is LAS 1.4 format, channels not separated in different files) and only points belonging to solid surfaces are kept.

```
> stripalign -qc -i file?.las -pof file.pof -gcp gcpfile -c -ch 2 -D 3 4
```

same as first example with point filtering: only channel 2 is used (assuming extended channel information available is LAS 1.4 format) and points having classes 3 and 4 are filtered out.

```
> stripalign -qc -i g1_data -sbet g1_sbet -gcp g1_gcp -oname g1
```

performs QC with GCP file g1\_gp, using input files listed in g1\_data and sbet files listed in g1\_sbet, with default formatting (columns: index x y z), tiling will be used; generates histograms, relative and absolute quality report files with base name g1, and lists used strips in .sp text file, and unused strips in swaths\_rej.log.

```
> stripalign -qc -i g1_data -i g2_data -i g3_data -sbet g1_sbet -sbet g2_sbet -sbet g3_sbet -gcp gcpfile
```

performs QC with GCP file gcpfile, using input files listed in g1\_data for group 1, g2\_data for group 2 and so on, and sbet files listed in g1\_sbet for group1 and so on, with default formatting (columns: index x y z), tiling will be used; generates histograms, relative and absolute quality report files with default base name out, and lists used strips in .sp text file, and unused strips in swaths\_rej.log; trajectory and LiDAR data files are matched per group, and GPS week determination is also done independently for each group (if not all files have adjusted GPS time).

## 5. ABSOLUTE CALIBRATION (-calib)

### Goals:

- Perform boresight calibration (roll, pitch, yaw, y-leverarm, scan angle scale), from:
  - a cross-flight with at least two lines flown in opposite directions + two cross lines
  - approx. parallel swaths, at least 3 swaths required, at least two flown in opposite directions
  - a single pair of swaths, which only constrains a subset of the parameters
- Perform relative quality control (see section 4) before/after registration to check the validity of the motion model used during calibration.

### Notes:

1. Calibration requires very short flight lines to minimize IMU drifts.
2. Flight lines must be approximately straight but can intersect at any angle, rejected swath files are logged.
3. Boresight angles must be provided if non-across-track scanners used (e.g. Riegl Q1560)

### Usage:

```
stripalign -calib {input options} {PO options} [-qc {QC options}]
([-f<0|1>] | [-D<i...>]) [-t<0|1>] [-x <v>]
[-mnt <file>] [-bore <w p k>] [-lever <x y z>]
[-int <file>] [-eps <v>] [-mu <v>]
(expert)
{EXP options} [-frac_min <v>]
```

### Options:

{input options}	See section 2 for LiDAR file inputs and related options
{PO options}	See section 2 for position/orientation options
-qc {QC options}	Enable QC, see section 4 for QC options
-f   filter_ground <0 1>	Filter out points with LAS classes {3,4,5,7,18} [default: 0]
-D   drop_class <i...>	Filter out points with specified LAS classes [default: empty]
-t   time_order <0 1>	Points are time-ordered, indexing used instead of sorting & caching [default: 0]
-x   true_lever_x	True x-leverarm value [default: x-leverarm given with -mnt or -lever]
-mnt <file>	Mount text file: {x y z roll pitch yaw} (leverarm in m, boresight in deg)
-bore <w p k>	Boresight angles, overrides file values [default: 0 0 0] Use 0 0 14 and 0 0 -14 for Riegl Q1560 channels 1 and 2, respectively
-lever <x y z>	Leverarm, overrides file values [default: 0 0 0]
-int <file>	Internal parameter (generic model) text file: {eps mu}
-eps <s o>	Eps parameter (internal generic model), overrides file value [default: 0]
-mu <s o>	Mu parameter (internal generic model), overrides file value [default: 0]
(expert)	
{EXP options}	Expert options, see section 4
-frac_min <v>	Minimum fraction of area with sufficient point density for registration [default: 0.1]

### Output files:

<name>.mnt	Mount file, text format {x y z roll pitch yaw} (leverarm in m, boresight in deg)
	Sum of provided parameters and estimated differences.
<name>.int	Internal parameter file, text format {theta_scale theta_offset range_scale range_offset}
	Only the scan angle scale is estimated and written as eps (mu is 0)
	Product of provided parameter and estimated ratio.
<name>_diff.mnt	Results: mount parameter difference (between real and used for georef.)
<name>_diff.int	Results: internal parameter difference (between real and used for georef.)
<name>.log	Log file with swath information and computed parameters
swaths_rej.log	List of rejected swaths (due to trajectory requirements)
{QC output}	qc=1: QC output files except global z-difference map and swath list, see section 4

## Examples:

```
> stripalign -calib -i *.laz -sbet sbet.out
```

shortest command line, runs the calibration on \*.laz files in the current directory, using the SBET trajectory file.out, assuming all mount and internal parameters have default values (0 for boresight/leverarm/offsets, 1 for scale factors), writes estimated corrections (differences and ratio) are as out\_diff.mnt and out\_diff.int; writes log file out.log and list of rejected data files swaths\_rej.log.

```
> stripalign -calib -i *.las -sbet sbet.out -gc -J -Q90 -mnt file.mnt -int file.int
```

runs the calibration on all las files from the current directory, using the SBET trajectory file.out, generates relative quality control histograms and z-difference maps (JPEG, quality 90, default color map) for all overlapping pairs used; assumes data were georeferenced with mount parameters from file.mnt and internal parameters from file.int; behaves as previous command and writes updated out.mnt and out.int files based on input files.

```
> stripalign -calib -i cal*.laz -sol file.sol -mul_gsd 2
```

uses all cal\*.laz files for calibration (current directory), with trajectory in SOL format file.sol, accelerates the computation by using a gsd twice the automatic gsd, see first example for output files.

```
> stripalign -calib -i Channel_1*.laz -sbet sbet.out -oname ch1 -bore 0 0 14  
> stripalign -calib -i Channel_2*.laz -sbet sbet.out -oname ch2 -bore 0 0 -14
```

performs calibration as in the first example, separately for channels 1 and 2 of the Riegl Q1560 instrument, and sets the base name for output files as ch1 and ch2, respectively.

```
> stripalign -calib -i *.las -sbet sbet.out -ch 0 -oname ch0  
> stripalign -calib -i *.las -sbet sbet.out -ch 1 -oname ch1  
> stripalign -calib -i *.las -sbet sbet.out -ch 2 -oname ch2
```

performs calibration as in the first example, separately for channels 0, 1 and 2, stored together in the LAS 1.4 files (channels not separated in different files), with the extended channel flag used to identify them, and sets the base name for output files as ch0, ch1, ch2 respectively.

```
> stripalign -calib -i *.laz -sbet sbet.out -D 3 4 18
```

same as first example but leaving out all points having LAS classes 3, 4 and 18.

## 6. ALIGNMENT: REGISTRATION AND LOCAL CORRECTIONS (-align)

### Goals:

- Compute relative registration parameters from multiple overlapping swaths
- Perform attitude drift and apparent leverarm calibration (systematic errors)
- Compute optimal local corrections (random errors)
- Apply local corrections to LiDAR swaths
- Perform multiple swath relative quality control before/after registration
- Perform relative and absolute swath quality control after correction

### Notes:

1. Only approx. parallel and straight swaths will be used. Others will be ignored and logged to swaths\_rej.log
2. For forward/backward looking instruments, calibration corrections should be applied beforehand if calibration errors are significant (e.g. 0.1 deg)
3. Multi-part flight lines cannot be used directly, they need to be merged before alignment
4. Computed registration parameters can be reused (relative corrections, fine tuning)

### Usage:

```
stripalign -align {input options} [{-r <files...>}...] [{-ref_list <lists...>}...] [-ch_ref <n>]
([-f<0|1>] | [-D<i...>]) [-t<0|1>] [-R <a b c>]
{PO options} [-qc {QC options}] {GCP options} [-Z<0|1>]
(-C1 {CAL options} [-x <v>] | -C0 {CORR options})
[-o<[pre]*[suf].[ext]>] [-oparse <format>] [-nil] [-corr <0|1>]

(expert)
{EXP options} [-frac_min <v>] [-angle_tol <a>] [-traj_tol <v>]
[-sort_asc <0|1>] [-s <v>]
[-gps_err <x y z>] [-imu_err <w p k>] [-adj_unc <0|1>]
[-S<0|1>] [-A<0|1>] [-B<0|1>] [-L<0|1>] [-loc_rej_out <0|1>]
```

### Options:

<b>{input options}</b>	See section 2 for LiDAR file inputs and related options
-r   ref <files...>	Input LiDAR files as fixed references (will not be corrected, but used for registration)
-ref_list <lists...>	Use text files containing file lists (one per row), same purpose <b>repeat options for each separate group -r &lt;files1...&gt; -r &lt;files2...&gt; -ref_list &lt;lists3...&gt;...</b>
-ch_ref <n>	Read only specified channel for reference data [0,3] (LAS 1.4 only) [default: 4 (all)]
-f   filter_ground <0 1>	Filter out points with LAS classes {3,4,5,7,18} [default: 0]
-D   drop_class <i...>	Filter out points with specified LAS classes [default: empty]
-t   time_order <0 1>	Points are time-ordered, indexing used instead of sorting & caching [default: 0]
-R   imu_body_rot <a b c>	IMU to BODY rotation, angles a b c about axes x,y,z (degrees) [default 0 0 0]
<b>{PO options}</b>	See section 2 for position/orientation options
-qc	Enable QC [default: 0]
<b>{QC options}</b>	qc=1: See section 4 for QC options
<b>{GCP options}</b>	qc=1: See section 3 for GCP options
-Z   corr_gcp_z <0 1>	Correct z bias btw. GCP and LiDAR by applying uniform z shift [default: 0]
-C   cal <0 1>	Perform attitude drift, leverarm and scan angle scale calibration [default: 1]
<b>{CAL options}</b>	cal=1: See section 5 for absolute calibration parameter options
-x   true_lever_x	cal=1: Force true x-leverarm value, disable x-leverarm estimation [default: estimated]
<b>{CORR options}</b>	cal=0: See section 2 for calibration parameter correction options
-o<[pre]*[suf].[ext]>	Output file name format: define prefix, suffix and extension [default: same name/format]
-oparse <format>	Parse string for input text LiDAR files [default: xyzt]
-nil	Do not write corrected files (run registration and optional correction comp.) [default: 0]
-corr <0 1>	Compute optimal corrections after registration [default: 1]
<b>(expert)</b>	
<b>{EXP options}</b>	Expert options, see section 4
-frac_min <v>	Minimum fraction of area with sufficient point density for registration [default: 0.1]
-angle_tol <a>	Angle tolerance: max. deviation from dominant angle (parallel swath check) [default: 5.0]
-traj_tol <v>	Trajectory tolerance or maximum error btw. original and simplified [default: 10.0]

-sort_asc <0 1>	Sort swaths in ascending Y order (in the common swath frame) [default: 1]
-s   reg_size <v>	Region size for spatially adaptive parameter estimation and correction [default: auto]
-gps_err <x y z>	cal=1: GPS error (RMS) x y z [default: 0.01 0.01 0.02]
-imu_err <w p k>	cal=1: IMU error (RMS) roll pitch yaw [default: 0.001 0.001 0.001]
-S   corr_int <0 1>	Enable internal geometry corrections (angle, range scale and offset) [default: 1]
-A   corr_att <0 1>	Enable attitude drift and mount corrections [default: 1]
-B   corr_bias <0 1>	Enable bias corrections (remaining systematic errors, slow time-varying) [default: 1]
-L   corr_loc <0 1>	Enable local corrections (spatially adaptive or fast time-varying) [default: 1]
-u   adj_unc <0 1>	Adjust estimated uncertainties to be compatible with gps_err [default: 1]
-loc_rej_out <v>	Outlier rejection threshold in std dev (registration artifact filtering) [default: 6.0, disabled]

### Output files:

<name>.ssrp	Swath segment registration parameter (SSRP) file #H header - version number, information about number of swaths, pairs, area geometry #S swaths - columns: {swath# direction <sub>(CS)</sub> group# file_name}* #T trajectory - columns: {swath# x <sub>(CS)</sub> y <sub>(CS)</sub> z <sub>0</sub> y1 <sub>(CS)</sub> z1...} #P pairs - columns: {pair# swath1# swath2# x <sub>(CS)</sub> y <sub>(CS)</sub> dx <sub>(CS)</sub> dy <sub>(CS)</sub> }* Registration parameters - columns: {index pair# Tx Ty Tz Ax Az s_Tx s_Ty s_Tz s_Ax s_Az z quality sigma_z out_frac npts}
<name>.sscp	swath segment correction parameter (SSCP) file #H header - version number, .ssrp file name (used for correction computation) #I internal parameters - columns: {theta_scale theta_offset range_scale range_offset} Correction parameters - columns: {index pair# Tx Ty Tz Ax Ay Az}
swaths_rej.log {QC output}	List of rejected swaths (due to trajectory requirements) qc=1: QC output files except global z-difference map and swath list, see section 4 _corr suffix appended to base name for post-correction QC files
{AQC output} {CAL output}	qc=1: See section 3 for absolute QC output, if GCP are used cal=1: See section 5 for calibration output

### Examples:

```
> stripalign -align -i *.laz -sbt sbet.out
```

shortest command line, runs the full alignment (corrections included) on \*.laz files in the current directory, appends c\_ to file names to avoid overwriting, uses the SBET trajectory sbet.out, uses default parameter values.

```
> stripalign -align -corr 0 -i *.laz -sol trj.sol -oname set1
```

runs only the registration and writes set1.ssrp to current directory, uses all laz files as input and SOL format trajectory file trj.sol.

```
> stripalign -align -corr 0 -i *.laz -sol trj.sol -oname set1 -qc -M -J -O results -nil
```

same as previous command, with relative QC enabled, and z-difference maps output in TIFF and JPEG, using base name set1, and all outputs are written to directory results; data files are not corrected, only QC and text outputs are written.

```
> stripalign -align -i *.las -potxt file.pos -x 0 -R 0 0 180
```

runs the full alignment (corrections included) on \*.las files in the current directory, appends c\_ to file names to avoid overwriting, uses the text trajectory file.pos, forces the x-leverarm to 0, points are not time-ordered and will be tiled, and there is a body to IMU rotation angle of 180 deg around the z axis.

```
> stripalign -align -i *.las -potxt file.pos -x 0 -R 0 0 180 -L0 -traj_tol 30 -frac_min 0.05
```

same as previous command, but local corrections are disabled (drift calibration and biases still enabled), trajectory tolerance level set to 30 m (more curved, noisier trajectories), and the minimum fraction is lowered to 5%.

```
> stripalign -align -i *.las -ch 2 -r *.las -ch_ref 1 -potxt file.pos -O results
```

aligns all channel 2 data with respect to channel 1 data, using extended channel information from LAS 1.4 files (channels not separated in different files), puts the corrected files and outputs into directory results, by computing optimal corrections and allowing only the swaths from the first group (-i) to be corrected, the others (-r) are fixed, unchanged.

```
> stripalign -align -i *_2.las -r *_1.las -potxt file.pos -O results
```

aligns all \*\_2.las strips with respect to \*\_1.las strips, puts the corrected files and outputs into directory results, by computing optimal corrections and allowing only the swaths from the first group (-i) to be corrected, the others (-r) are fixed, unchanged.

```
> stripalign -align -i *.laz -sbet sbet.out -qc -gcp GCP.txt -Z -O results
```

runs the full alignment (corrections included) on \*.laz files in the current directory, uses the SBET trajectory sbet.out, runs relative QC (histograms only), performs absolute QC with control points GCP.txt and shifts the data vertically to match the control data, puts all output in directory results.

```
> stripalign -align -l datafiles -sbet_list sbetfiles -qc -gcp GCP.txt -Z -O results
```

same purpose as previous command with input file names stored in text files (datafiles for LiDAR inputs, sbetfiles for SBET trajectory files); if more than one SBET file matching will be performed based on time stamps.

```
> stripalign -align -l datafiles1 -ref_list datafiles0 -sbet_list sbetfiles1 -sbet_list sbetfiles0 -qc -gcp GCP.txt -Z -O results
```

same purpose as previous command but with a reference data set datafiles0 (files that will not be corrected and assumed fixed), and a corresponding set of SBET trajectories sbetfiles0.

## 7. LOCAL CORRECTIONS FROM PARAMETER FILE (-corr)

### Goals:

- Perform attitude drift and apparent leverarm calibration (systematic errors)
- Compute optimal local corrections (random errors)
- Apply local corrections to LiDAR swaths
- Perform relative and absolute swath quality control after correction

### Notes:

1. Registration parameters have already been computed using -align, and saved as .ssrp file

### Usage:

```
stripalign -corr -p <file> [-qc {QC options} {GCP options}]  
(-C1 {CAL options} [-x <v>] | -C0 {CORR options})  
[-o<[pre]*[suf].[ext]>] [-oparse <format>] [-nil]  
(expert) [LOC options]
```

### Options:

-p   param	Parameter file generated with -align (extension .ssrp)
-qc	Enable QC [default: 0]
{QC options}	qc=1: See section 4 for QC options
{GCP options}	qc=1: See section 3 for GCP options
-C   cal <0 1>	Perform attitude drift, leverarm and scan angle scale calibration [default: 1]
{CAL options}	cal=1: See section 5 for absolute calibration parameter options
-x   true_lever_x	cal=1: Force true x-leverarm value, disable x-leverarm estimation [default: estimated]
{CORR options}	cal=0: See section 2 for calibration parameter correction options
-o<[pre]*[suf].[ext]>	Output file name format: define prefix, suffix and extension [default: same name/format]
-oparse <format>	Parse string for input text LiDAR files [default: xyzt]
-nil	Do not write corrected files (run registration and optional correction comp.) [default: 0]
(expert) {LOC options}	Local correction expert options, see section 6

### Output files:

{LOC output}	Local correction file <name>.sscp, see section 6
{QC output}	qc=1: QC output files except global z-difference map and swath list, see section 4 _corr suffix appended to base name
{AQC output}	qc=1: See section 3 for absolute QC output, if GCP are used
{CAL output}	cal=1: See section 5 for calibration output

### Examples:

```
> stripalign -corr -p out.ssrp
```

shortest command line, uses previously computed out.ssrp (containing file list), computes optimal corrections using default parameter values, writes corrected files, appends c\_ to file names to avoid overwriting.

```
> stripalign -corr -p -I results reg.ssrp -O final -olaz -qc -J -K3 -gcp mygcp.txt
```

uses previously computed reg.ssrp from directory results, computes optimal corrections using default parameter values, writes corrected files and all output to directory final, output format laz, performs relative and absolute QC using file mygcp.txt.

## 8. MULTICHANNEL CORRECTIONS FROM PARAMETER FILE (-corr -multi)

### Goals:

In a single step:

- Apply channel-specific calibration corrections
- Apply local corrections computed from reference channel to another channel

### Notes:

1. Calibration must be run for each channel separately, prior to correction
2. Registration parameters for reference channel have already been computed using -align, saved as .ssrp file
3. QC is not available with this command, needs to be run afterwards

### Usage:

```
stripalign -corr -multi -p <file> {input options} {PO options} {CORR options} [-R <a b c>]
[-o<[pre]*[suf].[ext]>] [-oparse <format>] [-nil]
```

### Options:

-p   param	Correction parameter file generated with -align (extension .sscp) for reference channel
{input options}	See section 2 for LiDAR file inputs and related options
{PO options}	See section 2 for position/orientation options
{CORR options}	See section 2 for calibration parameter correction options
-R   imu_body_rot <a b c>	IMU to BODY rotation, angles a b c about axes x,y,z (degrees) [default 0 0 0]
-o<[pre]*[suf].[ext]>	Output file name format: define prefix, suffix and extension [default: same name/format]
-oparse <format>	Parse string for input text LiDAR files [default: xyzt]
-nil	Do not write corrected files (run registration and optional correction comp.) [default: 0]

### Output files:

<name>.ssrp	Swath information file, short version of SSRP #H header - version number, information about number of swaths, pairs, area geometry #S swaths - columns: {swath# direction <sub>(CS)</sub> group# file_name}* {LOC output} Local correction file <name>.sscp, see section 6
-------------	---

### Examples:

```
> stripalign -corr -multi -p ch1.sscp -i *Channel_2*.laz -oname ch2 -sol file.sol -mnt_diff cal.mnt
```

uses previously computed parameters ch1.sscp (channel 1 data) and mount calibration corrections cal.mnt, uses inputs from channel 2, names them ch2, uses SOL trajectory data file.sol, writes corrected files, appends c\_ to file names to avoid overwriting; corrections are written to ch2.sscp with reference to ch2.ssrp for data file list.

```
> stripalign -corr -multi -p ch1.sscp -i *Channel_2*.laz -oname ch2 -pos file.pos -mnt_diff cal.mnt -int_diff cal.int -bore 0 0 -14
```

same but correct also internal parameters from file cal.int, and uses nonzero boresight 0 0 -14 corresponding to channel 2 of the Riegli Q1560 scanner; POS format trajectory file.pos is used.

```
> stripalign -corr -multi -p ch1.sscp -i *Channel_2*.laz -oname ch2 -sol file.sol
```

same as first example, but without calibration corrections, the local corrections from ch1.ssrp are simply duplicated and applied to the current channel data; trajectory file not used but required.

## 9. RELATIVE CALIBRATION (-calib -rel)

### Goals:

- Calibrate one channel with respect to another, using one target and one reference swath
- Perform relative quality control before/after registration for a single pair of strips

### Notes:

1. Only to be used with across-track looking instrument (use separate calibration for forward/backward scanners or when the boresight angles are not negligible, e.g. 1 deg or more)
2. Designed to work with different channels or sensors on same platform, but will also accept different trajectories (close, and parallel) flown in the same or opposite direction.

### Usage:

```
stripalign -calib -rel -i <file> -r <file> [-ch <n>] [-ch_ref <n>] [-iparse <format>]
          {PO options} {CAL options} [-qc {QC options}] [-x <v>] [-R <a b c>]
(expert)  {EXP options} [-frac_min <v>]
```

### Options:

-i   input <file>	Input LiDAR file (.las, .laz, .txt)
-r   ref <file>	Reference LiDAR file (.las, .laz, .txt)
-ch <n>	Reads only specified channel [0,3] (LAS 1.4 only) [default: 4 (all)]
-ch_ref <n>	Reads only specified channel for reference data [0,3] (LAS 1.4 only) [default: 4 (all)]
-iparse <format>	Parse string for input text LiDAR files [default: xyzt]
{PO options}	See section 2 for position/orientation options
{CAL options}	See section 5 for absolute calibration parameter options (apply to reference swath)
-qc {QC options}	Enable QC, see section 4 for QC options
-x   true_lever_x	True x-leverarm value [default: x-leverarm given with -mnt or -lever]
-R   imu_body_rot <a b c>	IMU to BODY rotation, angles a b c about axes x,y,z (degrees) [default 0 0 0]
(expert) {EXP options}	Expert options, see section 4
-frac_min <v>	Minimum fraction of area with sufficient point density for registration [default: 0.1]

### Output files:

<name>.mnt	Mount file for input swath, text format {x y z roll pitch yaw} (leverarm m, boresight deg) Sum of provided parameters and estimated differences.
<name>_rel.mnt	Results: relative mount parameters (difference between input and reference data)
<name>.log	Log file with swath information and computed parameters
{QC output}	QC output files except global z-difference map and swath list, see section 4

### Examples:

```
> stripalign -calib -rel -i ch2.laz -r ch1.laz -potxt traj.txt
```

simplest way to perform relative calibration, input ch2.laz and reference ch1.laz, using trajectory in text format traj.txt, and assuming the reference data has no boresight or leverarm error; the estimated relative boresight and leverarm are written to out\_rel.mnt; if the assumption is wrong the written out.mnt (absolute mount for ch2.laz) may be wrong, but the relative parameters are still right.

```
> stripalign -calib -rel -i file.las -ch 1 -r file.laz -ch_ref 0 -sbet file.out -feet -mnt ch2.mnt -x 0.2
```

same but both channels 1 and 2 are in the same LAS 1.4 file; using trajectory in SBET format file.out, data are in feet, mount parameters for input data are given by file ch2.mnt (from a previous calibration), and the x-leverarm for the input data is 0.2 m

```
> stripalign -calib -rel -i ch2.laz -r ch1.laz -pof traj.pof -qc -J -K2 -R 0 0 180 -oname ch2
```

same as first example, with POF trajectory file traj.pof, and an IMU/BODY rotation angle of 180 deg around the z axis; relative quality control files (histogram and z-difference JPEG image, color scheme 2) are written, all files have base name ch2

## 10. RELATIVE CORRECTIONS FROM PARAMETER FILE (-corr -rel)

### Goal:

- Apply local corrections computed from reference channel to another channel

### Notes:

1. Relative calibration must be run n-1 times for n channels, using same reference channel
2. Registration parameters for reference channel have already been computed using -align, saved as .ssrp file

### Usage:

```
stripalign -corr -rel -p <file> {input options} [-rel_mnt <file>] [-rel_bore <w p k>] [-rel_lever <x y z>]
[{{internal CORR options}}] [-qc {QC options}]
[-o<[pre]*[suf].[ext]>] [-oparse <format>] [-nil]
```

### Options:

-p   <file>	Correction parameter file generated with -align (extension .sscp) for reference channel
{input options}	See section 2 for LiDAR file inputs and related options (groups are ignored)
-rel_mnt <file>	Relative mount text file: x y z (leverarm) w p k (boresight)
-rel_bore <w p k>	Relative boresight angles, override file values [default: 0 0 0]
-rel_lever <x y z>	Relative leverarm, overrides file values [default: 0 0 0]
{CORR options}	See section 2 for calibration parameter correction options, internal parameters only If specified, override internal corrections from provided parameter file
-qc {QC options}	Enable QC, see section 4 for QC options
-o<[pre]*[suf].[ext]>	Output file name format: define prefix, suffix and extension [default: same name/format]
-oparse <format>	Parse string for input text LiDAR files [default: xyzt]
-nil	Do not write corrected files (run registration and optional correction comp.) [default: 0]

### Output files:

<name>.ssrp	Swath information file, short version of SSRP #H header - version number, information about number of swaths, pairs, area geometry #S swaths - columns: {swath# direction <sub>(CS)</sub> group# file_name}* Local correction file <name>.sscp, see section 6
{LOC output}	QC output files except global z-difference map and swath list, see section 4
{QC output}	

### Examples:

```
> stripalign -corr -rel -p ch1.sscp -i "*Channel 2*.laz" -oname ch2 -rel_mnt relcal.mnt
```

uses registration parameters for channel 1 (file ch1.sscp) to correct all files \*Channel 2\*.laz from current directory, with relative calibration parameters from file relcal.mnt computed using relative calibration.

```
> stripalign -corr -rel -p ch1.sscp -i "*.las" -ch 2 -oname ch2 -rel_mnt relcal.mnt -theta_so_diff 1.001 0
```

same but overrides internal scan angle scale factor (known value, or computed using absolute calibration with channel 2 data), and selects channel 2 using LAS 1.4 information, as all channels are in the same file.

```
> stripalign -corr -rel -p ch1.sscp -i "*.las" -ch 2 -oname ch2 -rel_bore 0.017 0.041 0.010 -qc -M -nil
```

same with explicit relative boresight angles, and runs a relative quality control, with text reports, histograms and z-difference map generation in TIFF, without correcting the data.

## A1. COMMON ERRORS MESSAGES, CAUSE, SOLUTION

### Nothing happens when I click on the icon...

Cause: normal behavior, this is a command-line software

Solution: run StripAlign in a terminal window (Cmd, ConEmu, etc.), please read the documentation!

### Executable not starting properly, no error message

Cause: antivirus software (e.g. Avast) may be blocking, or sandboxing "unsafe" or "unknown" executables

Solution: whitelist the application, or exclude the directory from scanning - please do not submit it to any virus scanning website, and use your antivirus to scan it instead.

### The application was unable to start correctly (0xc000007b)

Cause: missing or incompatible DLLs (Visual C++ Redistributable Packages for Visual Studio)

Solution: go to [www.microsoft.com/en-gb/download/details.aspx?id=40784](http://www.microsoft.com/en-gb/download/details.aspx?id=40784) and follow the instructions

### ERROR: stopping due to option argument errors (use -h for help)

Cause: some options have wrong argument types, or arguments are missing  
(unknown options will NOT cause errors)

Solution: check previous error messages and supply correct arguments

### ERROR: need more options, don't know what to do!

Cause: no valid command arguments found, or no arguments given

Solution: use -corr, -align, -calib or -qc, or a combination as explained above, depending on the task  
(or use one of the following options: -v -? -h -c -cleanup\_dir)

### ERROR: invalid output format, prefix\*suffix.ext required

Cause: syntax error or invalid characters \:?"<>|, or multiple '\*' and '.' detected

Solution: use simple names, without '.', for instance corr\_\*1.laz

### ERROR: coordinates look projected (geodetic expected)...

### ERROR: coordinates look geodetic (projected expected)...

Cause: trajectory or segment file provided not compatible with current projection setting (what is expected)

Solution: use -G0 for projected coordinates (UTM) and -G1 for geodetic coordinates (lon lat)

### ERROR: deviation from linear trajectory...

Cause: residual from trajectory segment computation exceeds SEGTRAJ\_TOL

Solution: use -traj\_tol <v> with an increased value, check the r=(..) displayed and current value of SEGTRAJ\_TOL

### ERROR: deviation from segmented trajectory...

Cause: residual from trajectory simplification exceeds TRAJ\_TOL

Solution: use -traj\_tol <v> with an increased value, check the r=(..) displayed and current value of TRAJ\_TOL

### ERROR: deviation from average direction...

Cause: angle between swath and average direction exceeds ANGLE\_TOL (swaths not sufficiently parallel)

Solution: use -angle\_tol <v> with an increased value, check the r=(..) displayed and current value of ANGLE\_TOL

### ERROR: missing both LAS timestamps and time interval VLR...

Cause: no time information with points, LAS format 0 (point cloud without time stamps): trajectory cannot be used

Solution: re-extract the swaths making sure time information is included

### ERROR: could not parse column format ..., expecting {...}

Cause: wrong column format specification for text file, most likely syntax error, duplicate or missing symbol

Solution: use all characters provided in {...}, with extra # for columns to ignore

example {xyz} -> #yxz for a 4-column or more file, with column 1 ignored, and y x z on columns 2 to 4

### ERROR: poor time ordering, swath ... (use option -t0)

Cause: points in LiDAR file not in time order (shuffled): too much disorder to use indexing for I/O operations

Solution: disable simple indexing set by -t (temporary files can be deleted with -c)

**ERROR: no overlap, swath registration failed**

Cause: swaths are too far apart, or not dense enough, or there is not enough data in overlapping regions

Solution: check relative quality with -qc -M and inspect z-diff maps; decrease the fraction area threshold -frac\_min

**ERROR invalid SSRP format, file ...**

Cause: wrong SSRP file version, created with an older version of stripalign

Solution: re-run the registration (-align) to generate a new file, or email us for file upgrade support@bayesmap.com

**ERROR: minimum one valid swath pair required**

Cause: in absolute calibration, no overlapping swath pair was found

Solution: check inputs: swaths should overlap, minimum number and configuration required, see 5. ABSOLUTE CALIBRATION

**ERROR: insufficient overlap or data quality**

Cause: in relative calibration, poor or no overlap, due to insufficient point density (below threshold) or excessive terrain roughness (above threshold), which leads to loss of data (e.g. over dense vegetated areas)

Solution: check inputs: swaths should overlap at least 10%, check quality with -qc, check thresholds set by options -density\_min and -rough\_max or default values if options are missing

**ERROR: boresight angles too large for ... calibration**

Cause: relative or drift calibration algorithms are limited to instruments with small boresight angles

Solution: perform absolute calibration separately for each channel, then apply boresight corrections to all the data

**Total number of points looks incorrect, mismatch between numbers displayed during different tasks**

Cause: not all points are used in the computations (non-last returns are discarded, and points can be filtered), however all points are corrected with the -corr option.

**ERROR: unsupported SOL file format, or unsupported SOL file timestamp format, file ...**

Cause: SOL files should contain post-processed smoothed trajectory and attitude data, and GPS timestamps only

Solution: check the output options in the trajectory processing software

**ERROR: cannot index swath, or failed to index swaths**

Cause: there is a mismatch between LiDAR data and trajectory coordinates (e.g. different projection and/or units)

Solution: provide a trajectory projected in the same frame as the data and with same unit, use -feet if necessary

**ERROR: group size mismatch, x PO groups and y swath groups**

Cause: a different number of swath groups (-i and -r options) and PO file groups was passed

Solution: each specified group of swaths must have a corresponding group of PO files for correct GPS week handling

**ERROR: cannot use both generic and scale/offset internal scanner parameters**

Cause: -eps\_diff -mu\_diff or -int\_diff used at the same time as -theta\_so\_diff or -range\_so\_diff

Solution: use only one model (scale/offset, allowed for uniform corrections only) or generic (with eps, mu)

**ERROR: missing or duplicate input files, cannot proceed**

Cause: files not found, or duplicate file names entered by mistake, or forgotten to enter input and reference channels, or identical input and reference channels

Solution: check inputs, specify input and reference channels correctly if all channels are in the same LAS 1.4 file

**ERROR: missing extended scanner channel ...**

Cause: extended scanner channel information not present in the file, or is in a currently unsupported format

Solution: use LAS 1.4, and point format 6 or greater, otherwise channel is not stored

**ERROR: swaths have different type of GPS time within group n**

**ERROR: PO files have different type of GPS time within group m**

Cause: some of the files have adjusted GPS time, other week seconds timestamps, within the same group n or m

Solution: split files into different groups depending on GPS timestamp type, if possible

## A2. CURRENT LIMITATIONS

- **Trajectories must be mostly straight (no data collection during turns, or on curved trajectories)**
- **Alignment and QC: flight lines must have an approximately parallel pattern**
- **UAV settings currently not supported (small-scale surveys, and curved, non-parallel trajectories) (except for calibration, accepting non-parallel straight lines at any scale)**
- **Multi-part swaths (flight lines split across two or more files) not supported, need to be merged before processing**
- **Complex topology not supported, may produce unexpected results (e.g. interleaved flight lines in a subregion) and requires manual multi-step processing**
  
- Input/output LiDAR file format LAS, LAZ, TXT (TXT only basic information)
- Input trajectory and attitude data format SBET, SOL, TRJ, POF, TXT or segment
- World files provided as an option with TIFF/JPEG z-diff maps; no GeoTIFF support yet
  
- Alignment: boresight angles must be small (across track-looking), otherwise boresight errors have to be corrected first to achieve best accuracy
- Alignment: artifacts may remain with forward/backward-looking instruments, depending on IMU quality
- Trajectory given in geodetic coordinates can only be projected to UTM (if LiDAR is not in UTM, a projected trajectory file is required, using the same projection as the data, vertical datum included)
- Multi-beam, multi-channel and multi-instrument data should be processed separately for each beam/channel/instrument, especially for calibration, or at least grouped by similar geometry
- Subsampled trajectories may not be processed correctly; original data should be provided
- GCP data can only be used to globally constrain z (uniform z shift)
- Registration uses solid surfaces; fuzzy objects (canopy) or complex topology (vertical surfaces, wires, power lines) will be removed automatically and may cause artifacts due to loss of data
- Large areas with dense vegetation slower to process; loss of data due to roughness-based filtering may cause artifacts
- Large scan angle scale (+-100 ppm or more) may cause unreliable boresight calibration when using perpendicular or diagonal lines
- Boresight calibration currently only estimates scan angle scale, while other functions use a generic internal model with two parameters eps and mu (though eps is strongly related to the scan angle scale)
- Large scan angle scale (+-100 ppm or more) can cause artifacts in local corrections
- Trajectories are required for uniform corrections, other tasks can use segments (.seg text files)
- Some unused options may not generate warnings
- Unexpected behavior regarding log files may occur if running multiple instances of StripAlign in parallel
- Multithreading limited to most time consuming functions, not optimized for small regions
- Multithreading of corrections, QC and heavy I/O tasks limited to 2 threads
- No GPU support yet