Ring Pull Coating Analysis (RPCA) Tutorial

Introduction

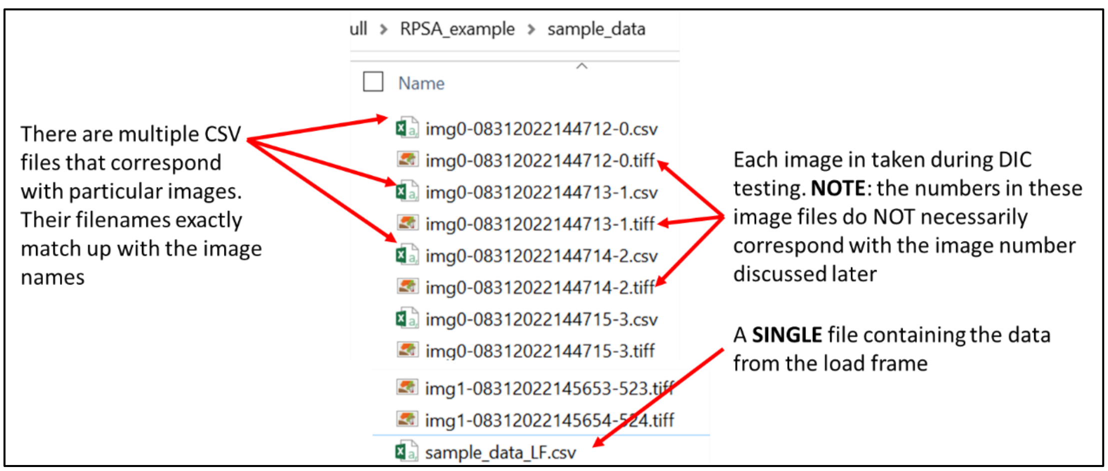
This tutorial is intended for a novice user to be able to run the Ring Pull Coating Analysis module. This requires the Ring Pull Strain Analysis script as well.

Getting Started

Please download Python from anaconda.org. There are multiple ways to download Python, however, this one contains all the necessary modules. Features of Anaconda will be displayed heavily in this tutorial.

File Structure

The files that are needed to run this program are shown below



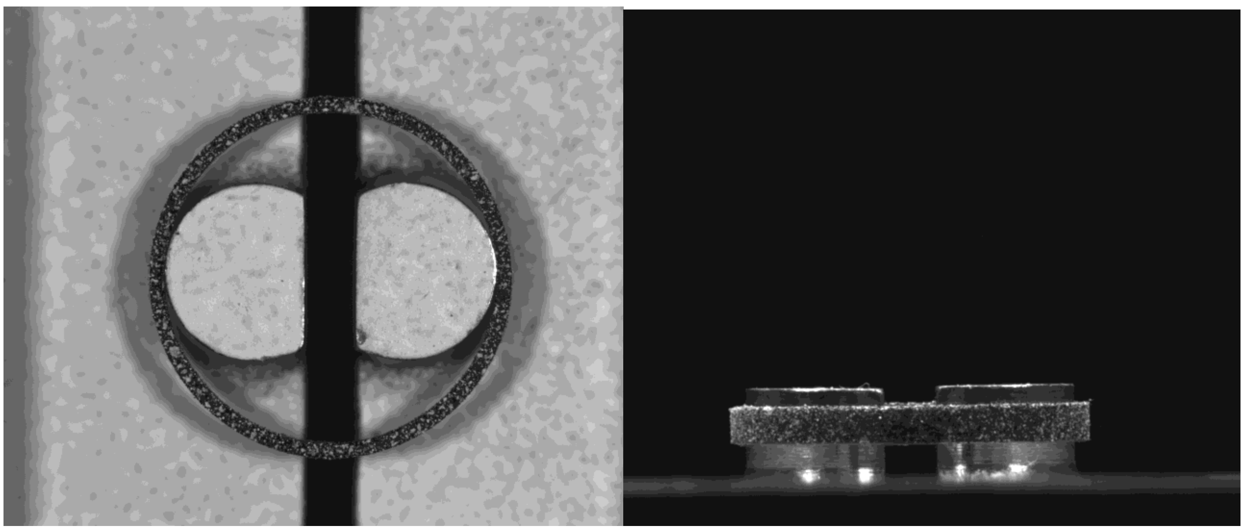
Below are some images showing the type of data present in each file. Note that the .csv files must have specific headers.

Load Frame File:

Each data point has to have image files associated with it. Specify the exact names of the image here. The default/first images should go in the top_img_file column. The side_img_file column is for coating analysis. The filenames must have the extension listed as well. You may either specify the FULL filepath OR you may have all the images in the same folder as this file.

	A	B	C	D	E
1	top_img_file	side_img_file	time (seco	displacem	load (N)
2	img0-08312022144721-9.tiff	img1-08312022144721-7.tiff	0	0	1.52449
3	img0-08312022144722-10.tiff	img1-08312022144722-8.tiff	1	0.005853	3.217074
4	img0-08312022144723-11.tiff	img1-08312022144723-9.tiff	2	0.015224	3.510204
5	img0-08312022144724-12.tiff	img1-08312022144724-10.tiff	3	0.02253	3.897891
6	img0-08312022144725-13.tiff	img1-08312022144725-11.tiff	4	0.030023	4.361225
7	img0-08312022144726-14.tiff	img1-08312022144726-12.tiff	5	0.037377	4.739456
8	img0-08312022144727-15.tiff	img1-08312022144727-13.tiff	6	0.04475	5.20279

Image Files:



csv files:

- These files are most often generated from a third party digital image correlation software. Supported file modes are VIC-2D and DICengine.

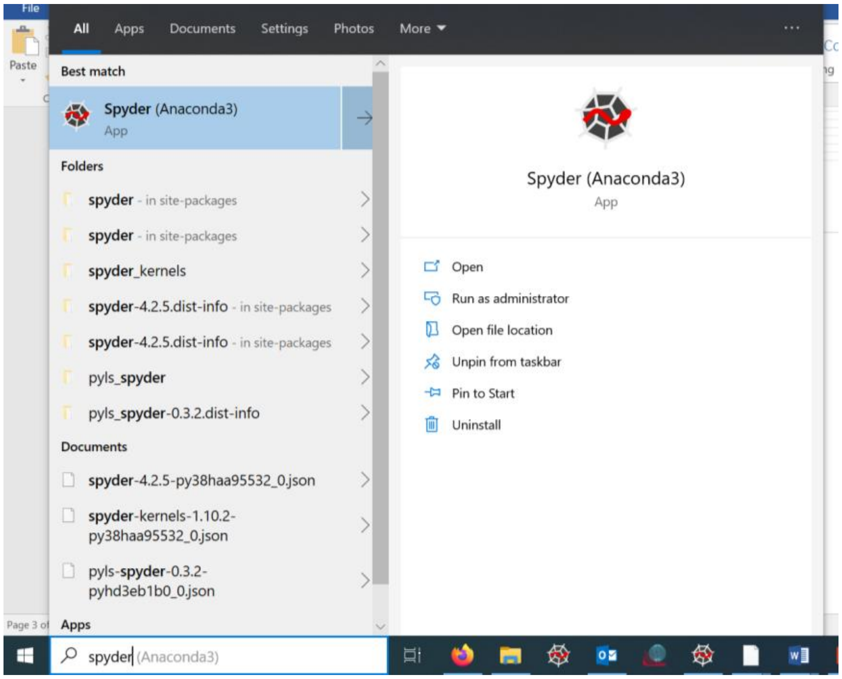
Necessary columns

A1 "x"

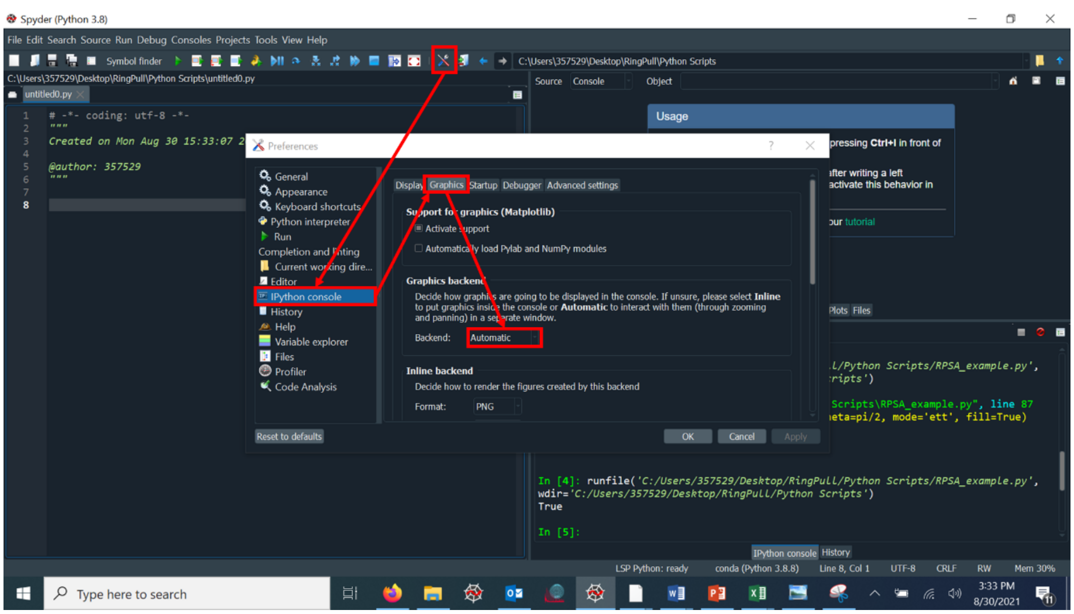
	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y
1	"x"	"y"	"x_c"	"y_c"	"u"	"v"	"u_c"	"v_c"	"exx"	"eyy"	"exy"	"fxx"	"fxy"	"fyx"	"fyy"	"e1"	"e2"	"gamma"	"e_tresca	"e_vonmi	"sigma"	"Xp"	"Yp"	"Zp"	
2	51	322	-1173	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
3	55	322	-1169	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
4	59	322	-1165	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
5	63	322	-1161	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
6	67	322	-1157	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
7	71	322	-1153	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
8	75	322	-1149	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	
9	79	322	-1145	702	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	-1	0	0	0	

Opening and running the script

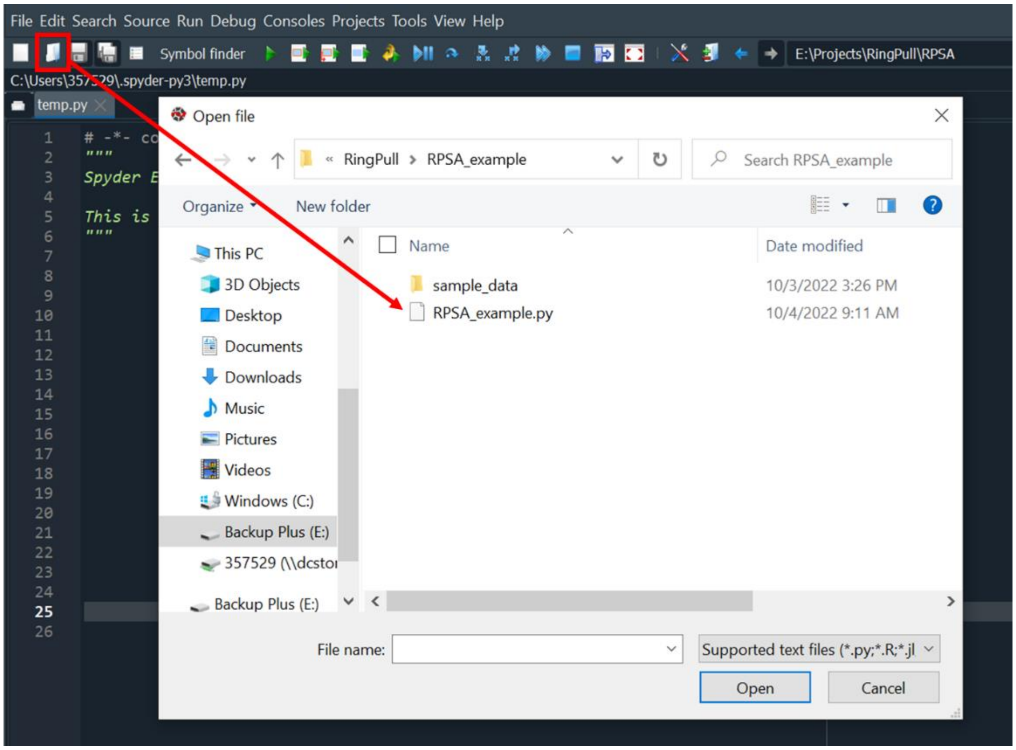
Start by opening Spyder:



It is required to change the graphics backend to display the plots in a separate window. The backend cannot be inline, so any other setting should work. This will allow Spyder to create additional windows for any plots or figures that are made.



Now the RPSA example script can be opened.



Now, walking through the script. Here are some imported packages. If you downloaded Anaconda, these should all be available with minimal hassle. We also define pi as we will often use it. Note the red box – this is where we import the module RingPullStrainAnalysis.py from a different folder.

```
## import the module from a different folder
import sys
sys.path.insert(0, 'E:\\Projects\\RingPull\\RPSA')
from RingPullStrainAnalysis import *

import numpy as np
pi = np.pi
import matplotlib.pyplot as plt

```

Here are the variables you can change in the code. The most important are the directory and the filename for the load frame data. Here is also where you specify geometric dimensions of the ring.

This variable must have the full filepath

Requires either double backslashes (shown) or single frontslash

```
## set all the variables you will need to run this RingPullStrainAnalysis code:

## the file with the load frame data and their corresponding images
LF_file = 'E:\\Projects\\RingPull\\RPSA_example\\sample_data\\sample_data_LF.csv'
#The DIC analysis software that was used
DIC_software = 'VIC-2D'
## geometric dimensions of the ring pull test
d_mandrel = 3.0
OD = 10.3
ID = 9.46
W  = 1
```

All filenames must have their extensions on them

This line of code creates an instance of the RingPull class which is intended to encompass an entire test. It loads in the load frame data and stores it in memory.

```
## create RingPull object
test = RingPull(LF_file=LF_file,
                DIC_software=DIC_software,
                ID=ID, OD=OD, d_mandrel=d_mandrel, W=W,
                get_geometry_flag=True)
```

Now we run the coating analysis. First you have to call the RPCA class and save it in a variable, named ‘method’ in this script. One important variable to set is the mode. Either set it to ‘compression’ or ‘tension’ depending on whether you are looking at the compression or tension surface.

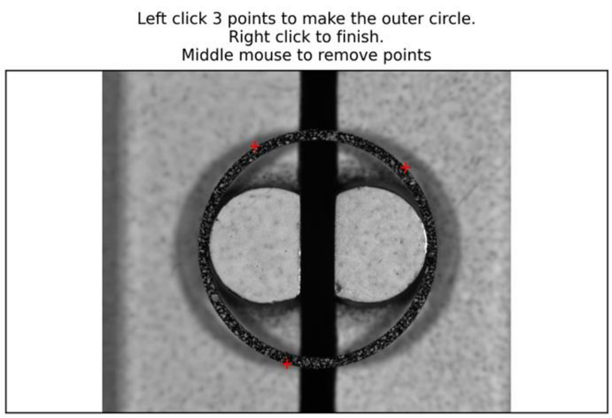
Next, choose the image numbers to analyze. In this case, images 60, 65, and 70 are analyzed.

Finally, run the primary method in RPCA, get_side_image_strain.

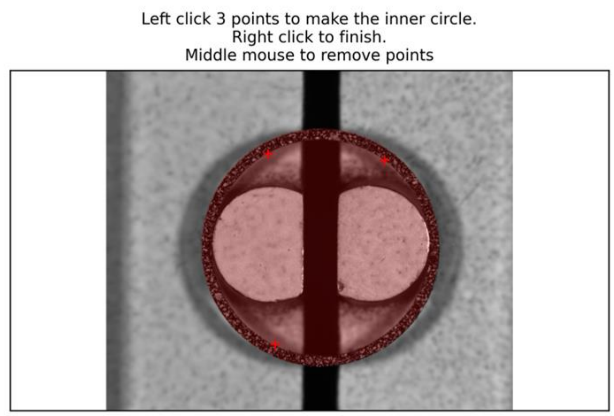
```
## Create coating analysis instance and run it on a few different frames
method = RingPullCoatingAnalysis(test,mode='compression')
n=[60,65,70]
analysis_data = method.get_side_image_strain(n,debug=False)
```

Now run the script. A series of prompts will pop up as figures. Follow the instructions with your figures. An example is reproduced below.

Start by defining the area of the ring. First, select three points to define the (a) outer surface, then three more points for the (b) inner surface.

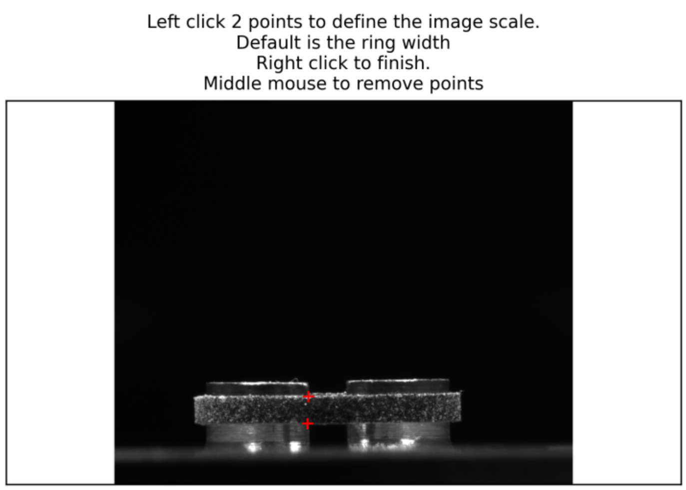


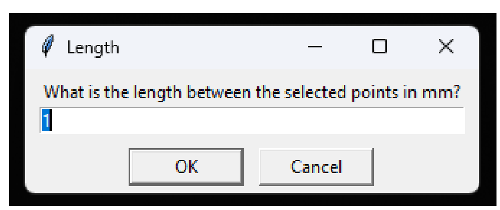
a)



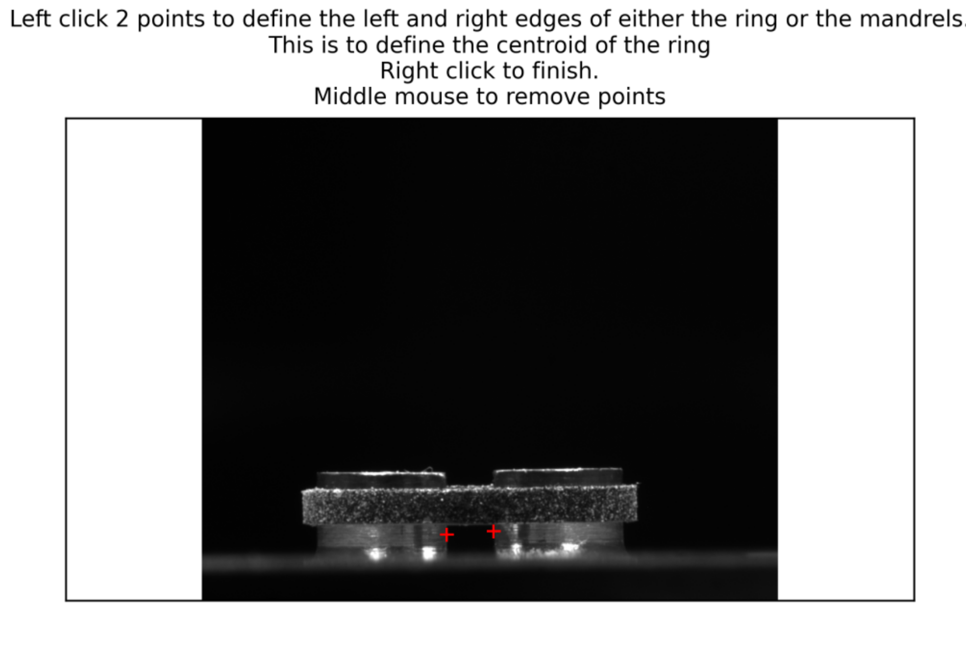
b)

Next, we are defining the image scale. Choose two points that you know the physical measurement for. Note that since this is a 2D image that may be imaging a 3D landscape, it is suggested to make this measurement in the same plane as the area of interest in the photo. Then a pop up text box will ask you for the measurement in mm.



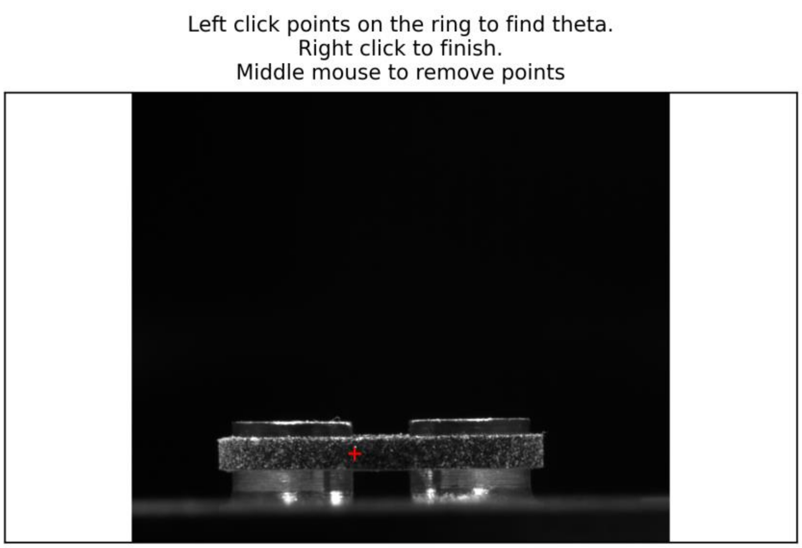


Next, we need to find the symmetric axis of the ring in the image. Select two points on known surfaces that are equidistant from the symmetry axis. The y-coordinate (vertical coordinate) of these points does not matter.



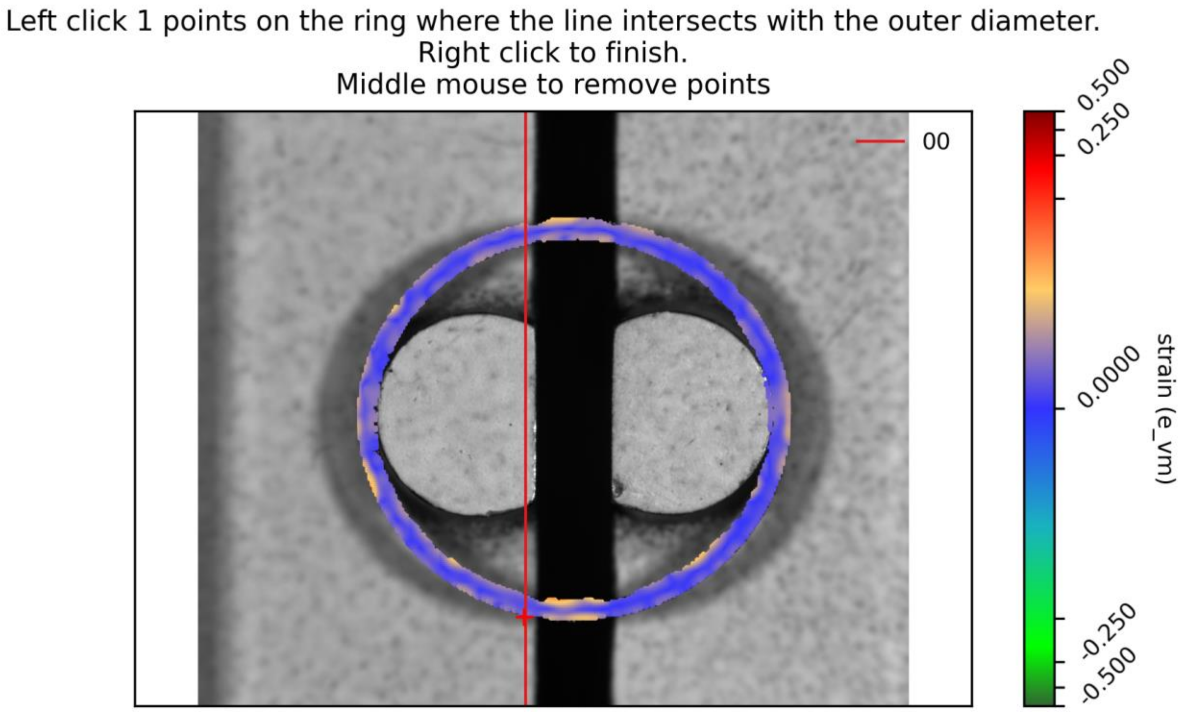
Next, select points on the ring where you want to get the strain. Multiple points are okay.

If multiple images are used, these will cycle in order.



Now images of the ring will show with either vertical or horizontal lines – these lines are projections of the points you clicked, choose the location where the line intersects with the outer surface of the ring that you took the image of. Note that you have to know where the cameras were located geometrically before doing this step. Also, it is okay to get close – the script will snap to the nearest location on the outer surface of the ring.

If you selected multiple points in the previous section, then multiple lines will show up in the image. Select one point per line IN ORDER.



Now the script will run and output the data as a DataFrame to whatever variable you saved it as. In this case, it is saved to the variable, analysis_data. It is recommended that you save this dataframe as a csv file (analysis_data.to_csv(filename)).

```
analysis_data = method.get_side_image_strain(n,debug=False)
```