\*\*These instructions are for the purpose of demonstrating step-by-step details on how to run the Shoreline-Mapping-Toolbox created by Mitch Harley.\*\*

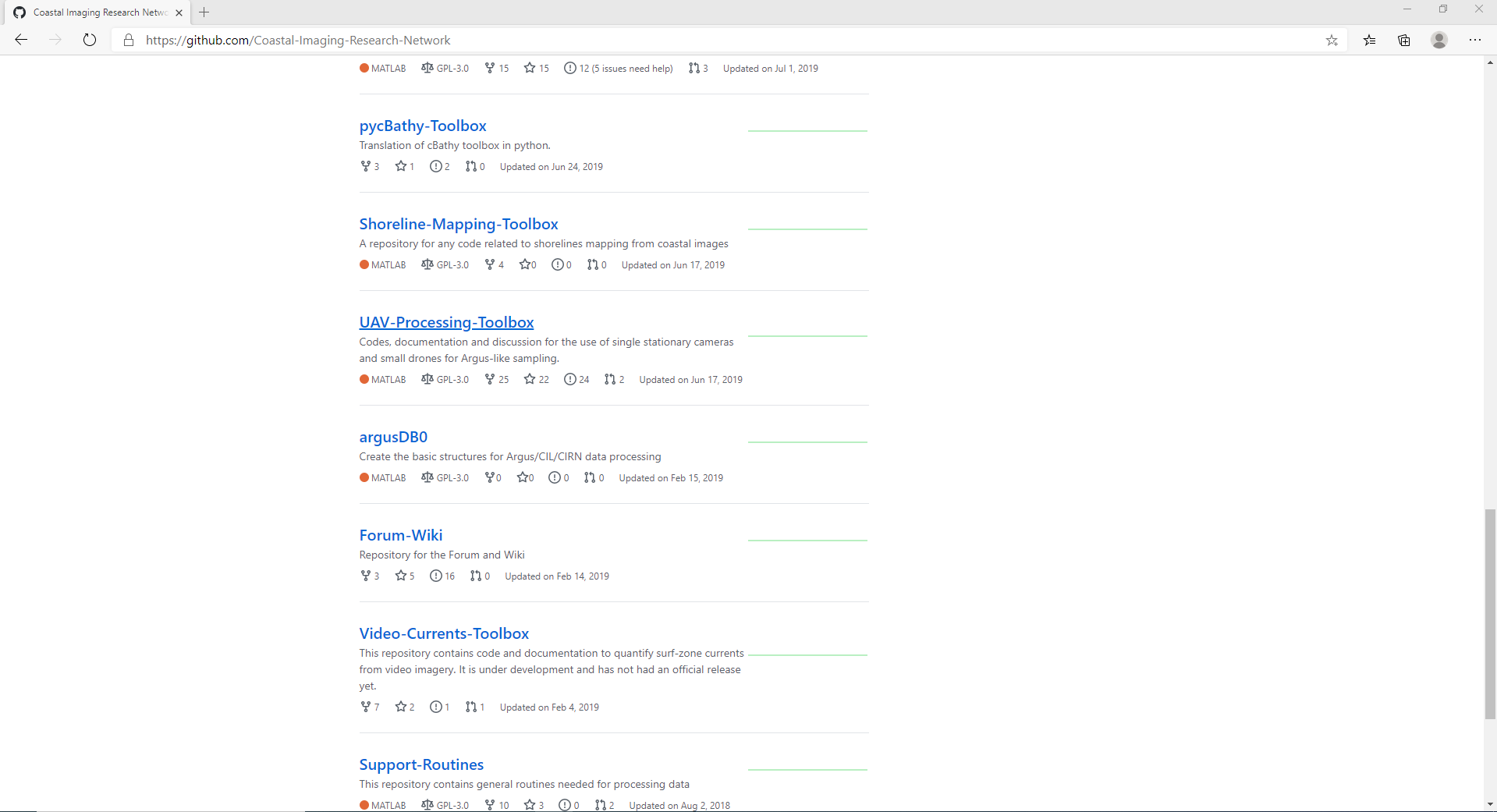
# MATLAB Requirements:

Image Processing Toolbox

* You can validate that you have the Image Processing Toolbox by typing ver in the Command Window

## Downloading the Toolbox

Locate the Shoreline-Mapping-Toolbox in the CIRN GitHub repositories



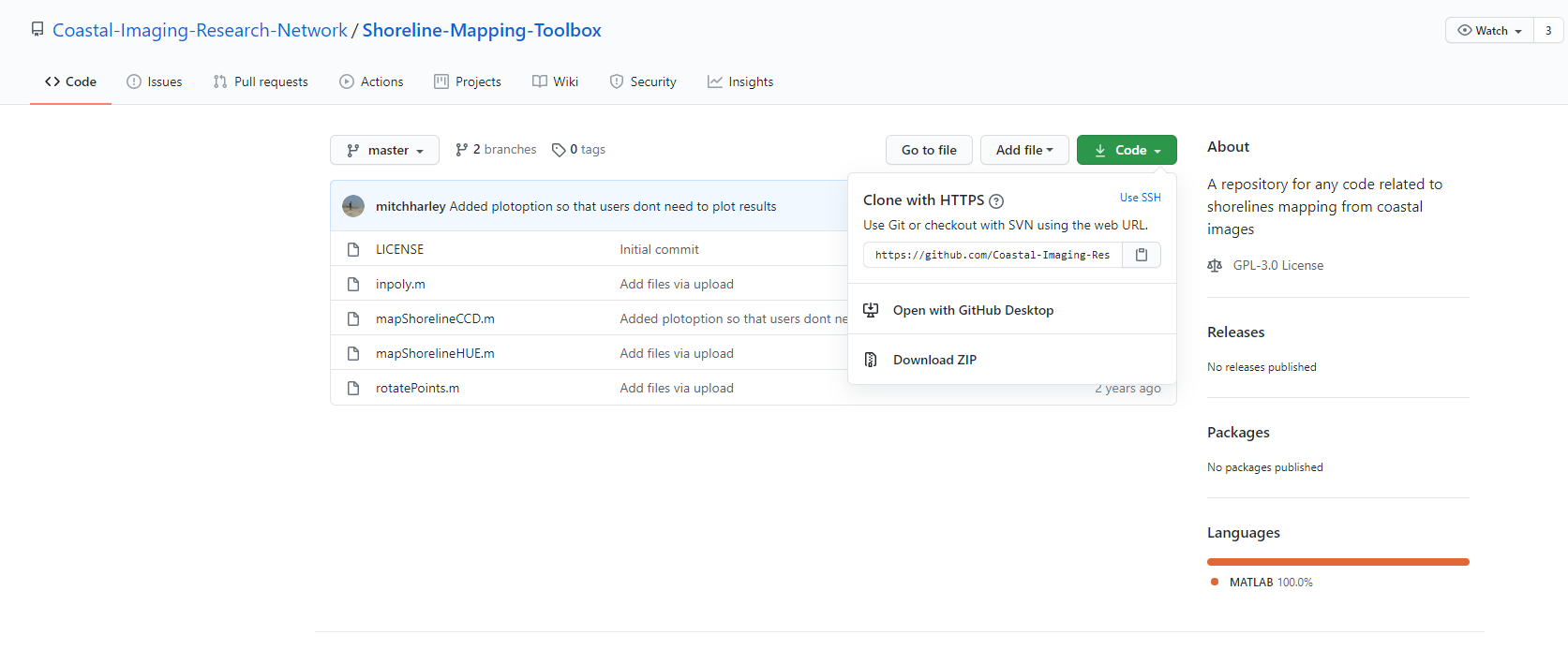
12111hmmn1

2

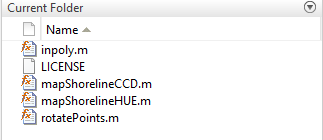
1111hmmn1

1

Clone the repository using Git or download and unzip it.



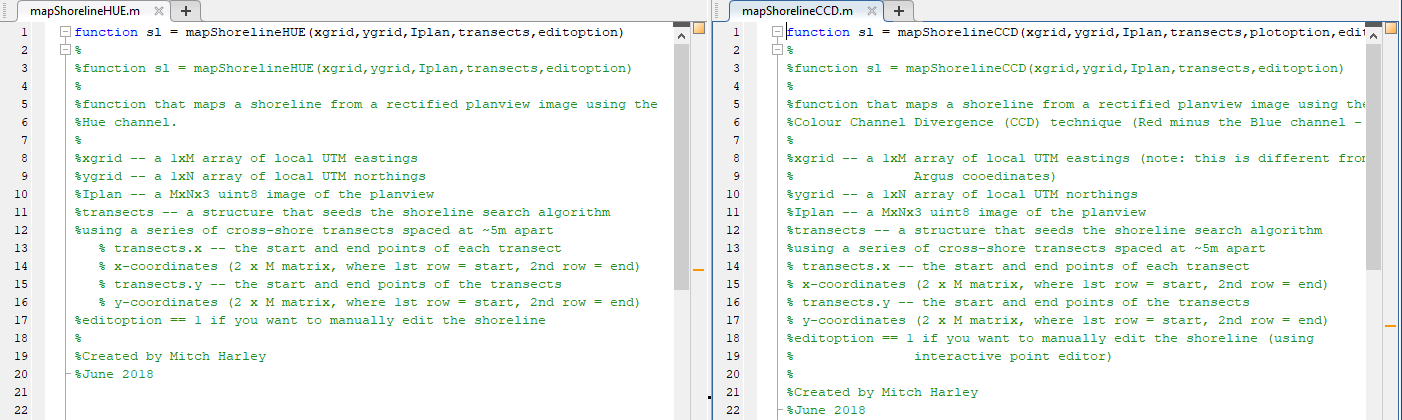
After you clone the repository you will find these files in the Shoreline-Mapping-Toolbox directory.



## Preprocessing

*Prior to running the codes in the Shoreline Toolbox, there are several steps that need to be taken in order to ensure a smooth process.*

The mapShoreline\_... codes have prerequisites for the inputs in the functions. See figures below.



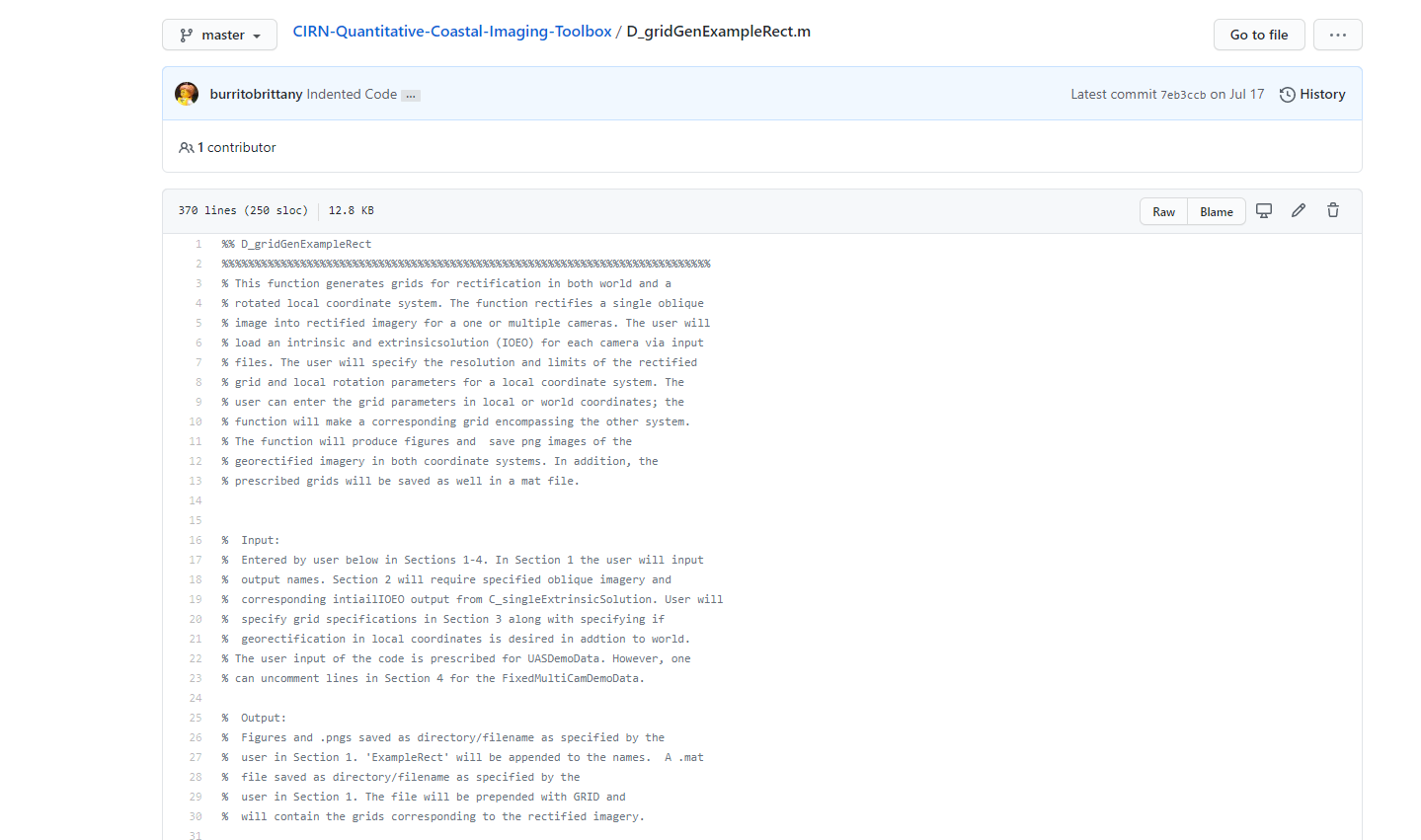
1. Images must be rectified in plan view
2. The coordinates must be in UTM eastings and northings or local rectilinear grid-oriented north-south
3. The transects may require manual alteration to adjust for the moving shoreline.
4. **Rectifying images**

*In order to run the mapShoreline\_... codes, you must have the images rectified in the planar view.*

There is more than one way to do this, but the following is what worked for this particular situation.

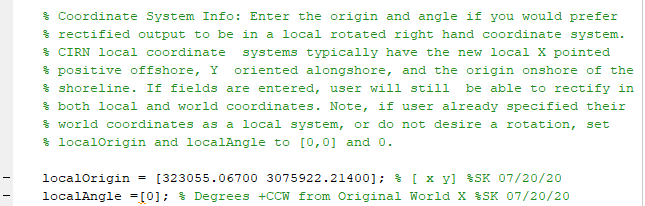
Go to the CIRN GitHub and clone or download the [CIRN-Quantitative-Coastal-Imaging toolbox](https://github.com/Coastal-Imaging-Research-Network/CIRN-Quantitative-Coastal-Imaging-Toolbox):

The D\_gridGenExampleRect.m will be the code you will want to use. D\_gridGenExampleRect.m requires an IOEO calibration.mat file, which is created in C\_singleExtrinsicSolution of the CIRN-Quantitative-Coastal-Imaging toolbox.



It is important to note that you may need to change the image resolution (around line 172 in D\_gridGenExampleRect; it is labeled idxdy).

**\*\*In order to rectify in UTM (grid oriented north-south) and not rotate the image into a shore normal coordinate system, localAngle in D\_gridGenExampleRect, should be kept at 0.\*\***



**Additionally, in Section 8 of D\_gridGen…in the Local Rectification, you will need to change '&’ to ‘|’. See below.**



In your workspace, the output will include a GRID… .mat file. There are certain variables such as localX, localY which you will need for the next part (ii).



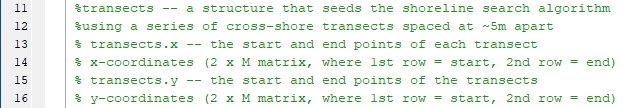
These variables are found in your workspace.

The output will also generate two rectified images in a PNG format. One will be in world coordinates and the other will be local.

1. **Transects**

*Editing transects manually may be necessary in order to capture the entirety of the shoreline. Manual editing may be time consuming, depending on your beach profile.*

The mapShoreline codes have a specific layout for the transects.



The first step is to visualize the location of the transects on the rectified image. Again, there is more than one way to do this, but a quicker way is to convert the RGB image to grayscale and plot it. To do this, the function rgb2gray will be utilized.



‘k’ is the variable representing the rectified RGB image that was obtained from the D\_gridGenRectExample code.

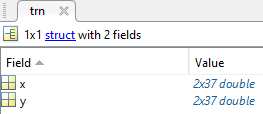
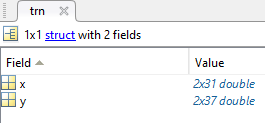
Once it has been converted to grayscale, use the imagesc function to display the image. Type “help imagesc” at the command line for more information.



localX and localY are used to specify the image location. The variable ‘k’ is the image. The format of the localX and localY in the above figure are also the xgrid and ygrid for the mapShoreline inputs.

The image may be distorted. To fix this, go into the Command Window and type help axis to determine what will best work for you. Use axis image and it will assign the dimensions to tightly fit around the data. Additionally, you may have to use axis xy to orient the image correctly.

After determining the start and ending coordinates, using imagesc, you will need to create a matrix that follows the above requirements. When creating the matrices, they must have the same number of inputs. The dimensions must be equal. See below for a good and bad example.



Variable ‘trn’ contains the transects for x and y. The start and end coordinates are spaced 5 meters apart, which is the default in the mapShoreline codes.

The trn.y variable has a larger range compared to the x coordinates. Notice the trn.x variable in this example is created using a vector of ones with the dimensions 2 x 37. 37 is the the number of elements in trn.y.

See below for an example of formatting transects.



In this example, transects are spaced at 5 m in the alongshore between 0 and 180 m in the alongshore direction, and are represented by the variable trn.y as a 2 x 37 matrix. The 2 rows in the matrix represent the starting and endpoint of the transects in the alongshore direction. In this example, the variable trn.x represents the cross-shore position of the start and endpoints of the transect at x= -300 m and -30 m in the local UTM coordinate system. The trn.x variable in this example is created using a matrix of ones with the dimensions 2 x 37m where the first and second row are multiplied by the x-value of the start and endpoints, respectively.

Type ‘hold on’ in the Command Window and plot the transects using plot (trn.x, trn.y). See below.

Chart

Description automatically generated

Transects will need adjustment in order to capture the entire shoreline. As the water level changes day to day, it is critical that the shoreline is not biased to show only a portion of the profile. This image contains transects that have not yet been edited.

To edit transects, pan in and out of the figure and determine where transects should stop or extend to.

Keep track of changes in transects and manually edit the trn.x or trn.y variables in the MATLAB workspace.

Avoid including vegetation in the transects.

# **Running the mapShoreline code**

At this point you have the x and y grid from the D\_gridGenRectExample code, the rectified image from the D\_gridGenRectExample as well and transects that may or may not have been edited.

The inputs for mapShorelineCCD or HUE should look like this:

