May4

May 4, 2020

1 Review

- No review

2 What I need to do this week

- Completed
  - 40 losses for fmincon
  - Compare the loss of fmincon and bads
  - Generate the tables and the plots for the manuscript
  - plot the landscape of the loss function
- Unfinished
  - SOC and ori-surround
  - Parameter recovery

3 40 losses for fmincon

3.1 Check figures

Have a look at the figures.

Criterion for the same loss:

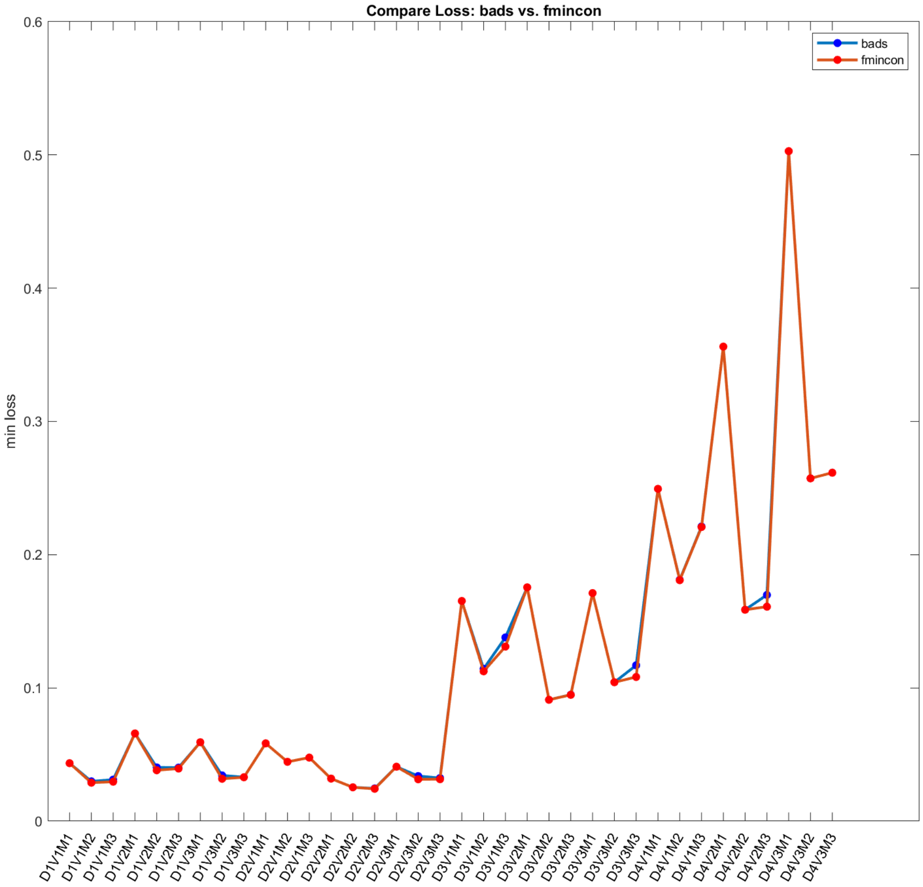
$$\text{loss}_i = \min(\text{losses}),\ \text{if } (\text{loss}_i - \min(\text{losses}) < 1e - 5$$

3.2 Conclusion:

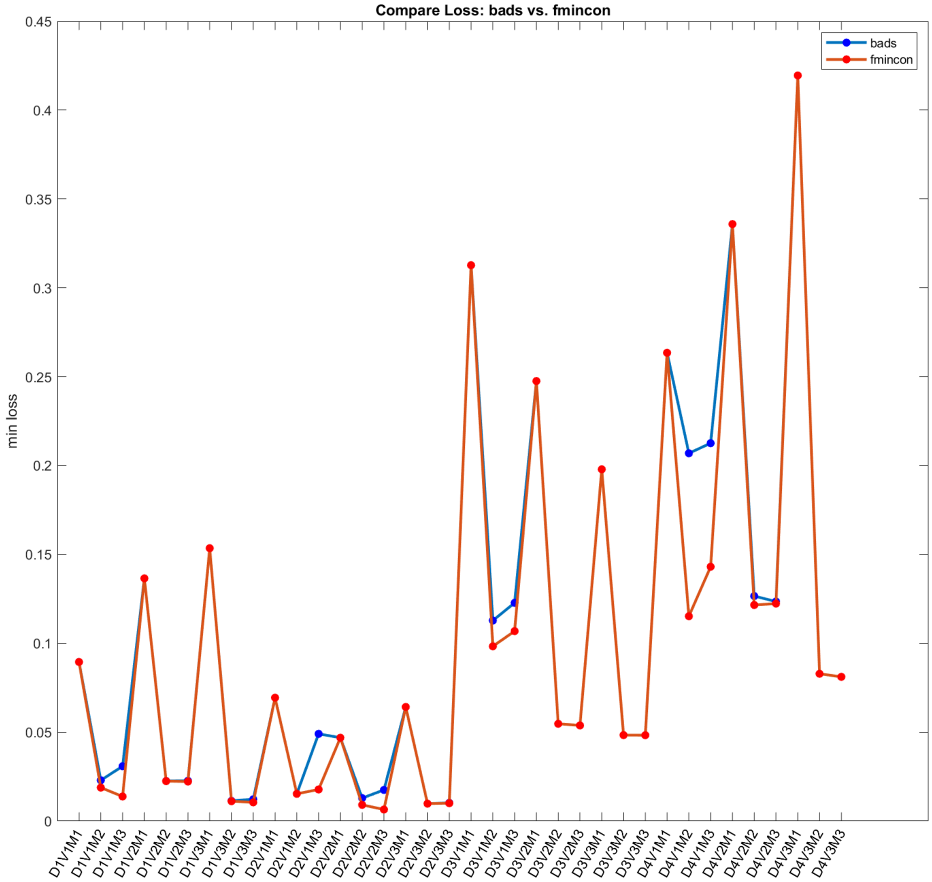
Fmincon is able to find the global minima reliably.

4 Compare the loss of fmincon and bads

4.1 Compare loss for “ALL” stimuli set.



4.2 Compare loss for “target” stimuli set.



4.3 Conclusion:

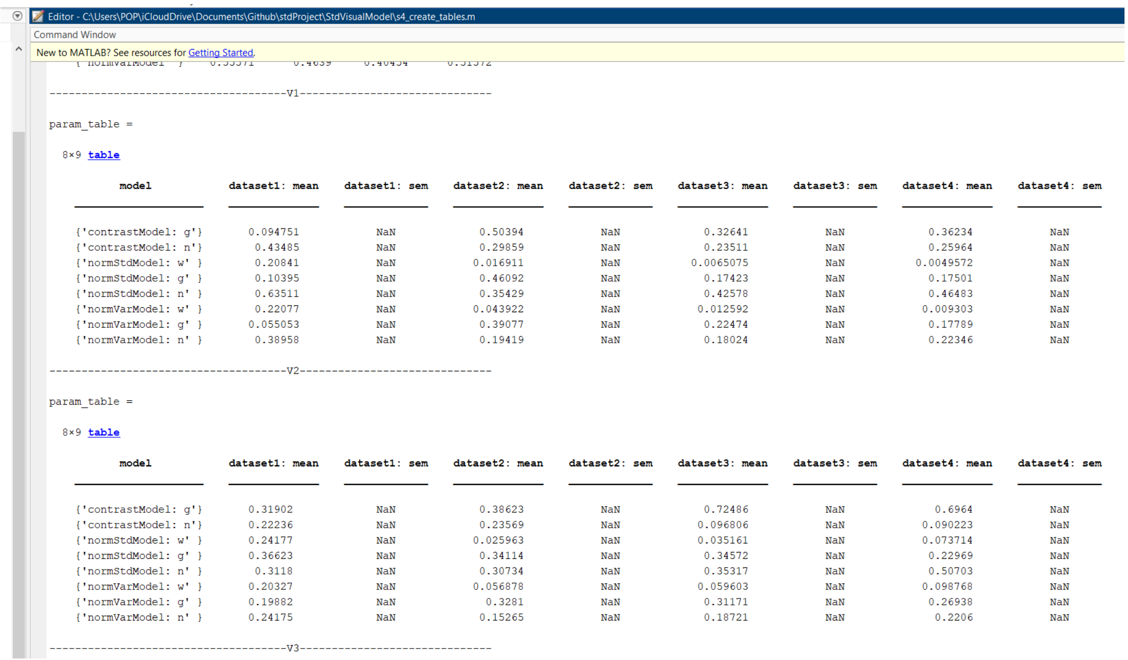
Fmincon can always find equal or even lower minima than those found by BADS.

5 Generate the tables and the plots for the manuscript

- s4_create_tables
- s5_create_plots

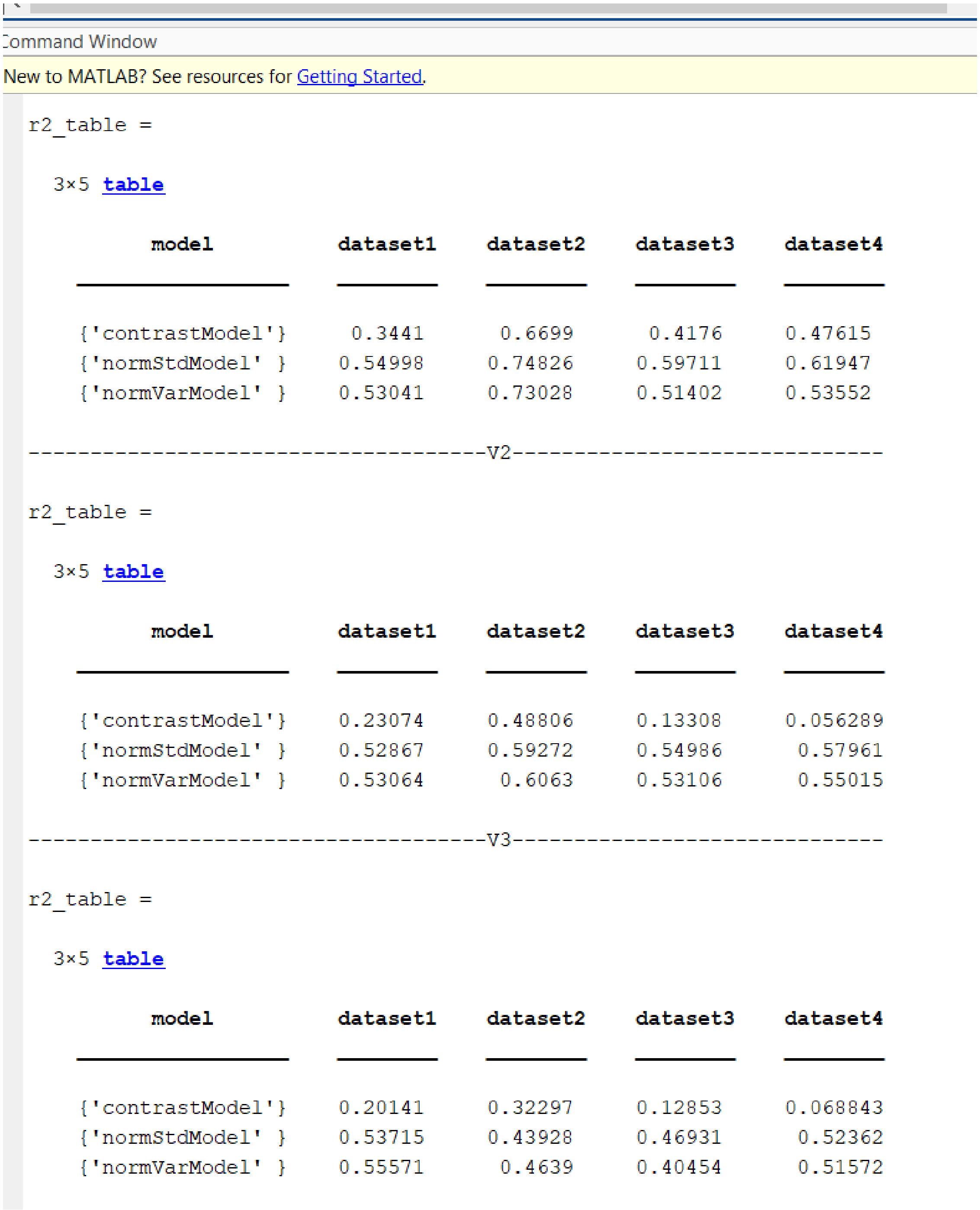
5.1 Tables

5.1.1 Param tables


Editor - C:\Users\POP\iCloudDrive\Documents\Github\stdProject\StdVisualModel\s4_create_tables.m
Command Window
New to MATLAB? See resources for Getting Started.

```
-----------------------------------V1-----------------------------

param_table =

  8×9 table

          model             dataset1: mean    dataset1: sem    dataset2: mean    dataset2: sem    dataset3: mean    dataset3: sem    dataset4: mean    dataset4: sem
    ___________________     ______________    _____________    ______________    _____________    ______________    _____________    ______________    _____________

    {'contrastModel: g'}       0.094751            NaN              0.50394            NaN              0.32641            NaN              0.36234            NaN
    {'contrastModel: n'}        0.43485            NaN              0.29859            NaN              0.23511            NaN              0.25964            NaN
    {'normStdModel: w' }        0.20841            NaN             0.016911            NaN            0.0065075            NaN            0.0049572            NaN
    {'normStdModel: g' }        0.10395            NaN              0.46092            NaN              0.17423            NaN              0.17501            NaN
    {'normStdModel: n' }        0.63511            NaN              0.35429            NaN              0.42578            NaN              0.46483            NaN
    {'normVarModel: w' }        0.22077            NaN             0.043922            NaN             0.012592            NaN             0.009303            NaN
    {'normVarModel: g' }       0.055053            NaN              0.39077            NaN              0.22474            NaN              0.17789            NaN
    {'normVarModel: n' }        0.38958            NaN              0.19419            NaN              0.18024            NaN              0.22346            NaN

-----------------------------------V2-----------------------------

param_table =

  8×9 table

          model             dataset1: mean    dataset1: sem    dataset2: mean    dataset2: sem    dataset3: mean    dataset3: sem    dataset4: mean    dataset4: sem
    ___________________     ______________    _____________    ______________    _____________    ______________    _____________    ______________    _____________

    {'contrastModel: g'}       0.31902             NaN              0.38623            NaN              0.72486            NaN               0.6964            NaN
    {'contrastModel: n'}       0.22236             NaN              0.23569            NaN             0.096806            NaN             0.090223            NaN
    {'normStdModel: w' }       0.24177             NaN             0.025963            NaN             0.035161            NaN             0.073714            NaN
    {'normStdModel: g' }       0.36623             NaN              0.34114            NaN              0.34572            NaN              0.22969            NaN
    {'normStdModel: n' }        0.3118             NaN              0.30734            NaN              0.35317            NaN              0.50703            NaN
    {'normVarModel: w' }       0.20327             NaN             0.056878            NaN             0.059603            NaN             0.098768            NaN
    {'normVarModel: g' }       0.19882             NaN               0.3281            NaN              0.31171            NaN              0.26938            NaN
    {'normVarModel: n' }       0.24175             NaN              0.15265            NaN              0.18721            NaN               0.2206            NaN

-----------------------------------V3-----------------------------
```

5.1.2 R2 tables



New to MATLAB? See resources for Getting Started.

```
r2_table =

  3×5 table

         model          dataset1    dataset2    dataset3    dataset4
    _________________   ________    ________    ________    ________

    {'contrastModel'}    0.3441      0.6699      0.4176     0.47615
    {'normStdModel' }   0.54998     0.74826     0.59711     0.61947
    {'normVarModel' }   0.53041     0.73028     0.51402     0.53552

-------------------------------------V2-----------------------------

r2_table =

  3×5 table

         model          dataset1    dataset2    dataset3    dataset4
    _________________   ________    ________    ________    ________

    {'contrastModel'}   0.23074     0.48806     0.13308     0.056289
    {'normStdModel' }   0.52867     0.59272     0.54986      0.57961
    {'normVarModel' }   0.53064      0.6063     0.53106      0.55015

-------------------------------------V3-----------------------------

r2_table =

  3×5 table

         model          dataset1    dataset2    dataset3    dataset4
    _________________   ________    ________    ________    ________

    {'contrastModel'}   0.20141     0.32297     0.12853     0.068843
    {'normStdModel' }   0.53715     0.43928     0.46931      0.52362
    {'normVarModel' }   0.55571      0.4639     0.40454      0.51572
```

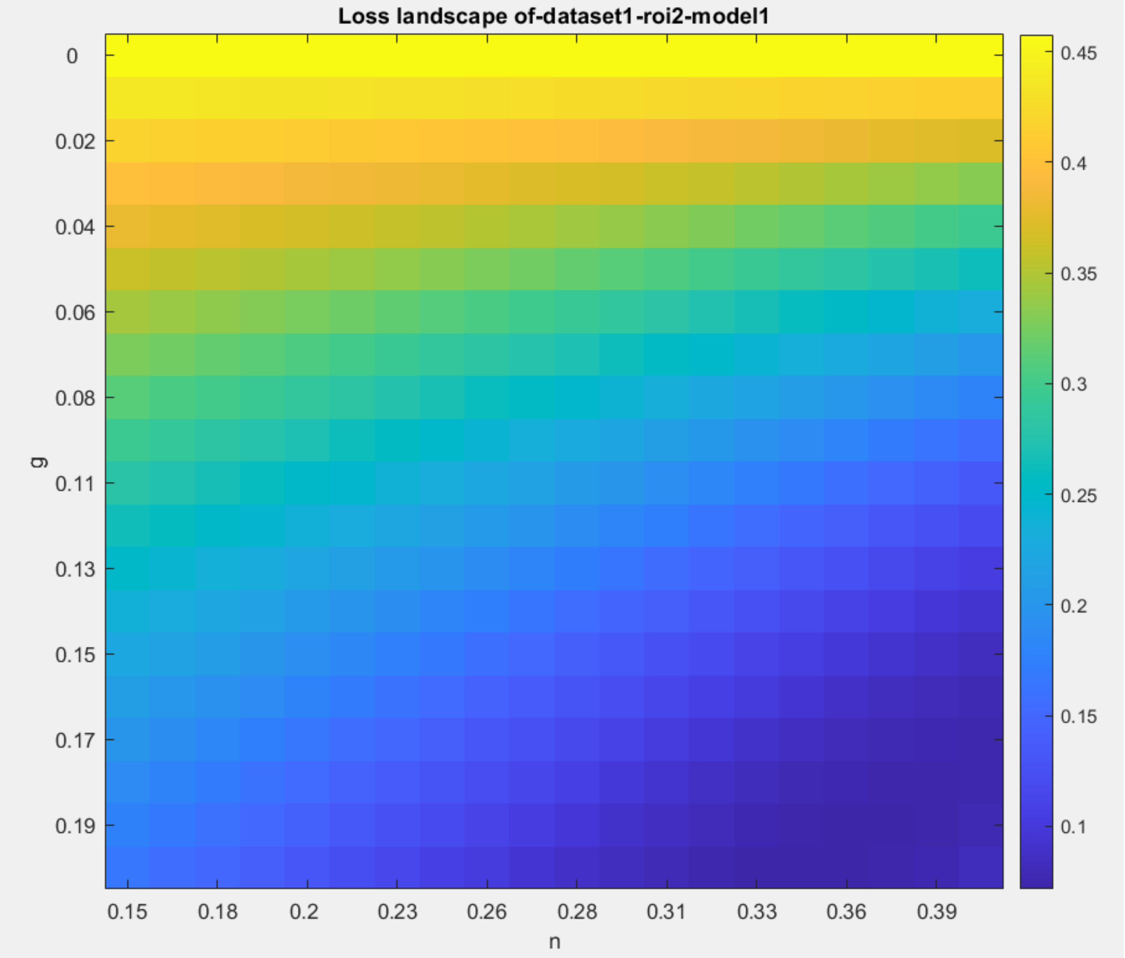
5.2 Plots

The same with what we have shown in the past few weeks

5.3 Issues

1. Adding cross validation. Due to good performance of fmincon, including the speed and accuracy, I suggest we can try to do the cross-validation again.
2. Share data. Github does not allow uploading the data. I remeber Jon mentioned a place to upload data.
3. About the manuscript. I cannot find the latest version of the manuscript in the google drive.

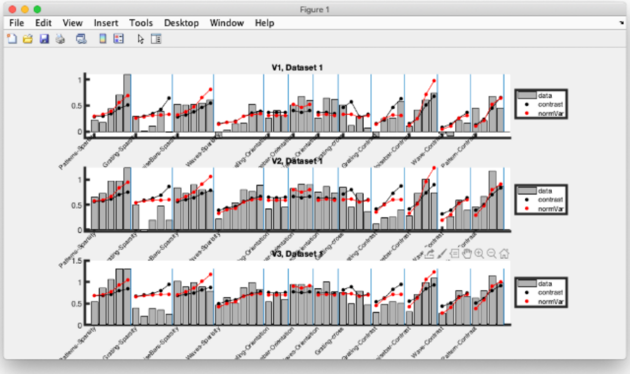
6 Loss landscape



7 SOC and ori-surround

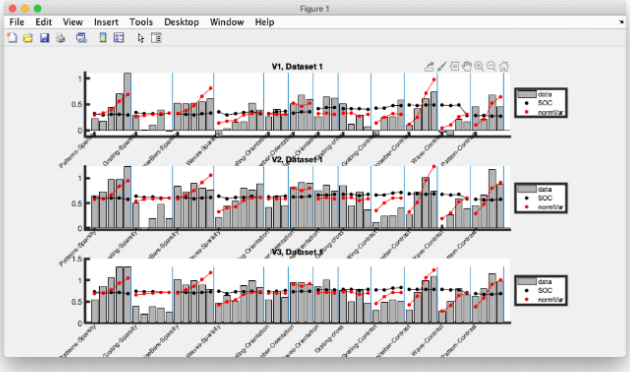
7.1 BIG PROBLEM!!!

SEP 2019:



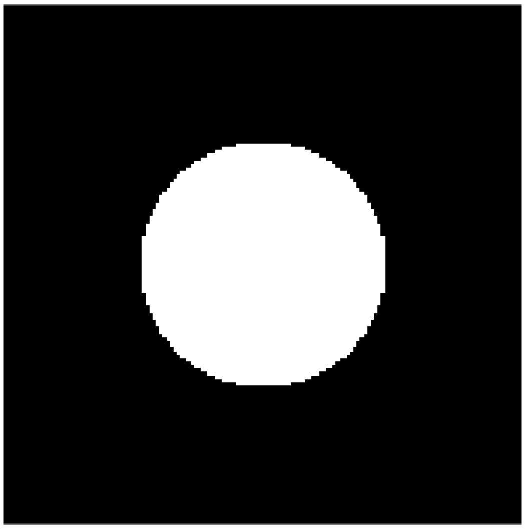
HOWEVER, the oriented surround model did not come out well at all. I am guessing there is some kind of bug. Possibly I introduced it in changing the code.

As you can see, nearly all the predictions come out the same for this model, which does not make sense. Maybe we can debug this together on Thursday.



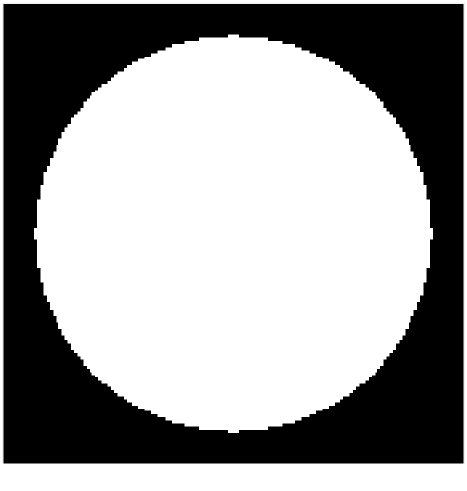
The problem comes from the following piece of code.

```
function [ w ] = gen_disk( size_e)

        % Create a meshgrid to
        [ X , Y ] = meshgrid( linspace( -1 , 1, size_e));

        % Create a disk with certain size
        w = zeros( size_e ,  size_e);
        panel = X.^2 + Y.^2;

        % Choose the radius of the disk ,  3 std of the edge size
        theresold = ( size_e  - 116)/size_e;

        % Any pixels
        [index] = find(panel < theresold);
        w(index) = 1;

end
```

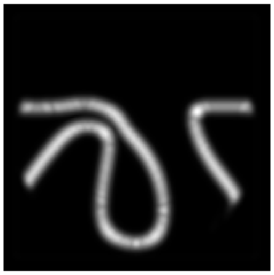


7.2 New code

```
function [ w ] = gen_disk( size_e)

        % Create a meshgrid to
        [ X , Y ] = meshgrid( linspace( -1 , 1, size_e));

        % Create a disk with certain size
        w = zeros( size_e ,  size_e);
        panel = X.^2 + Y.^2;

        % Choose the radius of the disk ,  3 std of the edge size
        theresold = .75;

        % Any pixels
        [index] = find(panel < theresold);
        w(index) = 1;

end
```



E_xy



[ ]:

```
```