**Fault Mapping Algorithm developed for: A novel tool to map normal fault geometry and scarp height from topographic data: Application to Volcanic Tablelands and the Hurricane Fault, Western US**

**To be submitted to Lithosphere in July 2021**

Algorithm developed by Chelsea Scott in collaboration with Isabelle Manighetti2, Ramon Arrowsmith1, Frédérique Leclerc2, Cassandra Brigham3, Tiziano Giampietro2, Lionel Mattéo2, and Daniel Lao Davila4

1Arizona State University, 2Université Côte d'Azur (France), 3University of Washington, 4Oklahoma State University

Here, we present our algorithm which maps topographic fault scarps and calculates scarp height.

**Software:**

The algorithm runs on Matlab.

Lasdata: This is available from the Matlab file exchange and is used to read the las files. <https://www.mathworks.com/matlabcentral/fileexchange/48073-lasdata>

Deg2utm: This is available from the Matlab file exchange and is used to convert the latitude and longitude in geortif files to UTM. https://www.mathworks.com/matlabcentral/fileexchange/10915-deg2utm

**Algorithm Input:**

Manually-drawn fault maps in the calibration zone: This calibration map is used to calibrate the fault mapping parameters (and ). The calibration map should be done at the desired scale of the fault mapping (e.g., fine or coarse). The mapping can be done at the top, steepest location (approximate middle), or bottom of the scarp, and this location must be specified in the algorithm. The fault map is expected to be save as a shape file.

Box bounding the area of the calibration map: This should be a polygon shape file that bounds the area to be used for the calibration.

Topography datasets: The algorithm accepts las files (typically used for point clouds) and geotiffs with horizontal coordinates in meters or latitude/ longitude.

Everything should be saved in the same coordinate reference system (CRS). Las files are assumed to be saved in meters. Geotiffs can be in meters (i.e., UTM grids) or lat/long (e.g., SRTM) but this information must be specified in open\_data (see below). Elevations are assumed to be in meters.

**Algorithm options:**

The master script is “Scarp\_height\_code\_master”. There are several options and variables that must be set prior to running the code.

**In opendata:**

*Data\_type*: 'las' (point cloud in las format), 'tif\_ll (geotif in lat/long)', 'tif\_UTM (geotif in UTM)'

*fault\_mapping*: options include 'top', 'steep', 'base' to indicate which part of the fault was mapped

*Shape files*: test\_polygon (outline of the area where the calibration mapping was done) and faults (calibration fault map). This can also include other lineations such as streams if the user would like these to be included in the calibration)

In “Scarp\_height\_code\_master”,

swath\_width is the width of strike-perpendicular swaths. We recommend 5 m for sUAS, lidar, Pleiades and 90 m for SRTM

slope\_dist is the moving window size used to calculate the topographic slope. Ideally, this dimension is at least several pixels. We recommend 2 m for sUAS, lidar, Pleiades and 90 m for SRTM

Calibration parameters: Ideally many parameters should be tested as a too small sample size may miss the correct parameters.

Here is a possible set of recommendations, although these are likely to vary based on the faulting and climate. To decrease run-time, we suggest testing parameters with a large step size over a large range and then to focus in better-fitting areas. The algorithm can find local minima if a sufficient set of parameters are not searched.

sUAS, lidar, Pleiades: scarp\_slope\_test=0.01:.04:.75; scarp\_dist\_test=1:4:75;% (this tests an extensive set of parameters, which is a good place to but may not be needed for all projects.)

SRTM: scarp\_slope\_test=0.02:0.02:4; scarp\_dist\_test=30:30:540

The algorithm generates several files with the results. save\_name indicates the base name for the saved files.   
save\_name='lidar\_southern';

Plot\_yes (1 for yes, 0 for no) will plot the intermediate results. These are good to look at but be aware that many figures are generated.

**Output:** The following information is save in “save\_name”\_best.mat

x\_alg: x position in the rotated coordinate system of the faults  
y\_alg: y position in the rotated coordinate system of the faults  
x\_alg\_UTM; x position of the faults in UTM   
x\_alg\_UTM; y position of the faults in UTM   
height\_alg: fault height in meters  
error\_fault: counts, std error, lower bound error, upper bound error, all in meters  
scarp\_slope\_best: best scarp parameter for mapping   
scarp\_dist\_best: best distance parameter for mapping  
list: y points of measurements  
average strike of faults as indicated by the user.

The following information is save in “save\_name”.mat

A single matrix with the output of “save\_name” and the list of tested scarp length and slope parameters.

**Demo examples:**

We provide two examples for running the fault mapping algorithm.

*Southern test zone, Bishop with airborne lidar topography:* The fault maps and bounding box are in the folder southern\_test\_zone. The mapping was done at the top of the scarp. The topography dataset can be generated by running this job at OpenTopography.

https://portal.opentopography.org/lidarDataset?opentopoID=OTLAS.042016.26911.1&minX=-118.48879812336241&minY=37.42271607442166&maxX=-118.4846703469309&maxY=37.42629217632438

*Hurricane fault, Arizona with SRTM topography:*

The fault map and bounding box are in hurricane\_fault. The mapping was done at the base of the scarp. The topography dataset can be generated by running this job at OpenTopography https://portal.opentopography.org/raster?opentopoID=OTSRTM.082015.4326.1&minX=-113.4549222986141&minY=35.7624929247002&maxX=-113.075506121718&maxY=36.88965004473896