Image Processing For Soft X-Ray Self-Seeding

[1]Prajjwal Dangal, [2]Dorian Bohler

[1]Department of Electrical Engineering and Computer Science, Howard University, 2400 Sixth St., NW, Washington, D.C., 20001, USA   Contact: prajjwal.dangal@bison.howard.edu

[2]Linac Coherent Light Source, SLAC National Accelerator Laboratory, 2575 Sand Hill Road, Menlo Park, CA 94025, USA.   Contact: dbohler@slac.stanford.edu

Motivation

- Automate the machine setup for SXRSS experiment. Hence, cut down experiment setup time.
- Right now the process of steering the camera towards the slit location is manual; one has to find the slit in space and align the camera accordingly. But given slit location, it could be automated.
- Investigate image processing techniques that would be useful in other projects like TTO (Transfer TO Operations).

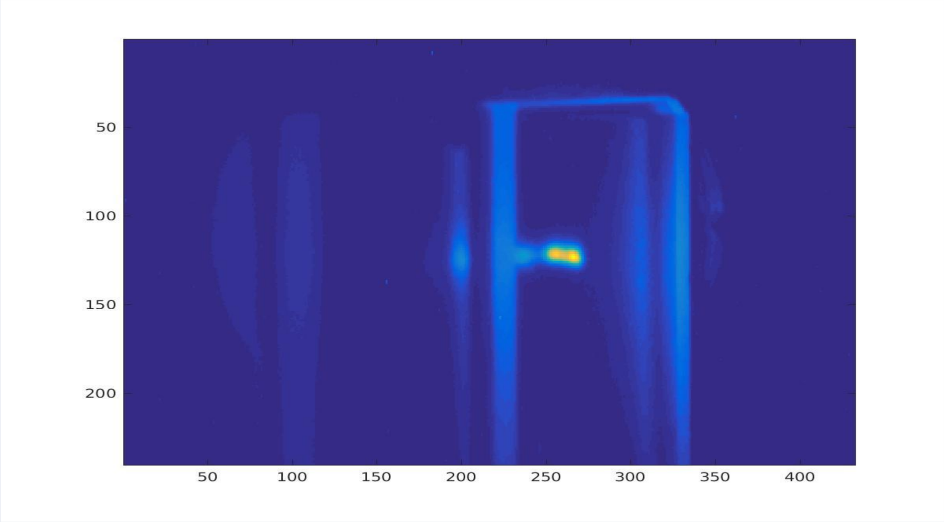

Fig 1: A sample of Yag Slit Image with beam and slit.

Objective

There are three objectives of the MATLAB program:

1. Find if a beam is present in a given image
2. If yes, return the location of the beam in x y plane. Returns [-1 -1] if no beam present.
3. Find the location of the slit in x y plane. Returns -1 if no slit present.

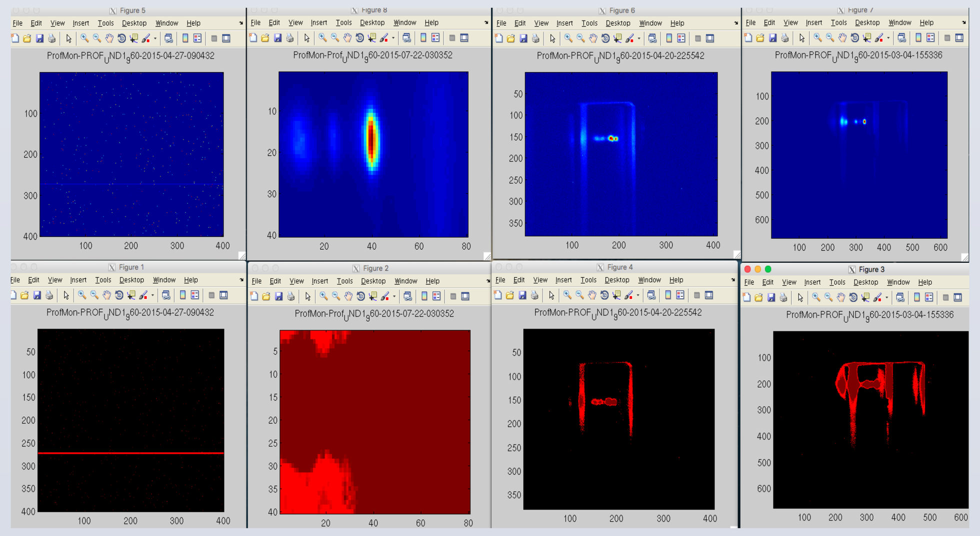

Fig 2: Sample images before and after RGB filtration.

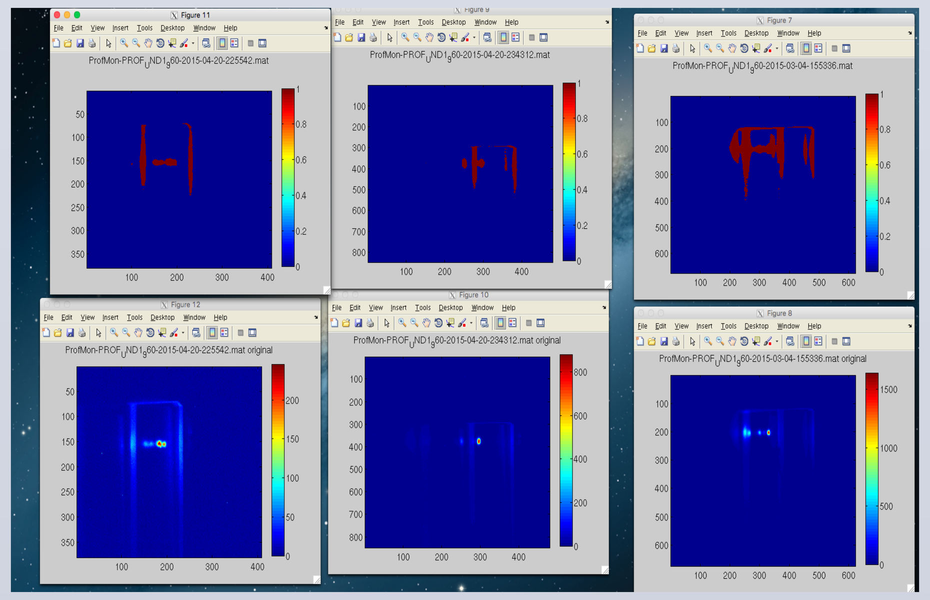

Fig 3: Sample images before and after conversion to B&W (median pooling.)

Methods

Peak Finding Method:

This algorithm converts given image (indexed) to RGB (Red, Green, Blue) and sets g and b pixel values of all the pixels to 0. The region of interest (ROI) of these Yag Slit images have r > 100.

The resulting image looks like following:

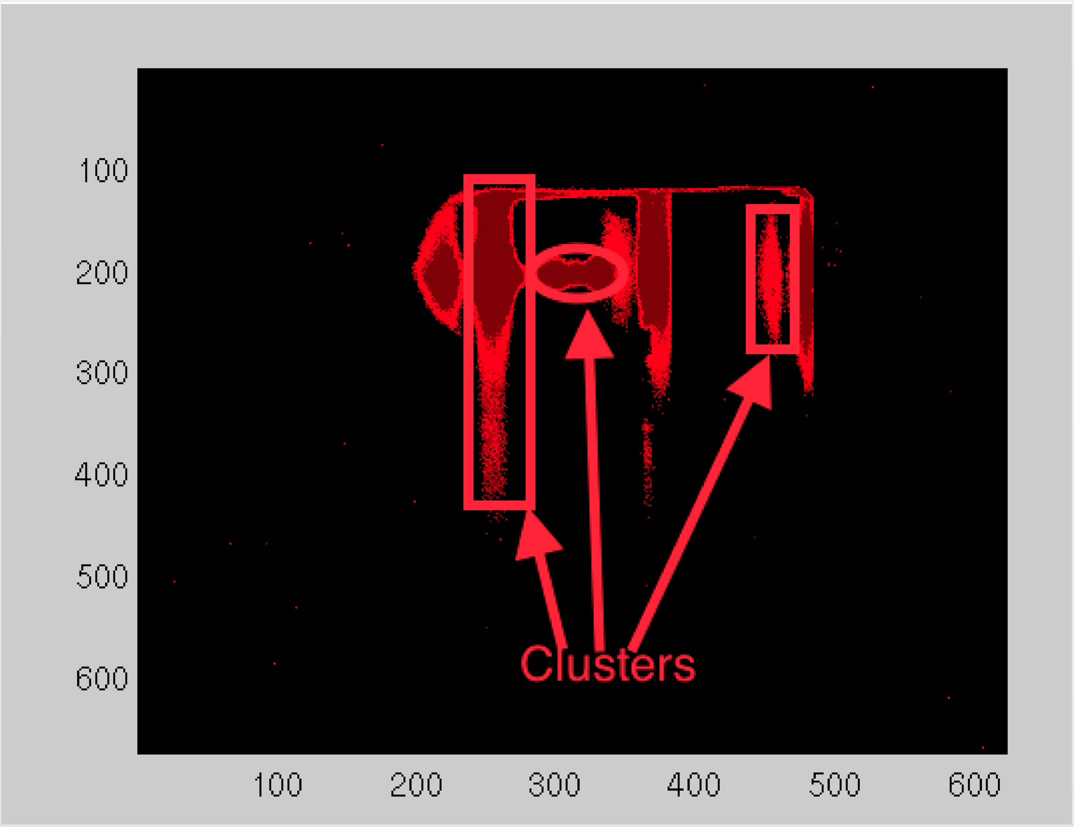

Fig 4: RGB filtered image is treated as a collection of clusters.

This is a binary image. Peaks are defined to occur at any x value where the corresponding number of non-zero pixels along the y-axis, $n > 0.1*\max(n_i)$,

$i = 1, 2, .. k$ where k is total number of such x values.

Presence of beam (range $\mu - \sigma$ to $\mu + \sigma$):

In order to determine the presence of beam, we define Δi = max_instensity – ave_intensity for a given image. Using the sample of 84 images, I calculated Δi and the mean μ and standard deviation σ over Δi. Using the calculated μ and σ for either beam/no beam image samples, I used the range $\mu - \sigma$ to $\mu + \sigma$ to classify the whether or not beam is present.

Grayscale -> Black & White -> Peak Finding:

This methods includes converting indexed images to grayscale to black and white. The later conversion is done using a threshold value. After that, the same peak finding algorithm can be applied in the binary image.

Apart from these, changing color spaces, applying median pooling, using thresholding techniques were some of the other methods I used while playing around.

Results

Feature Detected	**Method Description**	**Accuracy**
Beam	Find max intensity point (y)	28.9474
	Find max intensity point (x)	34.2105
Beam	RGB -> filter pixel -> peak finding (y)	82.8947
	RGB -> filter pixel -> peak finding (x)	81.5789
Presence of beam	Range μ - σ to μ + σ	88.1579
Slit	Grayscale -> B&W -> peak finding	68.4211
Slit	Index img -> thresholding -> peak finding	78.9474
Slit	RGB -> filter pixel -> peak finding	81.5789

Conclusion

As one can see, the best results were given by the RGB -> filter pixel -> peak finding method. For beam detection, the algorithm incorporates comparison to see if the max intensity point falls in one of the clusters. Hence, this could be seen as an example of making use of multiple features.

Using factory built toolboxes (like the machine learning and statistics toolbox for clustering) could have resulted in better performing and more accurate MATLAB program.

Further Work

Naturally, the more features I add, the more accurate the detection becomes. Assigning weights to the feature is another tweak that could improve the accuracy and accommodate varied image sample.

I could improve the way threshold values are calculated. Right now, it's hardcoded to a reasonable value. But they could be calculated separately for each image.

The algorithm could be optimized further and several scripts and functions could be made more modular. Finally, the tool will be used in experiments with real machine once we get beam time.

Acknowledgement

I would like to thank my mentor for the summer, Dorian Bohler for his guidance and support, Prof. Katona from Dept. of Physics, Howard University for explaining several physical concepts pertaining to the project, Tim Maxwell for suggesting methods in MATLAB, David Schneider for help with adaptive algorithms part (which is the other part of my summer project), Dr. Alan R. Fry for coordinating everything from the beginning of the summer to the end and the Department of Energy for providing us, the interns with this opportunity.