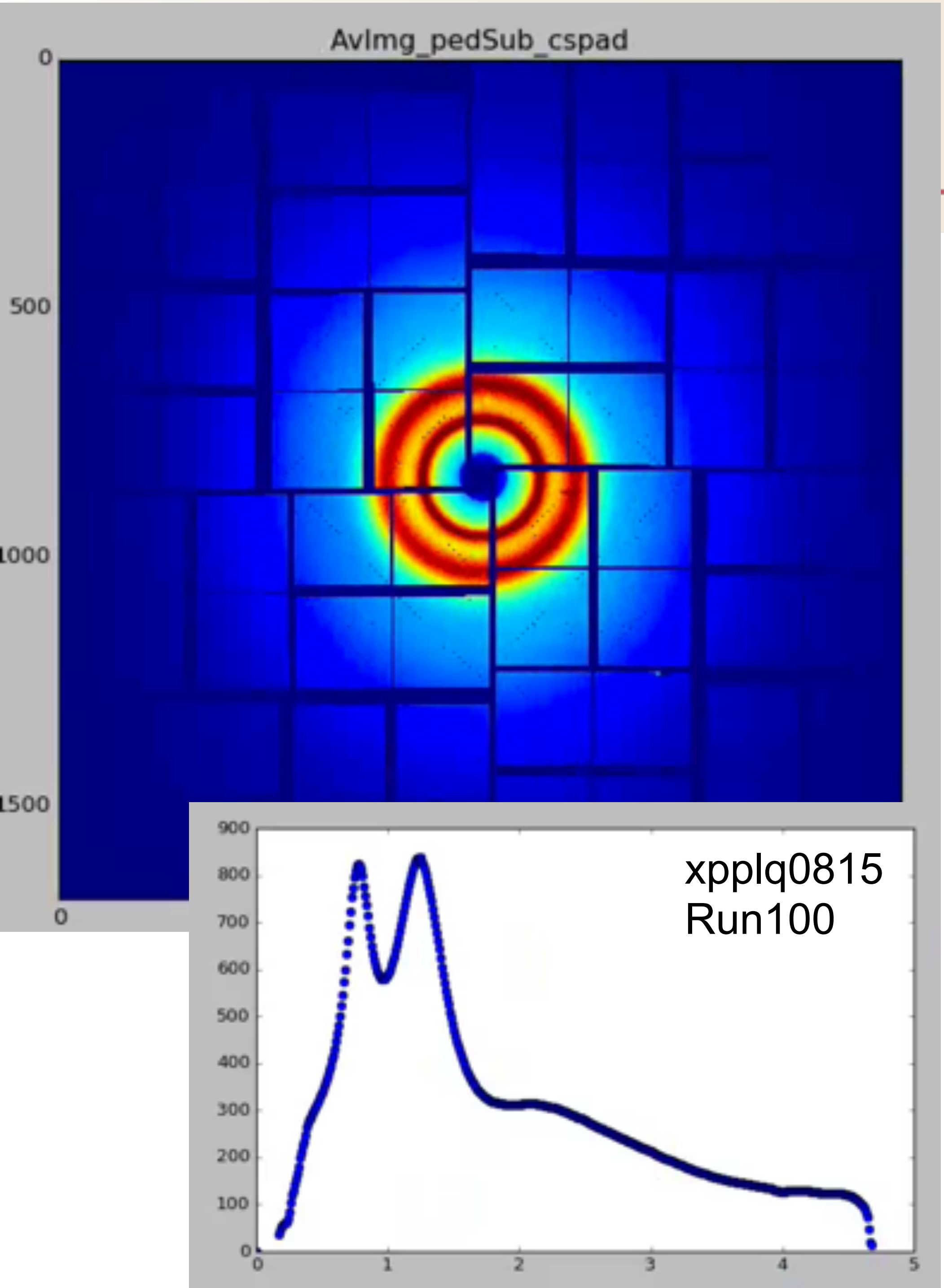
Beam Center Finding

Silke Nelson

U.S. DEPARTMENT OF ENERGY
Office of Science

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction



xpplq0815
Run100

beam center is necessary for azimuthal averaging

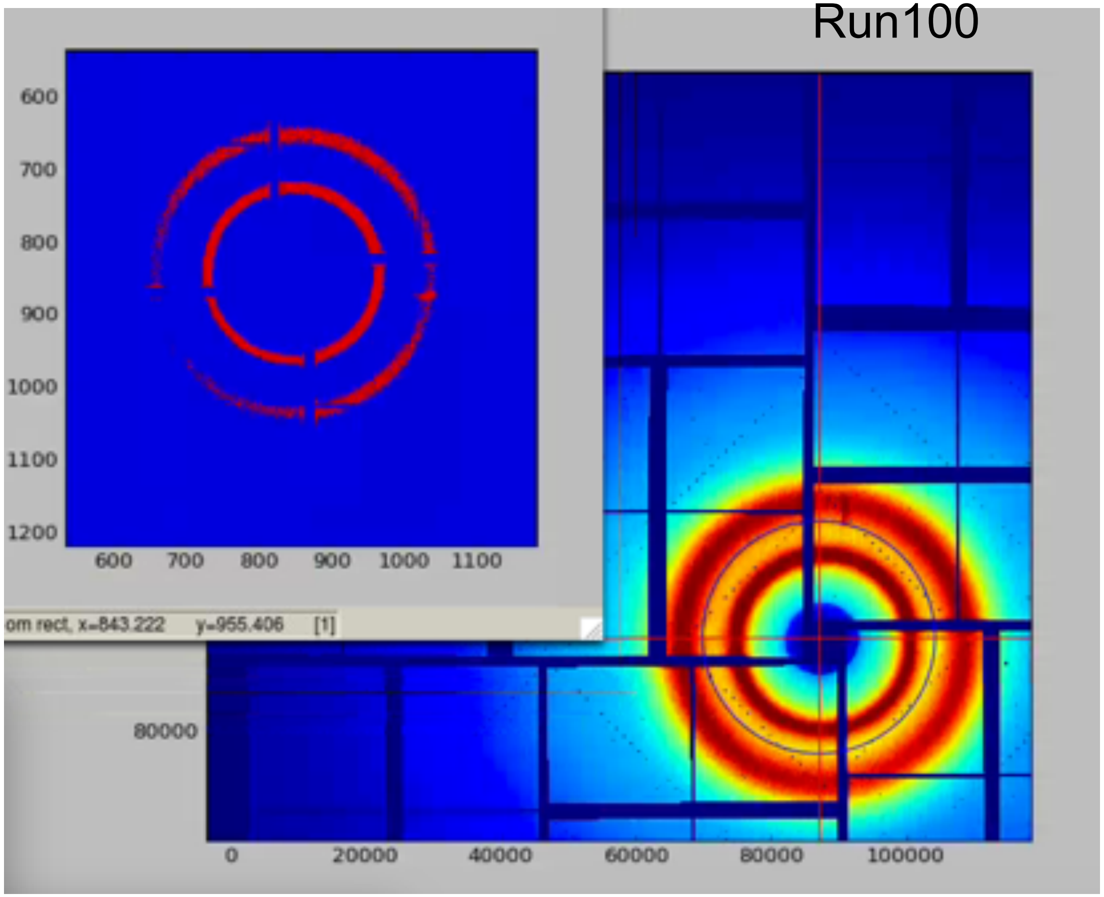
need simple, stable method to get it from image

For most experiments, we will start with a decent guess, but reprocess later.

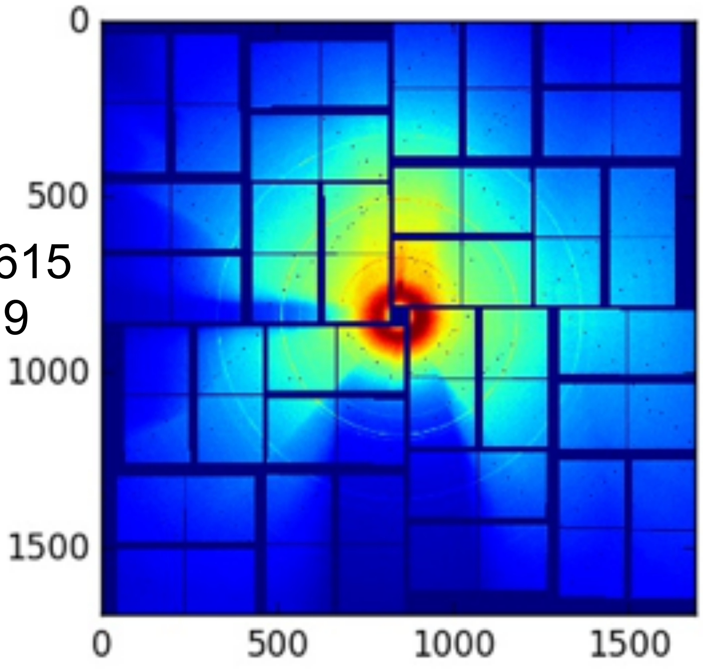
Extract Features

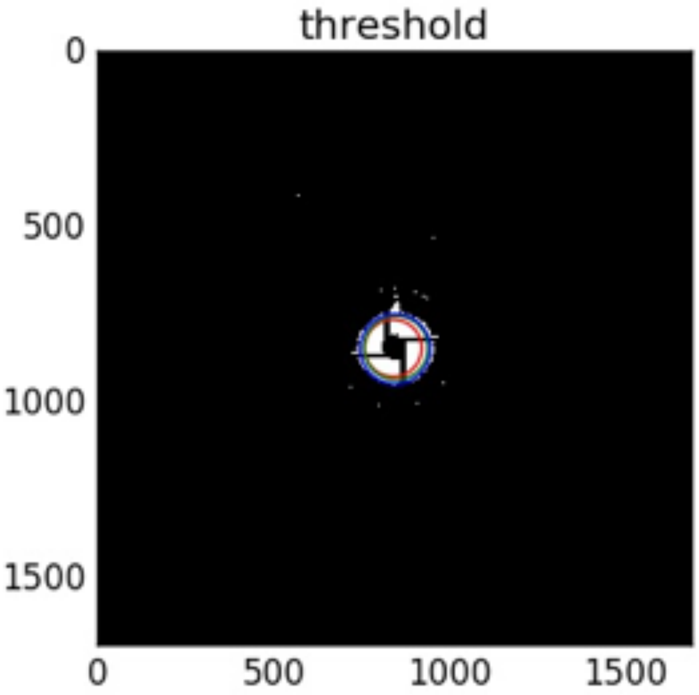
simple threshold is not easy….

xpplq0815
Run100

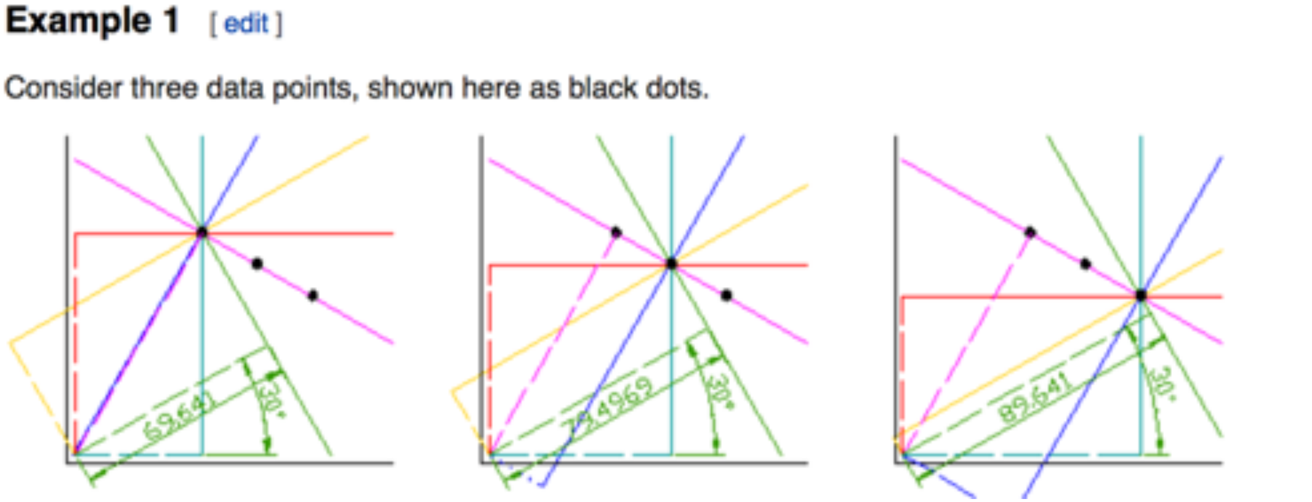


xppo5615
Run619





Hough Transform



Example 1 [edit]

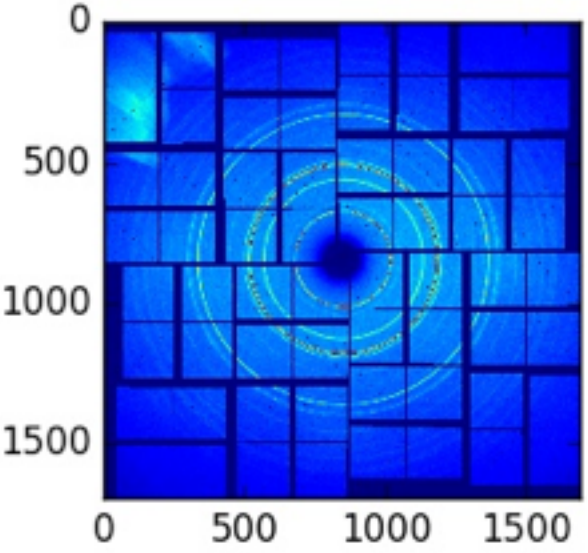
Consider three data points, shown here as black dots.

Example: for each point in x/y space, make entries in r/phi space for all possible lines through point. Fill histogram. Find maximum/maxima.

Here:

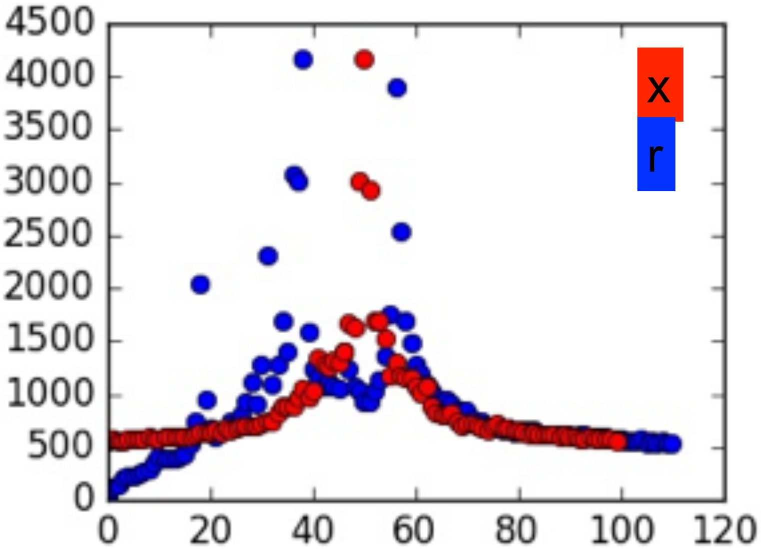
Hough transform from x/y space to R/x_center/y_center.

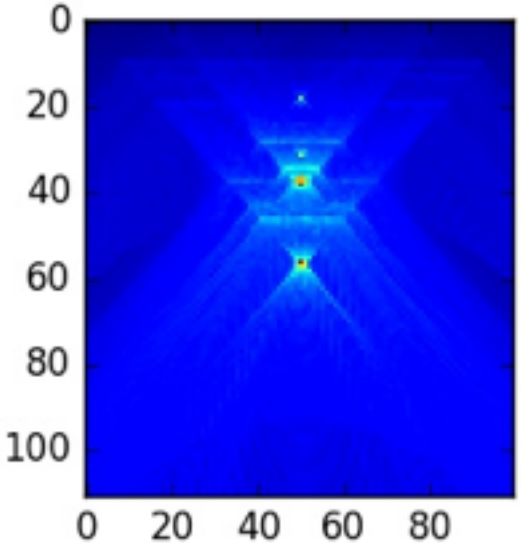
Hough Transform



Hough transform from x/y space to R/ x_center/y_center.

Use iterative procedure (full space for R, iterate for center position)

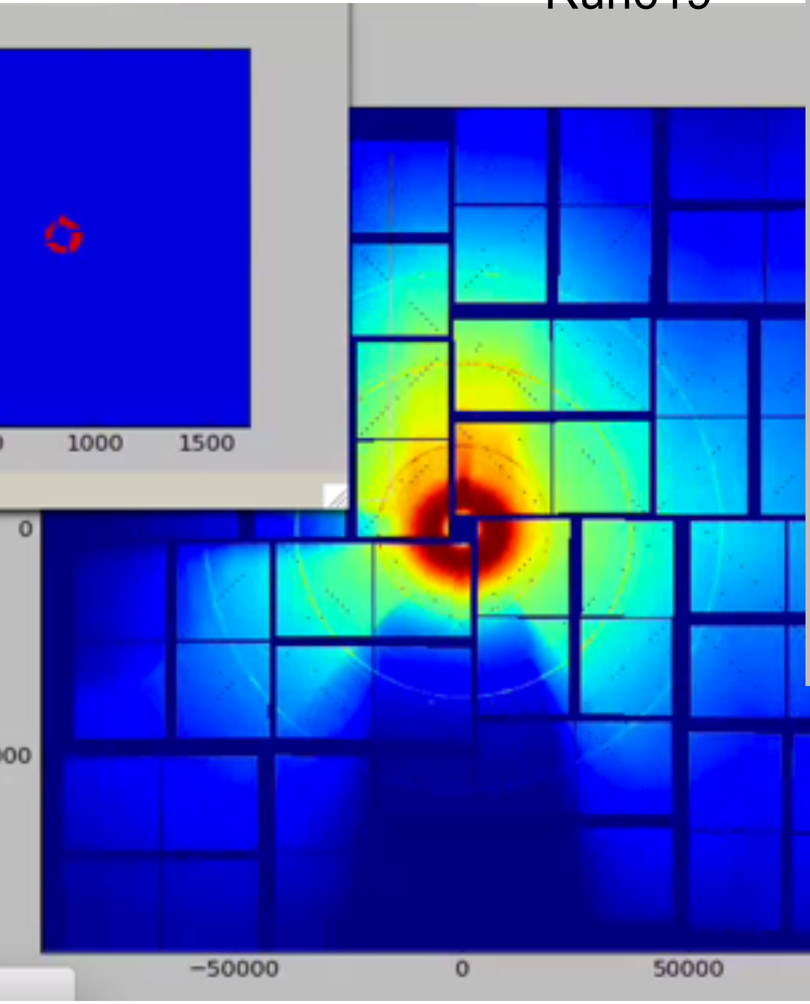


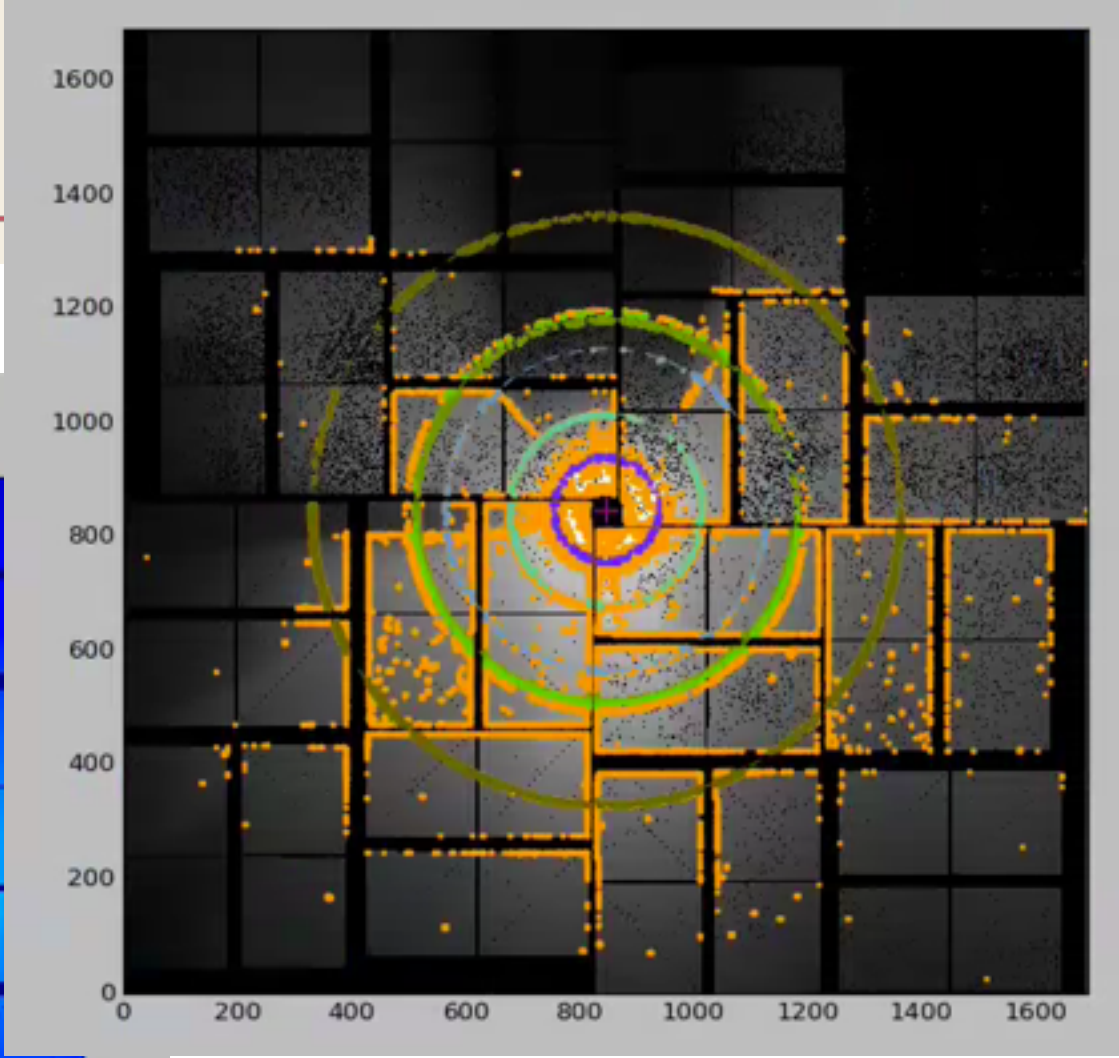


For speed: need to select point to use for Hough transform

Find Points

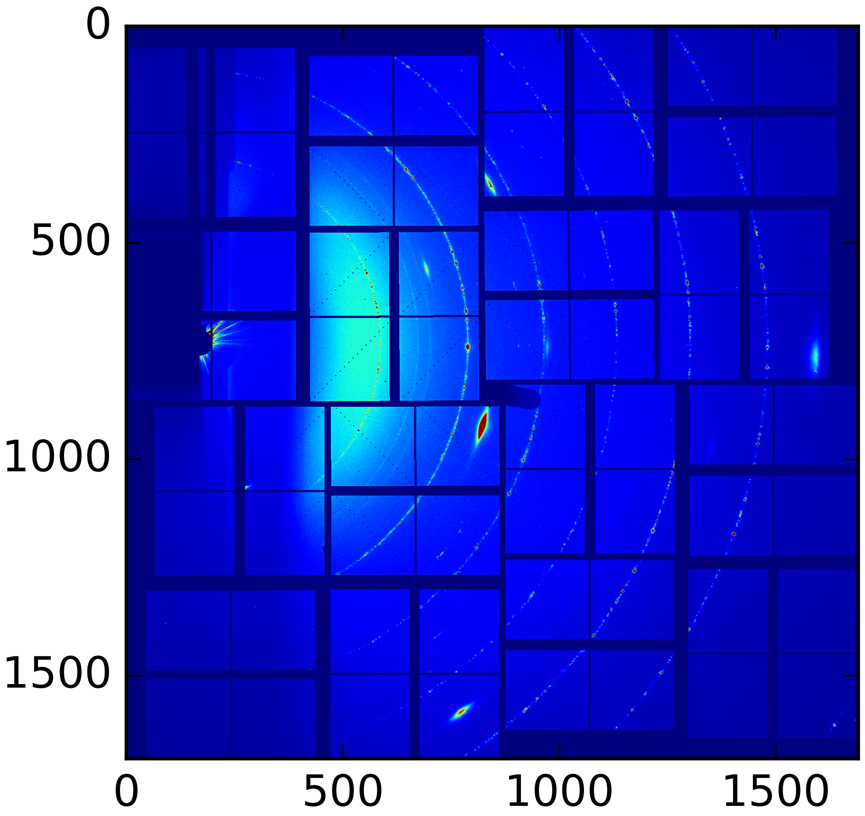
xppo5615
Run619

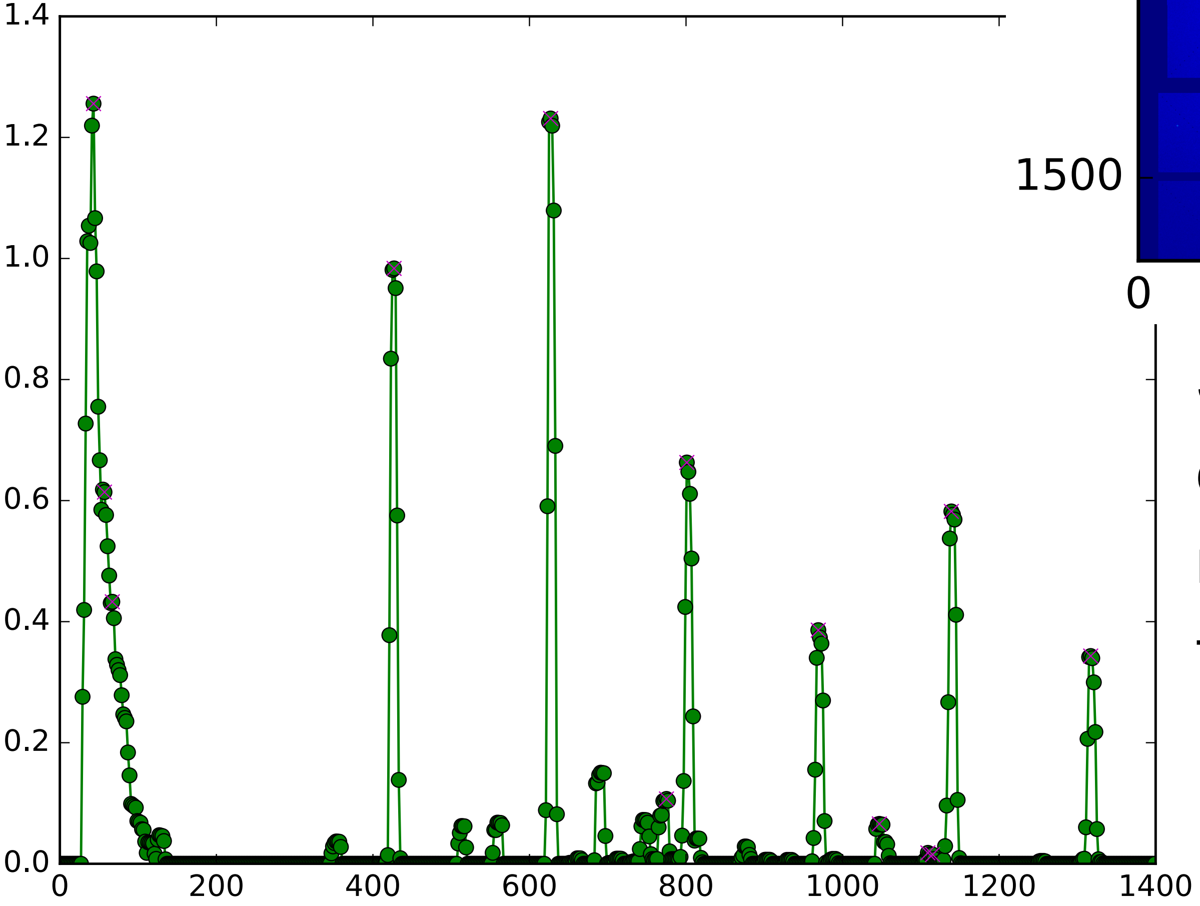




use canny algorithm from skimage.feature

xppo3016, Run 386



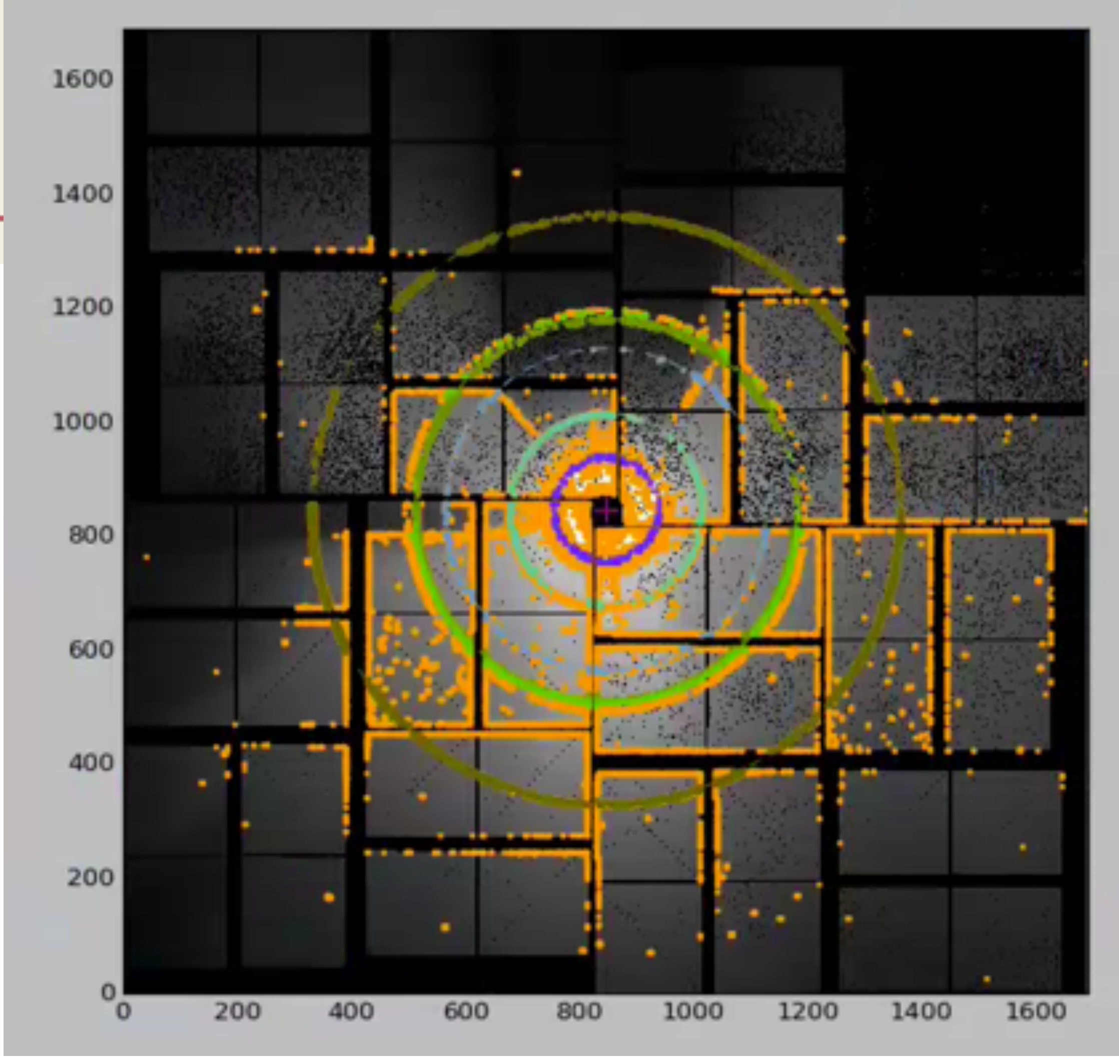


Select points around candidate rings, use ransac to select points that fit will to circle

Select Points for fit

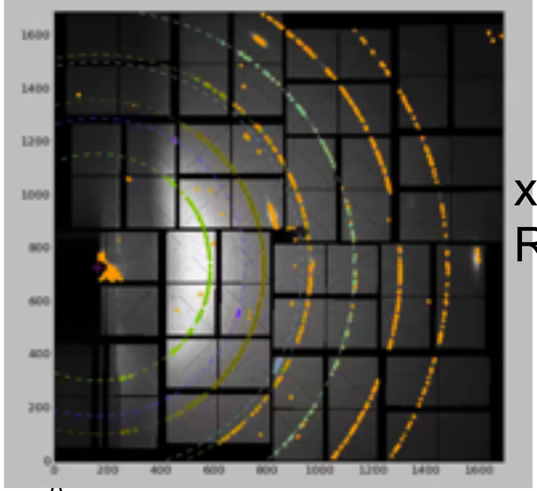
xppo5615
Run619

- use points in “donut” around found radii
- use ransac to find points for circle fit
- fit all point together in a fit to a common center, but different radii (one per circle candidate)

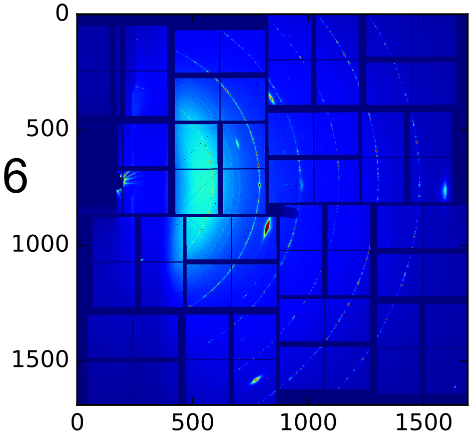


use canny algorithm from skimage.feature

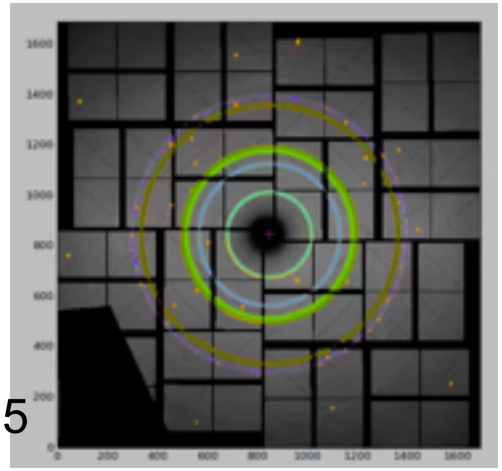
Example Images.

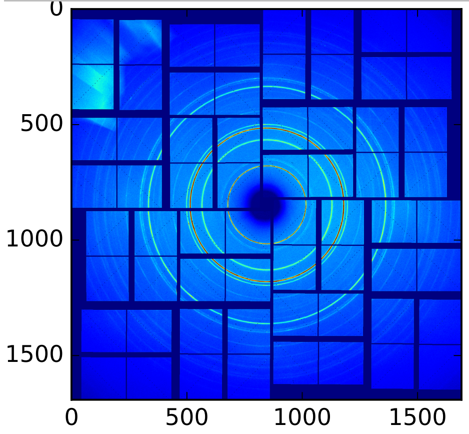


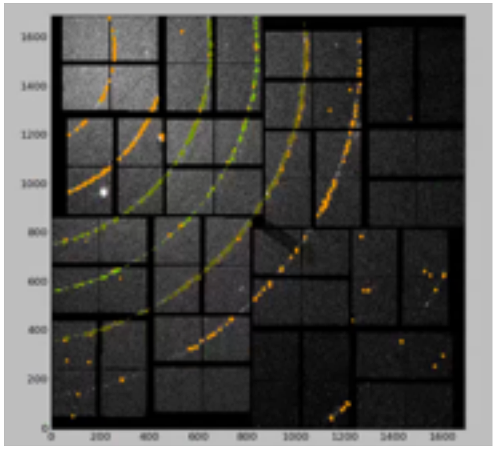
xppo3016
Run386



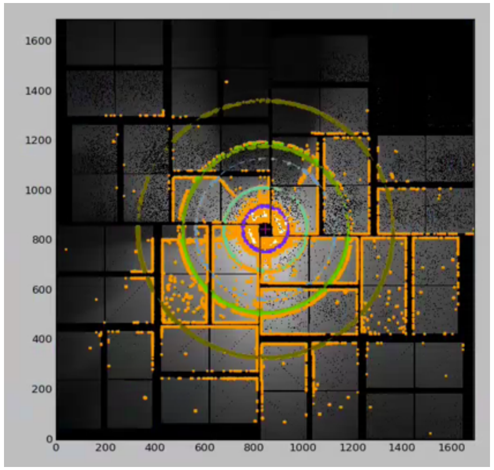
xppo5616
Run716

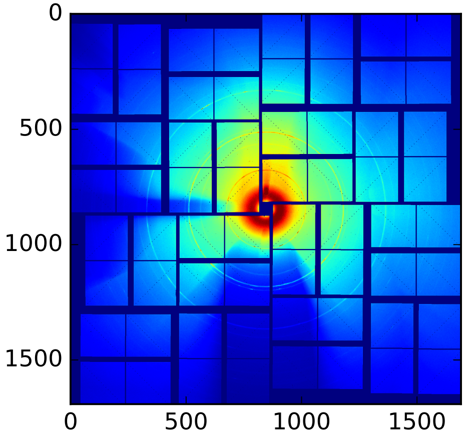




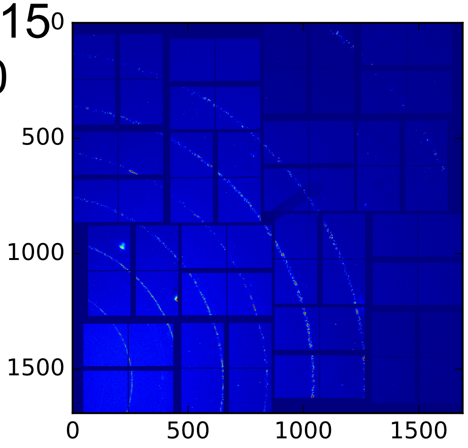


xppo5615
Run619





xpph6615
Run190



Final Fit & onwards

We developed a fairly fast and parameter free algorithm to find the beam center.

Will use the radii of the final fit to determine detector-sample distance for known sample (will need sharp circles for that)

There are two conditions where this approach fails:

very faint/wide rings

data with an obvious beamstop that is not masked out

Depending on the setup with calibration samples, this may not warrant further work

Fitting - scikit-beam auto_center_

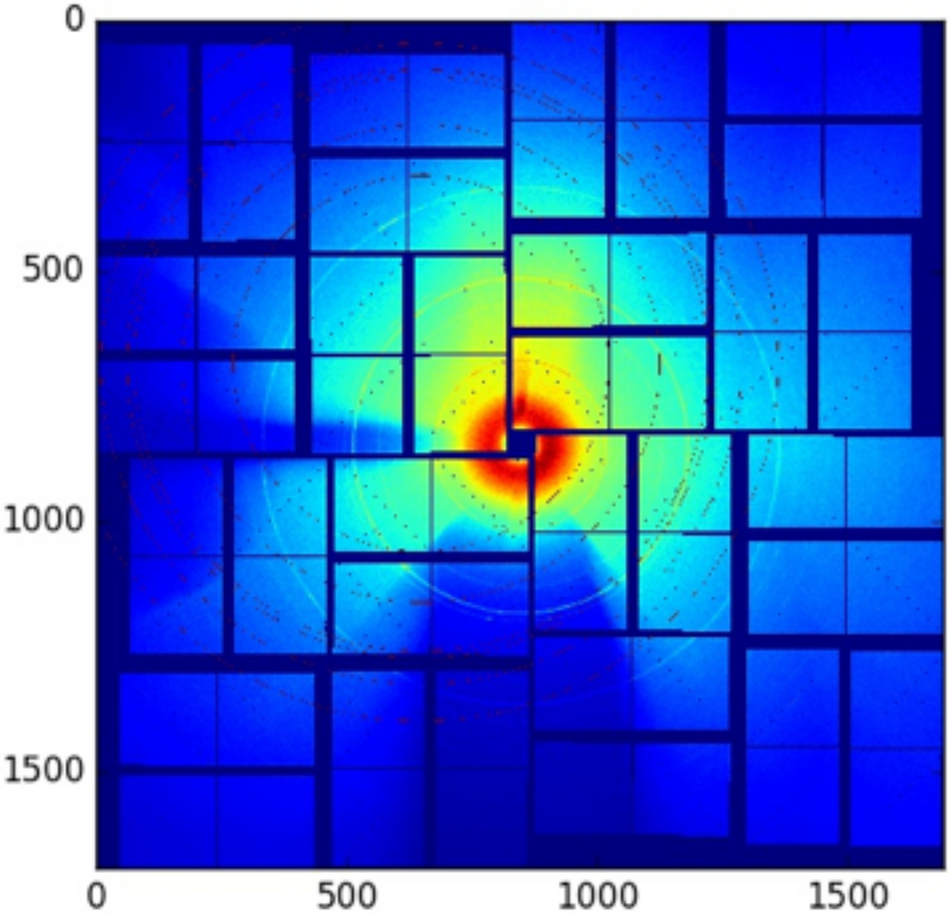
does not work at all.

looked at code: uses default canny and takes no mask (does not work well for our samplws). Then uses ransac to fit circles.

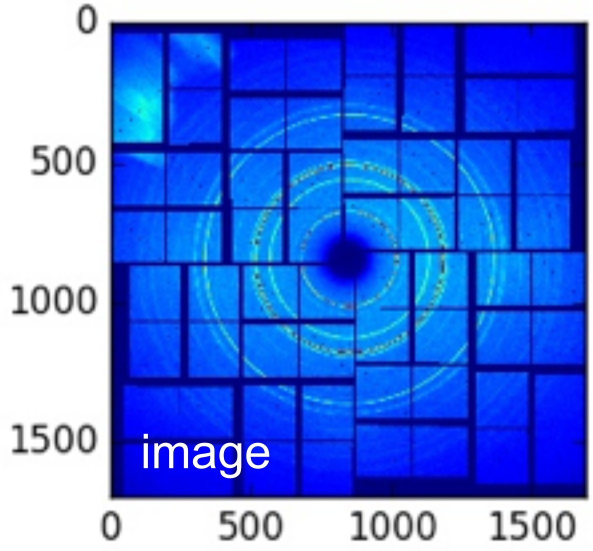
use canny with “stable” parameters & mask.

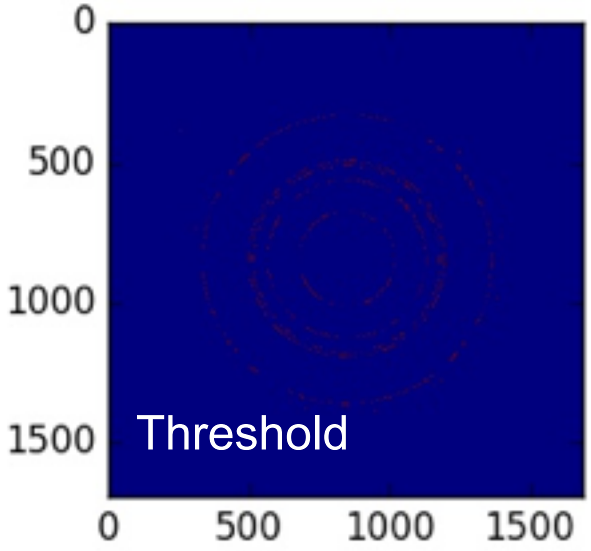
Play with ransac: does not find our circles.

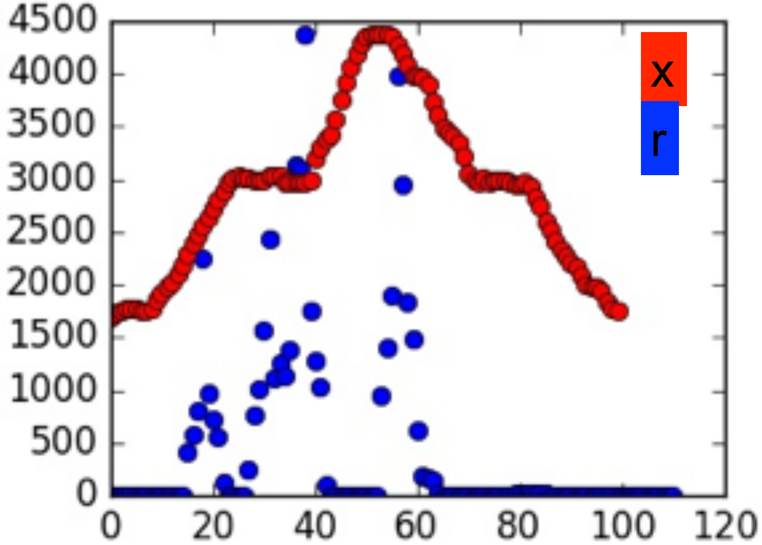
Canny results in two rings for every “real” ring. Ransac picks points from both mixed and for a set of attempts it will result in different centers.

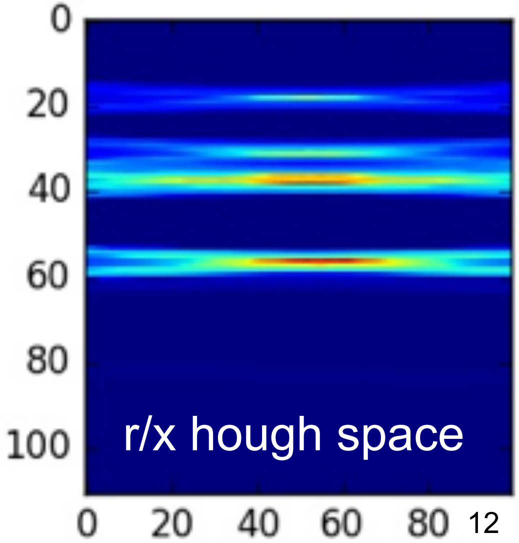


Hough - Last iteration









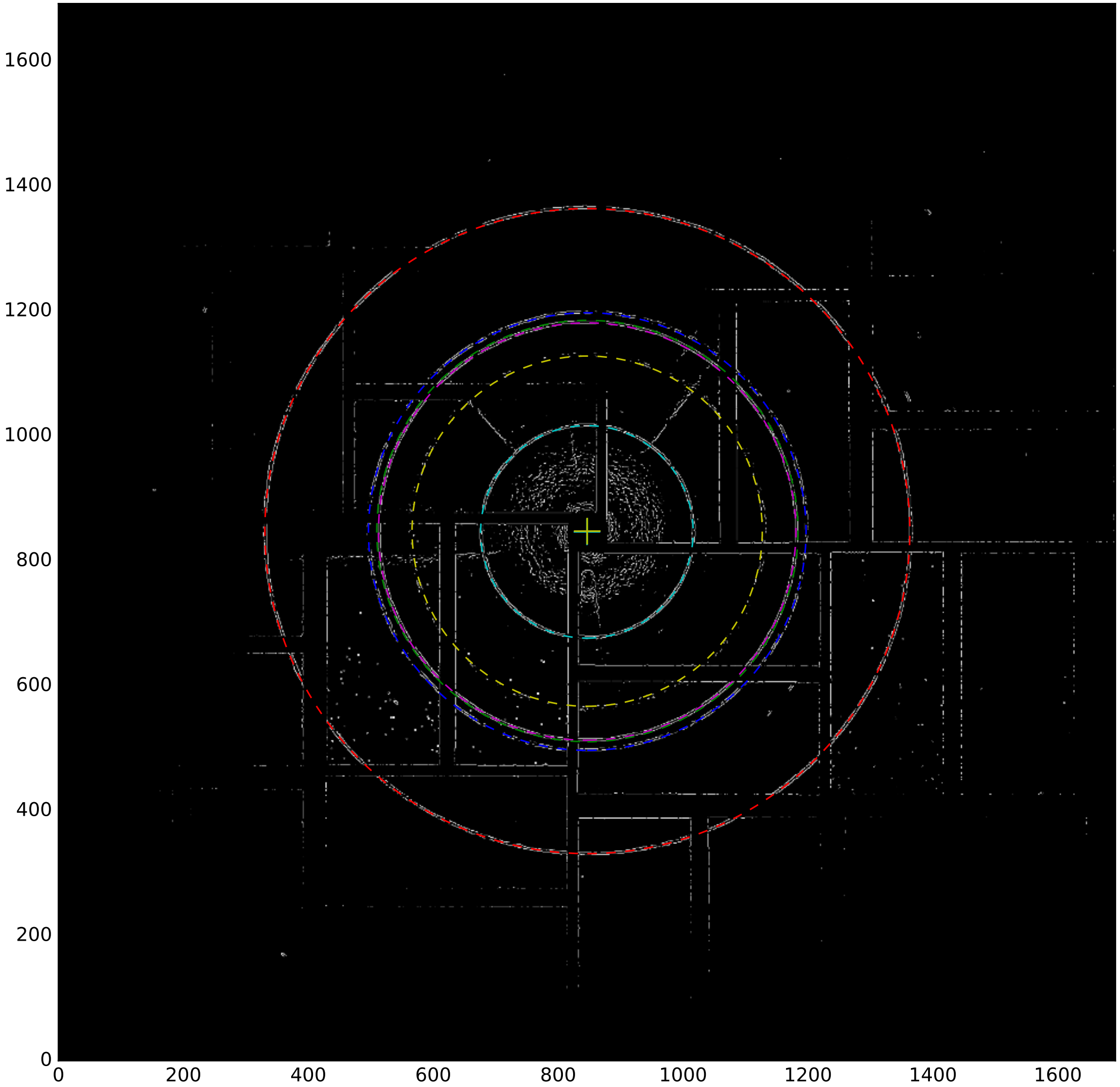
And now:

use radii & center from hough to select points in donut rings.

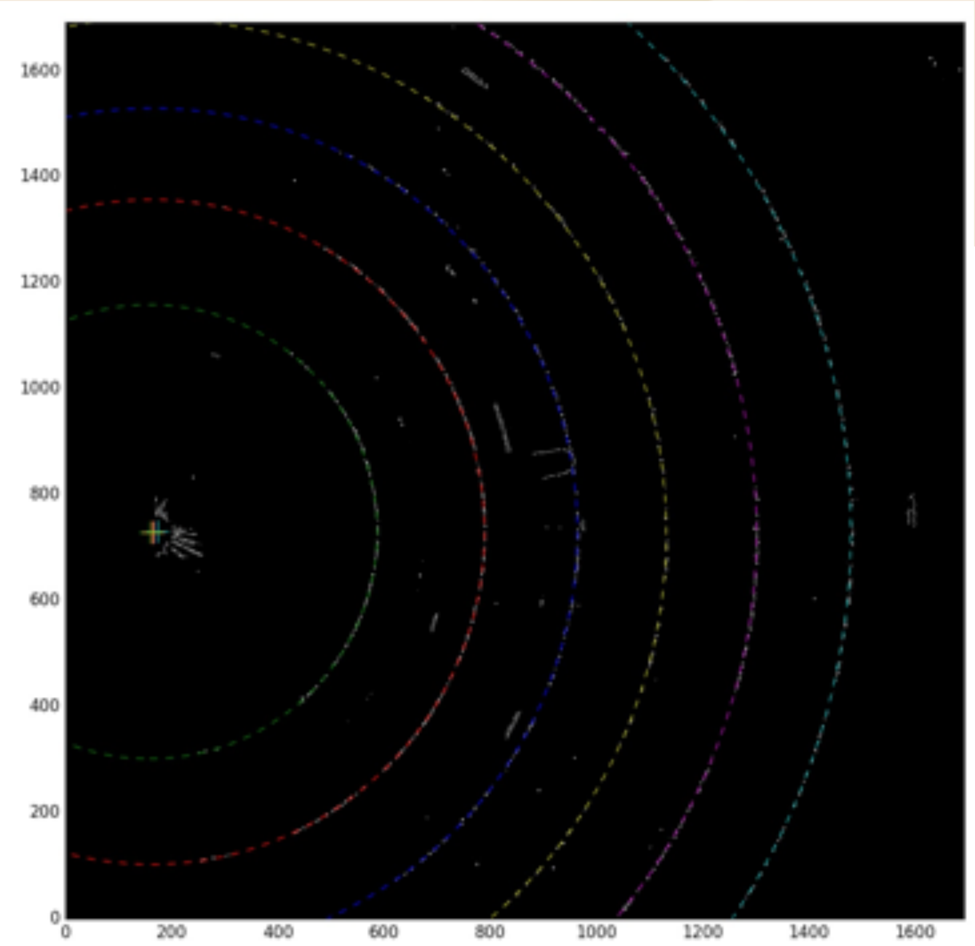
Ransac to reject outliers.

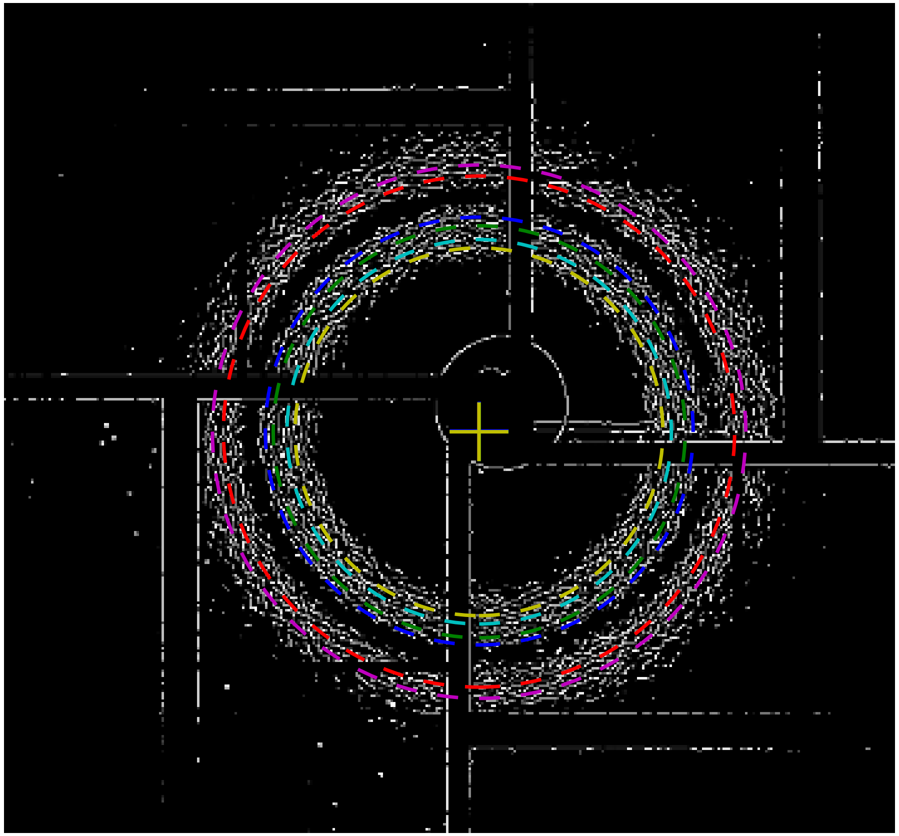
Optimize to fit center & r for each ring.

Example Image



More Examples.





Backup

Canny for Features

xppo5615
Run619

Threshold

def

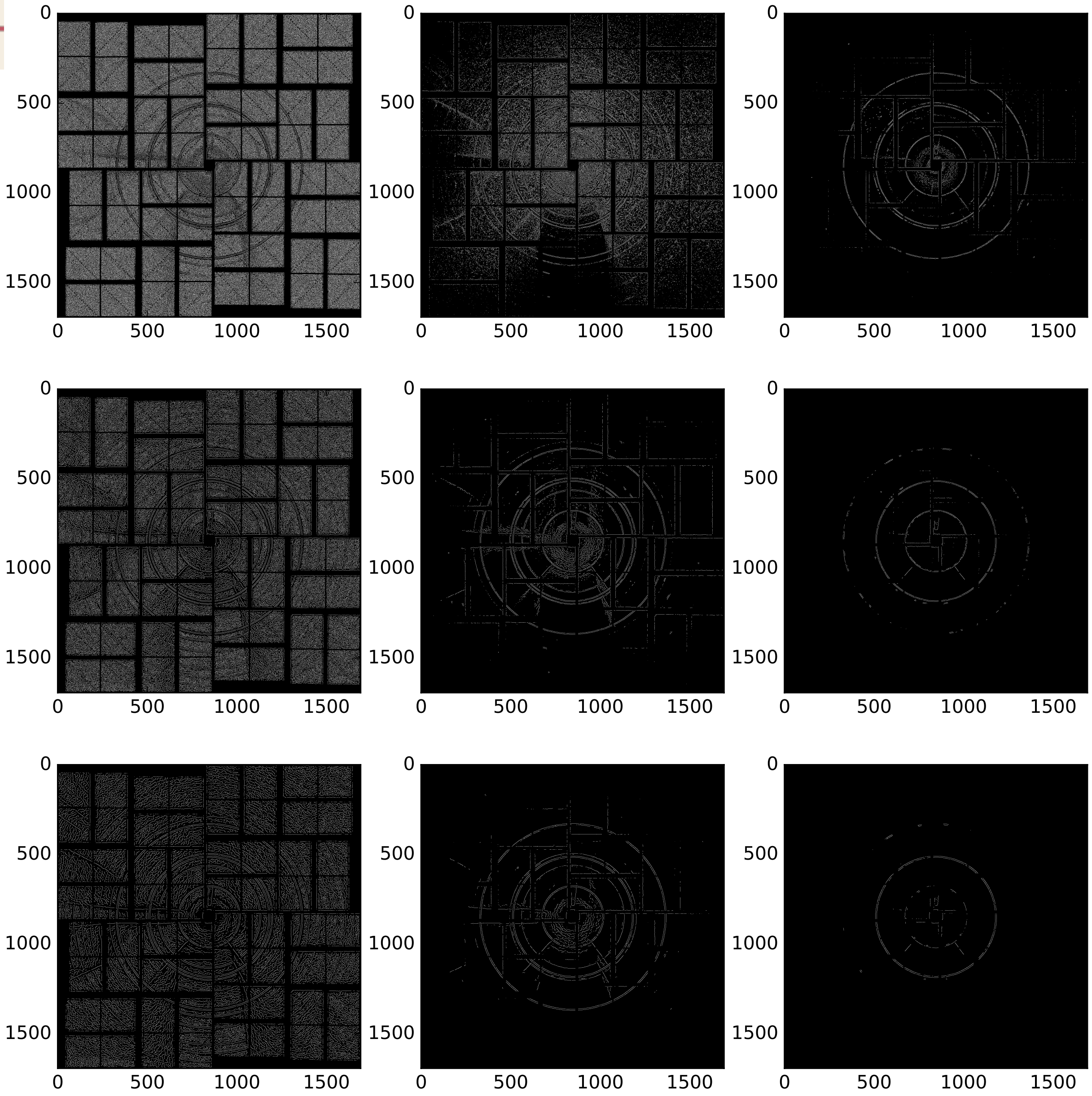
30

95

1 Sigma

2 Sigma

3 Sigma


0
500
1000
1500
0
500
1000
1500