Image Processing For Soft X-Ray Self-Seeding

1Prajjwal Dangal, 2Dorian Bohler

1Department of Electrical Engineering and Computer Science, Howard University, 2400 Sixth St., NW, Washington, D.C., 20001, USA  
Contact: prajjwal.dangal@bison.howard.edu  
2Linac 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).

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 [x - y] if no beam present.
3. Find the location of the slit in x y plane. Returns -1 if no slit present.

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 t > 100. The resulting image looks like following:

![Fig 1: A sample of Yag Slit Image with beam and slit.](image1.png)

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

Peak Finding Method:

Para of beam (range μ - σ to μ + σ)

Presence of beam

In order to determine the presence of beam, we define ΔI = max intensity - 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 each beam/no beam image samples. I used the range μ - σ to μ + σ 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.

![Fig 2: Sample images before and after RGB filtration.](image2.png)

Fig 2: Sample images before and after RGB filtration.

![Fig 3: Sample images before and after conversion to B&W (median pooling.)](image3.png)

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

![Fig 4: RGB filtered image is treated as a collection of clusters.](image4.png)

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

Methods

Results

<table>
<thead>
<tr>
<th>Feature Detected</th>
<th>Method Description</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Beam</td>
<td>Find max intensity point (y)</td>
<td>28.9474</td>
</tr>
<tr>
<td>Beam</td>
<td>Find max intensity point (x)</td>
<td>34.2105</td>
</tr>
<tr>
<td>Beam</td>
<td>RGB → filter pixel -&gt; peak finding (y)</td>
<td>82.8947</td>
</tr>
<tr>
<td>Beam</td>
<td>RGB → filter pixel -&gt; peak finding (x)</td>
<td>81.5789</td>
</tr>
<tr>
<td>Presence of beam</td>
<td>Range μ - σ to μ + σ</td>
<td>88.1579</td>
</tr>
<tr>
<td>Slit</td>
<td>Grayscale → B&amp;W → peak finding</td>
<td>68.4211</td>
</tr>
<tr>
<td>Slit</td>
<td>Index img → thresholding → peak finding</td>
<td>78.9474</td>
</tr>
<tr>
<td>Slit</td>
<td>RGB → filter pixel → peak finding</td>
<td>81.5789</td>
</tr>
</tbody>
</table>

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.

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 intern with this opportunity.