



DTM PRODUCTION TEAM
UNIVERSITY OF ARIZONA, TUCSON, ARIZONA

AutoTriangulation Instruction Manual

Authors:

Aaron KILGALLON, Jon STEPHENS, Sarah SUTTON

Contact:

ssutton@pirl.lpl.arizona.edu

Contents

1	Introduction	1
2	Program	1
2.1	Command Line Arguments	1
2.2	Inputs	1
2.3	Outputs	2
3	Algorithm	4
3.1	Filtering	4
4	Using the Results	5
4.1	DTM and Laser Altimetry Quality	5
4.2	Applying the Updated Ground Point Coordinates	5
5	Installation	6
5.1	Machine Requirements	6
5.2	Installation Procedure	6
6	References	6

1 Introduction

AutoTriangulation is a tool used to perform spatial adjustment of HiRISE [1] or LROC [2] Digital Terrain Models (DTMs) to align them to MOLA [3] and LOLA [4] data, respectively. This tool is designed to be used within the SOCET SET™ (©BAE Systems, Inc.) work flow for updating pre-existing XYZ-control points in SOCET SET's Multi-Sensor Triangulation (MST) program. It is designed to augment, or possibly replace, the need for the human operator to determine control intuitively. *AutoTriangulation* automates the steps required to adjust the stereo model to the laser data and minimizes the error by iterating through the adjustment steps.

2 Program

2.1 Command Line Arguments

The main program requires three input files as well as various parameters in order to determine an alignment solution.

- geotiff
- laser data
- report file

The first input file is a DTM in geotiff format. The geotiffs tested have been exported from SOCET SET, and have a size limit of 4GB. The second input file consists of MOLA or LOLA track data in three possible file formats: database files (.dbf part of the shapefile), tabulated files (.tab), or comma separate value files (.csv). These files are obtained from the PDS Geosciences Node Orbital Data Explorer (<http://ode.rsl.wustl.edu>) or from queries of internally stored databases (i.e. PEDR2TAB.pl). The third input file is the report file (.rep) generated after running a solution in MST in SOCET SET. **It is critical that the report file corresponds to the solution used to generate the DTM.** Note that this tool currently only works for projects in an equirectangular projection.

2.2 Inputs

Begin by generating a geotiff of the DTM to be aligned, and ensure that there are no blunders or artifacts in the DTM that may compromise the results from *autoTriangulation*. Run the main program by using the following command:

```
autoTriangulation <DTM geotiff> <laser track file> <MST .rep file> [options]
```

Table 1: Command Line Options Table

Command Name	Default Value	Command Line Argument Type
Maximum Number of Stages	20	-s %d
RMS Convergence Value	0.0001	-c %f
Translation Increment (degrees)	0.001	-d %f
Translation Range (degrees)	0.035	-p %f
Linear vs. Quadratic Polynomial Fit	Linear Fit	-lf or -qf
HiRISE vs. LROC Project	HiRISE Projects	-h or -l
LROC filtering (LROC only)	Off	-f
Laser Orbit Filtering	None	-of %d %d %d %d ...
Modify Rater's Top Left Coordinate	None	-tl %f(x) %f(y)
Outlier Point Error Threshold (meters)	30	-et %f
Statistical Filtering Dec. Cutoff (meters)	2	-sfc %f
Statistical Filtering Max Error (meters)	10	-sfe %f
Statistical Filtering Decrease/Stage (meters)	10%	-sfd %f
Statistical Filtering Standard Deviation (meters)	7	-sfsd %f
Statistical Filtering Remove Percent	5%	-sfp %f
Statistical Filtering Starting Stage	maxStage/4	-sfs %d
Add String to Output Filenames	Empty String	-as %s
Debug Mode	Off	-dm

Note that the three input files can be entered in any order. The command options are shown in Table 1.

AutoTriangulation can also be used to generate a standalone error map during manual triangulation. Enter the MOLA track file and the DTM geotiff with no .rep file, and the program will output an errormap for that given state.

Another method of running *autoTriangulation* is by using a standalone Python file called *importCoords2GPF*. This file acts as both an *autoTriangulation* wrapper script and as a way of applying the solution in SO CET. If only the *autoTriangulation* report file and the project GPF file are entered, the script will automatically update the SO CET GPF file. If all the *autoTriangulation* files and options are entered, the script will run *autoTriangulation* and then update the GPF file. To run the full wrapper version, run the following command. To only update the GPF file without running *autoTriangulation*, see section 4.2.

```
python importCoords2GPF.py <ProjectName> <file1> <file2> <file3> [options]
```

2.3 Outputs

AutoTriangulation outputs four primary files, in addition to several support files. The first file is the *autoTriangulation* solution report file, denoted with the DTM file name followed by `_report.txt`. This file gives the coordinates of the adjusted control points in a list. The coordinates for appropriate control points can be copied from this file into SO CET SET and solved upon in MST. Following these coordinates is a list of the orbit statistics. The orbit number, statistical weighting, point count, mean error, and standard deviation values of each track used in the solution are listed. These statistics are useful for detecting outliers and are helpful for understanding the solution to the tracks.

The second file is the *autoTriangulation* error map. This file is denoted by the DTM file name followed by `_Error_Map_Filtered.png`. This gives the user a very helpful graphical explanation of the solution calculated by the program. The left panel shows the initial differences between the laser shots and the terrain model. The right panel shows the predicted error map based on *autoTriangulation's* final solution. Statistics and a color range are included to help the user.

The third file is the statistical filtering weighting map. This map displays the MOLA or LROC tracks color-coded by weighting values used in the statistical calculation. These weightings are a metric that describe the effect of each track on the final solution.

The fourth file is a control coordinate map. This illustrates the location of each control point in their initial and final states, along with a bounding box to demonstrate the position of the DTM. This enables the user to understand

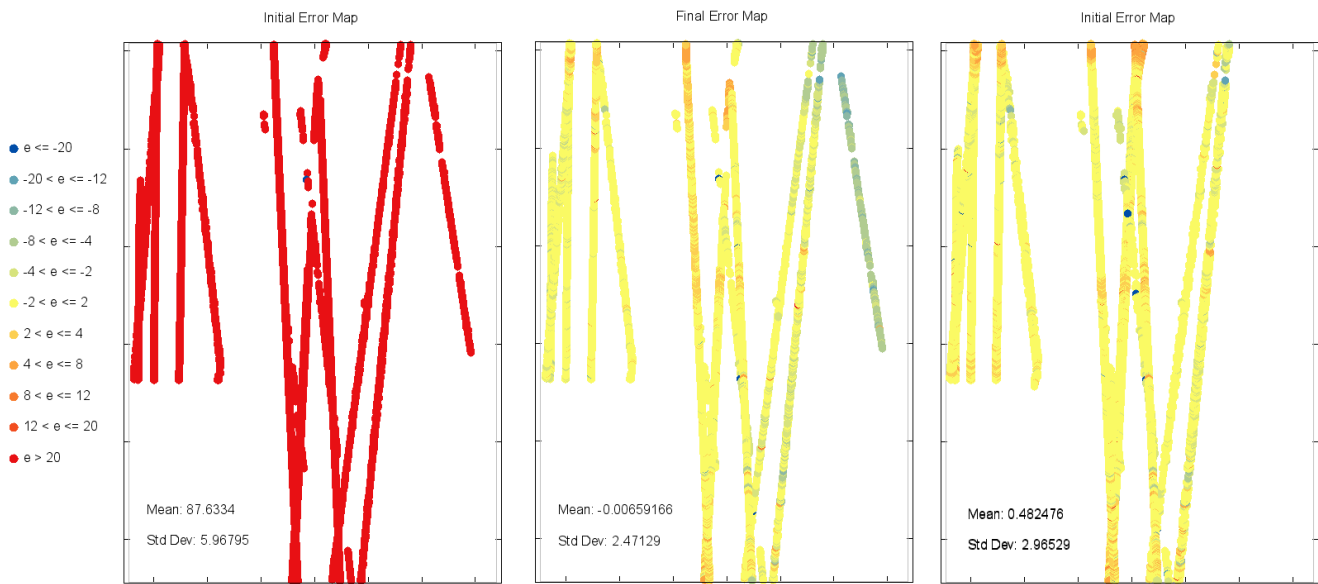


Figure 1: On left: Initial Error Map, Middle: Predicted autoTriangulation solution, Right: Actual result in SOCET SET.

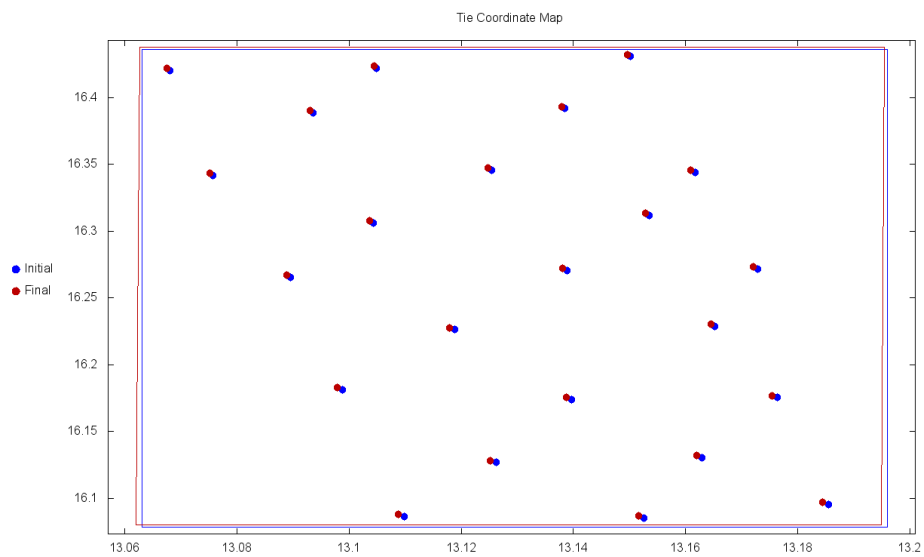


Figure 2: Initial and predicted final coordinate map.

the general translation and rotation applied to the initial solution to derive the final solution. In general, too large a translation or rotation is probably not a valid solution. The translation increment might need to be decreased, or there might not be sufficient laser track coverage to arrive at a solution that makes sense. An example of this is shown in figure 2.

Additional support files are the GnuPlot (.gnu) files which are saved before the map files (.png) are generated.

3 Algorithm

AutoTriangulation applies an iterative, multi-staged approach to deriving the final solution for the DTM. It utilizes a modified least-squares minimization routine that converges to a solution with an average error across the DTM as close to 0 as possible. The program has several internal functions that are implemented with each stage: x-y translation, a polynomial fit and subtraction to the errors across the DTM, and a rotation function. The algorithm is not guaranteed to converge to the global solution, but the parameters of the solution have been adjusted to yield the best results.

- Each stage begins with translating the DTM in the x-y (longitude-latitude) direction. The DTM is translated over a grid of potential positions, and the root mean square (RMS) of the error is calculated for each solution stage. The minimum over all the potential RMS values is selected as the best fit for the x-y position of the DTM and the initial state for the succeeding stage.
- A polynomial fit is applied to the set of errors between the DTM and the laser data. This polynomial fit can be linear or quadratic, but defaults to a linear fit. This polynomial is subtracted from the raster, thereby ensuring that the average z errors are accounted for as well as rectifying any overall tilt to the DTM. A second order polynomial fit proves to be most useful when a sensor model issue in SOCET SET results in a large warp or curvature in the DTM.
- The raster is then rotated through a set of potential rotations in the x-y (longitude, latitude) plane. This range of angle values can be selected by the user, but defaults to $\pm 5^\circ$ in increments of 0.1° . The correct angle of rotation is the respective angle that brings about a minimization of the RMS values. A polynomial fit is then applied again to the DTM.
- The first three steps are repeated a number of stages selected by the user (defaults to 20 stages). With each stage, the grid of potential x-y translation positions is decreased in its overall space of search, allowing the program to converge to a final solution at the completion of the total number of stages.

The program defaults to various set parameter values for certain types of projects inputted. Projects associated with LROC projects do not have any rotation applied to the solution.

For each call, n , of the function, the value of the RMS is calculated using equation 1. N denotes the number of laser tracks lying above the DTM, z_i denotes the elevation value of each laser track i , and d_i denotes the elevation of the DTM at the same x-y coordinate of each corresponding laser track. $w_{i,track}$ is the weighting of each track, which is given the following range: $0 \leq w_{i,track} \leq 1$.

$$RMS = \sum_{i=0}^{N_n-1} w_{i,track} \frac{(z_i - d_i)^2}{N_n} \tag{1}$$

3.1 Filtering

Several statistical filtering methods exist specifically for LROC projects. The program argument $-f$ will initiate filtering of the LOLA data by emission angle, phase angle, and shot number with the following constraints:

Table 2: LROC Filtering Constraints

Parameter	Constraint
Emission Angle	$0^\circ \leq \phi \leq 15^\circ$
Phase Angle	$20^\circ \leq \theta \leq 70^\circ$
Shot Number	3, 5

The filtering of the emission and phase angle is for aggregate quality improvement, whereas the the shot filtering is implemented to ensure that the program runtime for LROC projects is acceptable. Shot filtering will select only the 3rd and 5th shots from each track to use in the program's algorithm, as this greatly reduces the runtime of the program for the very dense LOLA data.

Here, we consider that purely random shot noise is insignificant in the overall solution and in the algorithm's approach, however, inconsistent tracks are very problematic in the program's ability to converge to the best solution. In order to reduce the effect of outlier data on the program's final solution, a weighting system was implemented for each track. Each track is given an initial weighting, and user-specified parameters determine the thresholds and rating values for reducing the weightings of outlier tracks.

The parameter *Statistical Filtering Max Error*, denoted by *-sfe*, dictates the threshold error of a particular track needed for its weighting to be reduced. A similar parameter called *Statistical Filtering Max Standard Deviation*, denoted by *-sfd*, is the threshold value in terms of the standard deviation of the errors for each laser track for weighting reduction. The quantity of this track's weighting reduction is the parameter *Statistical Filtering Remove Percent*, denoted by *-sfp*. This value dictates the percentage of weighting that is reduced by each stage, if the track is above either of the threshold values for track average error or track error standard deviation. An additional argument is the parameter *Statistical Filtering Starting Stage*, which indicates the stage at which this filtering will begin to be used within the algorithm.

4 Using the Results

There are several ways of applying the results from *autoTriangulation* to update your triangulation solution. These depend on the quality of the DTM and the laser data. We find that *autoTriangulation* outputs the best results when the topography is relatively flat, and there are abundant laser data. This is due to the fact that our algorithm does not currently guarantee convergence globally.

4.1 DTM and Laser Altimetry Quality

The input DTM should be free from major artifacts and should have a sufficient number of laser data points as well as a good distribution of those points (≥ 2 laser tracks). This is generally not a problem for LOLA data, but certain HiRISE DTMs may not have sufficient MOLA coverage. Visual inspection of the laser data in SOCET SET can show the existence of outlier laser tracks or orbits. These should be removed from the solution by either using the Orbit Filtering parameter in *autoTriangulation*, removing the track from SOCET SET and then re-exporting the laser data, or by adjusting the statistical filtering parameters in *autoTriangulation* to account for these tracks.

4.2 Applying the Updated Ground Point Coordinates

One method for applying the *autoTriangulation* solution in SOCET is by copying the coordinates from the output report file into Interactive Point Measurement in SOCET SET and then updating the point type (tie point, Z-control, XYZ-control, etc.) and the accuracy values. However, we also provide a standalone Python script called *importCoords2GPF* that will automatically update the Ground Point File for your chosen SOCET SET project. The script will update the Ground Point File with the updated coordinates for each control point automatically. It will not update the point type, but will update the accuracy values for any pre-existing points. Note that the path to your SOCET SET directory will need to be updated in the Python file.

Run the Python file to update the GPF coordinates using the following command:

```
python importCoords2GPF.py <ProjectName> <autoTriangulation report file> <Project GPF file>
```

Note that the solution from *autoTriangulation* does not necessarily reflect what will occur in SOCET SET after updating the XYZ and Z coordinates. This is largely due to the fact that SOCET SET is solving for the least squares solution within a set of parameters and weights that are not considered in *autoTriangulation*.

In general, selecting 1 to 3 XYZ control points is ideal. Any number of the remaining tie points can be converted to Z control to control tilt. For flatter DTMs with consistent laser data, we find that applying a large number of the control points to the SOCET SET solution, with tight accuracies, generally results in the best solutions. For DTMs with more variation in topography, applying fewer control points to the SOCET SET triangulation solution seems to result in better overall solutions. Due to the variability in projects, manual editing of the control points may still be required. Trial and error seems to be a good approach to applying these coordinates.

5 Installation

5.1 Machine Requirements

Operating System: Windows XP - Windows 8 (Tested on Windows 7)

Recommended Hardware:

RAM: If *autoTriangulation* is to be run GeoTiffs greater than 2 GB, the machine should have over 8GB of RAM to run in an appropriate amount of time.

5.2 Installation Procedure

Installing *autoTriangulation* is fairly straightforward. To begin installation run the setup.exe file provided. This setup.exe file will install *autoTriangulation* and GnuPlot executables and add those executables to the machine's path. If another version of GnuPlot is already installed on the machine, the path can be modified to remove the version installed with *autoTriangulation* but do not delete or move GnuPlot from its location. It is also worth noting that *autoTriangulation* will create .gnu files which are configuration files for gnuplot to create the various maps, and are guaranteed to run on the version of GnuPlot provided with *autoTriangulation*; however they may not run on earlier versions.

After running the setup.exe file, the machine will be checked to see if it has the Microsoft Visual C++ 2013 Redistributable Package (x86) already installed, if it does not the setup program will prompt the user to install the redistributable package by clicking Install. The redistributable package should quickly install on the machine and once it is done the main installation window will appear.

Once the machine has the correct redistributable package, the main setup window will appear. To begin installing *autoTriangulation* click next. The next page will prompt the user to enter an installation location on their machine. It is automatically set to be installed in the Program Files folder, however, the location can be changed using the change button. If the path name is changed, be sure that the length of the path is less than 250 characters. Once the path is set to the appropriate place, click Next. On the next page, the installation settings can be reviewed and if everything is satisfactory, the installation can be started by clicking Install. A screen should pop up and show the progress of the installation with a progress bar, and finally end with an Installation Wizard Complete window. Once this window is shown, *autoTriangulation* should be installed and the path should be set on the machine as was specified.

To update *autoTriangulation*, download the new setup.exe file. Before running it, you should uninstall the current version of *autoTriangulation* using the Windows Control Panel.

6 References

- [1] A. S. McEwen, et al. (2007) JGR-Planets 112:E05S02.
- [2] M. S. Robinson, et al. (2010) Space Sci Rev 150:81.
- [3] D. E. Smith, et al. (2001) JGR-Planets 106:23689.
- [4] D. E. Smith, et al. (2010) Space Sci Rev 150:209.