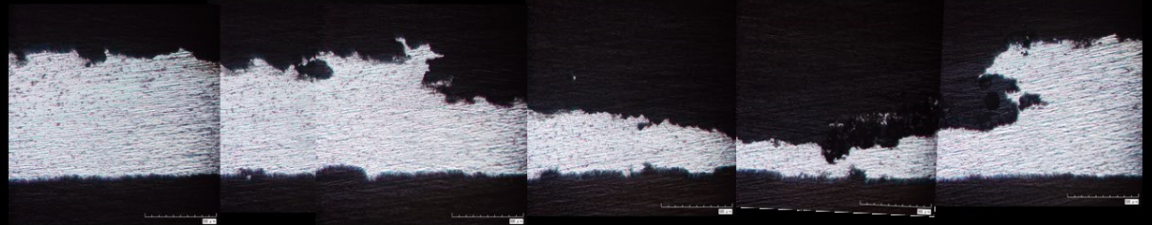
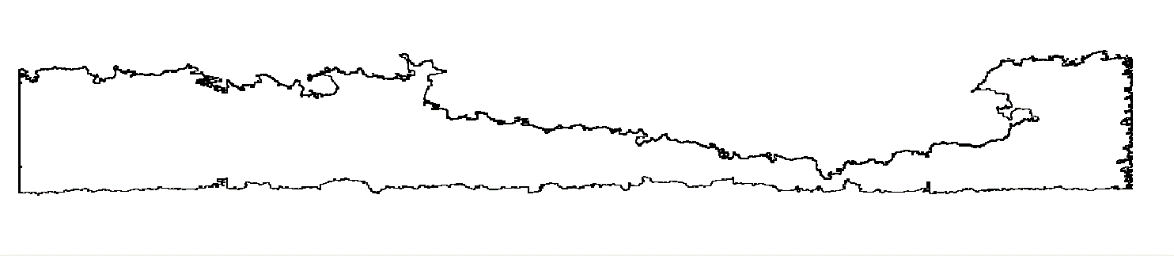
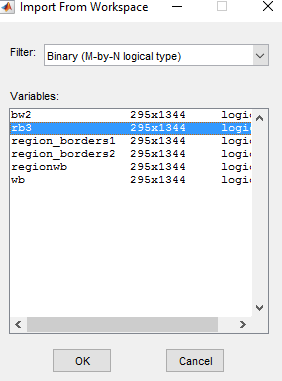
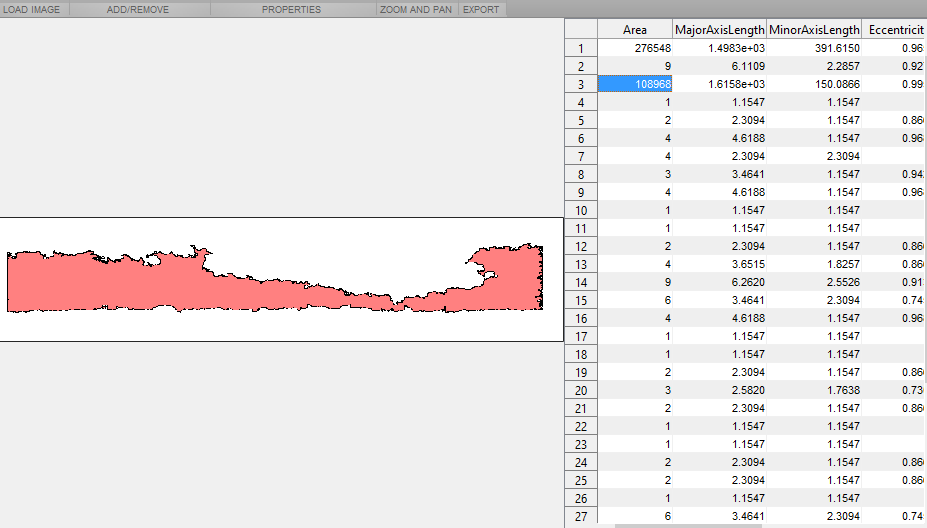
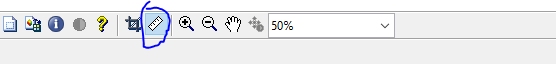
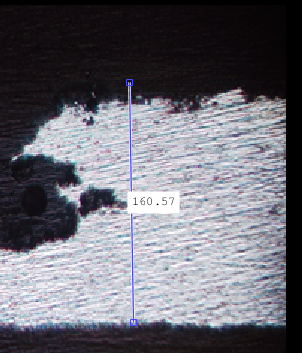
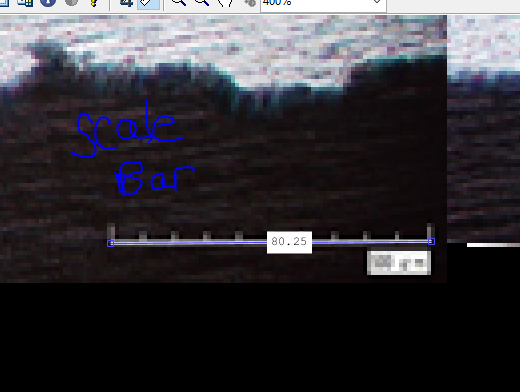
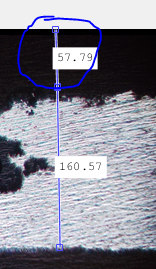
Manual for Border+Depth Analysis Code

Table of Contents

1. Objective
2. How to Use
3. Rotated Pictures
4. More on Exporting
5. Code
6. **Objective**

The goal of this code is to take a horizontal-rectangular cross sectional photo and obtain more data than other methods, primarily ImageJ. Using this code you will be able to obtain a depth profile, total un-corroded area, histogram of the depth profile, mean depth, and max depth. There is also an option to export raw data into Microsoft excel. It is important to note that you will need a **scale bar on the original photo** so you can obtain values that are not in pixels.

1. **How to Use**
2. Have MATLAB installed on computer. If you do not have MATLAB installed, UVA has a remote desktop you can use called the HIVE. Here is the link to get this installed <http://its.virginia.edu/hive/connected.html> .
3. Ensure that there is **black** on the **sides** of the photo as pictured below. This will make certain that there is a complete border. If your photo does not have black on the sides, an easy fix is to paste your photo in paint and slightly move it to the right and use the bucket with black paint and fill the background as black.
4. Open the Script **Border.m**
5. In order to use this code you must have the photo inside of the same folder as the scripts (the code). On line 1 of the code, change the name in between the ‘ ‘ to your photo’s filename.
   1. If your photo is not a .jpg change the extension in line 2 of the code to your extension.
6. Run the script by going to the editor tab at the top and clicking Run. This step is to check to see if you need to make any changes to line 7’s threshold value. The picture that the script releases labeled Figure 1 should resemble the black and white photo on the top of the next page.
   1. If the photo does not resemble this, change the number value in line 7 between the values of 0 and 1 until you receive a photo that matches you original photo and is a complete border.
7. Open the app in MATLAB called Image Region Analyzer
   1. This is found in the Apps Tab under Image Processing and Computer Vision, if this is not at the very top of your apps selection, click the black arrow at the end of the toolbar.
8. After opening Image Region Analyzer, you want to load in rb3 from the workspace.
   1. You can get to this option by clicking the down arrow on the load image button pictured to the right and select import from workspace.
   2. You will then get a screen that will give you a lot of options and you want to select rb3 and press OK. Pictured to the right.
9. This will open your photo with a table on the right. You will want to select the value in the Area Column that fills in the border with RED.
   1. Select the large numbers until you get the correct one which should look like the picture below.
   2. This value is the Total Un-corroded Area, measured in pixels which you can convert to your unit of measurement with the scale bar in the original photo.
   3. Take note of the row number, this will be the value that you plug into line 2 of the script **DepthAnalysis.m** to create the correct border.
10. Open the script **DepthAnalysis.m**  and on line 2 input the row number you figured out in step 8.
    1. You will need to put this number in the x portion of the parenthesis (x,y)
11. Open the App Image Viewer, which is in the same tab as the Image Region Analyzer in Step 6-7.
    1. Once the app is open, use the file tab and open your original photo
    2. First zoom in on your scale bar so that it takes up the whole screen and use the ruler tool pictured below to measure how many pixels the scale bar is
    3. Take note of this value as you will need to use this later for line 14
    4. At this point, if you have a part of your sample that is un-corroded in the photo move on to step 11, if you do not continue to step e.
    5. This will take prior knowledge of how thick your sample was and converting that into pixels using the value you measured in step 10 b.
    6. Once you know this value, use the ruler tool and measure out this amount of pixels starting at the bottom of the sample.
    7. Now from that point measure to the top of the entire photo. THIS is the value you will type into line 8. Below is a series of photos showing this assuming the sample was 1000 µm and the scale bar was for 500 µm.
    8. Be sure to switch the % from line 8 to line 7 as depicted below.

** TO**

1. In line 14, change the ration to fit your scale bar ratios.
   1. The default currently is 500 µm to 80 pixels
   2. Feel free to change Micro\_per\_pixel to a different name but be sure to that when you start to rename it to **SHIFT+ENTER** when done to switch all places where this function is used in the code switches also.
2. To change how the Histogram is labeled in line 37 you can change the title by changing the words from INPUT TITLE HERE to the title you wish.
   1. The same goes for lines 38 and 39 with changing the y and x axis labels.
   2. For greek letters, the correct way to type them is {\GREEK LETTER NAME}
   3. In line 40 you can change the name of what the Histogram will be saved as in the folder.
3. This is **IF** you want to export the data into excel.
   1. On line 43, in purple there is written Input Name, this is where you type in the name of the file you wish to create. The program will create a new excel with this name in this folder.
   2. The following two lines 44 and 45 are where it writes the raw data from the first graph (Depth Profile)
4. Line 47 writes a .csv that has columns seperated by , and rows by ,,
   1. Type in the name you want where it says Input Name
   2. This is an alternative to data storage from excel, if you wish to not have excel simply delete the lines that deal with excel
5. Click Run in the Editor Tab
   1. You should get 2 figures
      1. Depth Profile
      2. Histogram
   2. In the command window you will receive MaxCorr and MeanCorr which correlate to max depth of the corrosion and the Mean corrision depth both in the units you specified with the conversion from **Step 11**.
6. **Rotated Pictures**
7. If the image taken is not level, you may wish to rotate it to receive more accurate data. The commented out code from lines 22-37 can be utilized to solve this. This portion of the code estimates a slope for the edge of the sample, finds the angle between that slope and the x-axis, and then rotates the plot of corrosion damage so that it is flush with the x-axis.
   1. To generate the slope, a linear portion of the corrosion damage plot is selected by observation. The variables ‘lowerbound’ and ‘upperbound’, in lines 22 and 23, respectively, set endpoints in terms of indices of the ‘DifferenceY’ matrix. Make sure that these are in terms of indices (not microns or pixels) and that they represent a flat part of the edge.
   2. Using matrix math, the angle between the edge and the x-axis is found in line 29. If the resulting figure that is displayed does not accurately rotate the image to flat, either the lower and upper bounds may be changed, **OR** you may simply multiply the value of ‘theta’ in line 29 by a scalar, which is often easier.
   3. Additionally, if you have noticed that the plot was rotated clockwise instead of counterclockwise (or vice versa), you may change the sign of theta in line 29 by simply adding or removing a negative sign. This stems from the fact that the “rotation matrix” used in line 31 is used for rotating points counterclockwise, when indeed a clockwise rotation is sometimes necessary.
8. **More on Exporting**
9. For excel if you wish to continue using same file and have the data all on the same workbook you have two options, change the sheet it is exported onto for each photo or different columns on the same sheet.
   1. To change the sheet number, simply change the number after ‘Sheet1’ to ‘Sheet2’.
   2. To keep the data on the same sheet but different columns change ‘A1’ (for x data) and ‘B1’ (for y data) to either ‘C1’ and ‘D1’ respectively or if you want a space in between ‘D1’ and ‘E1’.
10. **Code**

***Border.m***

location='BareB7';%file path minus type extension goes here

pic=strcat(location,'.png');%change this extension if the file is not a jpg

img = imread(pic);%this reads a file ( you must type in the file name and be sure that it is in the same folder as this script)

I = rgb2gray(img); %changes image from rgb color scheme to gray

[x,map] = gray2ind(I,100); %take the image and maps the gray scale to 100 different shades

bw2=im2bw(x,map,0.45); % takes the gray scale and makes it black and white

wb = imcomplement(bw2); %changes the image to white and black (inverse the black and white)

regionwb=bwareaopen(wb,300); %this takes any area that is 350 pixels or less and fills it with black you can see this from figure 1 to 2

region\_borders1 = imdilate(regionwb,ones(3,3)) > imerode(regionwb,ones(2,3)); %this will make the border of the image

region\_borders2 = bwareaopen(region\_borders1,300);%Found this way to be a better way of producing the border Included previous version to show you the difference

rb3=imcomplement(region\_borders2);

figure1=figure;

imshow(rb3,'InitialMagnification', 'fit')

***DepthAnalysis.m***

boundaries = bwboundaries(rb3,'noholes');

X=boundaries(3,1); %Change first number in order to obtain border in question

Y=X{1,1};

I = find(Y(:,2)==max(Y(:,2)));

Yx=min(I); %finds where the border turns to an edge

plot(Y(1:Yx,2),Y(1:Yx,1))

maxy = min((Y(1:Yx,1)));

%maxy=57.59; %This is where you type in the value you found from measuring in Image viewer

Z = (Y(1:Yx,1))-maxy;

hold on

line('Xdata', [0 Yx],'Ydata', [maxy maxy])

hold off

Micro\_per\_pixel = 500/80;%second number is for scale bar pixels

DifferenceY = Z\*Micro\_per\_pixel;

figure2=figure;

Xunit= (Y(1:Yx,2))\*Micro\_per\_pixel;

DifferenceX=min(Xunit);

Xunits=(Y(1:Yx,2))\*Micro\_per\_pixel-DifferenceX;

plot(Xunits,DifferenceY)

MaxCoated=max(Xunits);

% lowerbound=2000; %sets lower and upper bounds of the x scale to linearize

% upperbound=6000;

% unos=zeros(length(Xunits(lowerbound:upperbound)),1);

% unos(:,1)=1;

% X=[unos Xunits(lowerbound:upperbound)];

% B=X\DifferenceY(lowerbound:upperbound); %generates solution matrix that carries the slope of the edge of the sample

% theta=-atan(B(2)); %theta is the angle of rotation...may be hard coded...USE THIS COMMENTED OUT SECTION IF IMAGE NEEDS ROTATING

% rotate=[cos(theta) -sin(theta);sin(theta) cos(theta)];

% flat=[Xunits DifferenceY]\*rotate; %rotates DifferenceY matrix...should be flat

% maxy2=min(flat(:,2));

% figure

% plot(flat(:,1),flat(:,2)-maxy2)

% corrLength=flat(:,2)-maxy2;

name=strcat(location,'\_histogram');

figure3=figure;

histogram(DifferenceY) %use 'DifferenceY' matrix if pic does not need rotating, use 'CorrLength' matrix' if it does

title('INPUT TITLE HERE') %Title for Histogram chart

xlabel('Damage Depth ({\mu}m)')

ylabel('Frequency')

saveas(figure3,'name\_histogram.jpg')

MaxCorr=max(DifferenceY)%max corrosion damage...use 'DifferenceY' matrix if pic does not need rotating, use 'CorrLength' matrix' if it does

MeanCorr=mean(DifferenceY)%mean corrosion damage...use 'DifferenceY' matrix if pic does not need rotating, use 'CorrLength' matrix' if it does

filename='Input Name.xlsx'; %change name here for excel sheet

xlswrite(filename,Xunits,'Sheet2','A1');

xlswrite(filename,DifferenceY,'Sheet2','B1');

M=[Xunits,DifferenceY];

csvwrite('Input Name',M,0,2);