LA-UR-20-29050

Stereo-tomography instructions

Introduction:

The python code performs calculations and visualizations to aid in the three-dimensional interpretation of TEM/STEM (transmission imaging) micrographs. This is based on the parallax concept. Features in the micrographs are treated as points or a series of points. The coordinates of the points (in three dimensions) are calculated with respect to each other.

Citation:

If presenting results that utilize this software, cite as:
B.P. Eftink and S.A. Maloy, *Obtain3D_open,* Los Alamos National Laboratory (2020).
In addition, if using the Mayavi visualization include the citation:
Ramachandran, P. and G. Varoquaux, Mayavi: 3D Visualization of Scientific Data. Computing in Science & Engineering, 2011. 13(2): p. 40-51.

Instructions:

The python script Obtain3D_open.py is run in a Python 2 environment. For the optional visualization of the models, the Mayavi package should be installed. All the input files (see Section 1), should be placed in the same folder, running the program will prompt the user to select the proper folder (the folder is the working directory). Examples are provided in github as different branches of the project.

Input files:

There are three input files that are used to run the program. in.txt and key.txt are necessary, part.txt is optional. Files 1 and 2 are required while file 3 is only necessary for the visualization of spheres (cavities, precipitates, etc…).

1. in.txt
2. key.txt
3. part.txt

File 1 "in.txt":

File 1 is a tab delineated text file in which the columns in order represent

i. X_1
ii. Y_1
iii. X_2
iv. Y_2
v. Point type
vi. Point number

The x_1 and y_1 represent the pixel coordinates in the first image of the points. In the case of this script, positive x is in the direction to the right and positive y is in the down direction.

The x_2 and y_2 represent the pixel coordinates in the second image of the points.

The “point type” instructs the script as to how to visualize the point.

	“point type”
Spheres	0
Dislocations	1-4999

Spheres are visualized individually and all take the “point type” of 0.

Dislocations are made of a series of points, each individual dislocation should use the same number for the “point type” for the series of points. For example, all of the points composing one dislocation could use 23 for the “point type” and a second dislocation could use 5 for the “point type” while a third dislocation uses 4950 for each of the points comprising the dislocation. The points of the dislocation are connected by lines. The order of the points is important as the lines will connect from the first listed through each point of the same “point type” to the last point.

In the sixth column, “point number” is the number corresponding to each point starting with 1. For example 1,2,3,4,5…

File 2 “key.txt”:

The “key.txt” file provides the necessary information for the script to run properly. The user inputs are in the second column on different rows.

Rows:

1. X-tilt (of the first image)
2. X-tilt (of the second image)
3. Y-tilt (of the first image)
4. Y-tilt (of the second image)
5. Scale (pixels per nanometer)
6. Interfaces (keep value at 0)
7. Particles (a value of 1 if spheres are to be visualized, 0 if not)
8. View (1 if viewing in Mayavi, 0 if only text output)
9. Number markers (1 if the points are to be labeled by numbers on the visualization, 0 if not.)
10. Movie (1 creates saves frames for a video, 0 does not. View must also have a value of 1 for saving video frames)

File 3 “part.txt”:

This text file is optional but contains information for visualizing spherical features. The input information is done for each spherical point and contains three columns:

1. The number of the point
2. The size (diameter) of the sphere (in pixels)
3. The type (used for color. 1. Green 2. Black 3. Purple 4. Red 5. Yellow)