Beam characterization code setup

# Summary:

The BCS code include 0) preparing all inputs into the same datatype of uint8, 1) edge finding capability for CENER and Sandia data, 2) Rectification with perspective transformation to bring the target into a square, 3) Low pass filtering with FFT to get rid off some background noises, 4) Gamma correction to suppress the effect of background pixels 5) centroid finding to locate the beam center. Not all functions are necessary for center location finding. This flow is illustrated in Figure 1. Not all images require all steps.

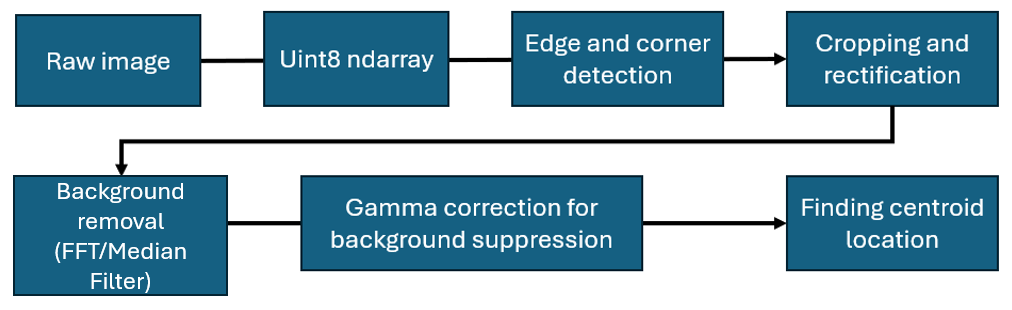


Figure . Flowchart of steps to finding beam center location.

# Attentions:

The edge finding and FFT/Median filtering may not work for every circumstance, the edge detection could fail to detect every edge wanted, especially where there are complex backgrounds.

FFT low pass may not work for every circumstance, especially when the undesired feature stands out very much. Pushing the power of low pass filtering could lead to reduced actual features. Figure 2 Illustrate one of the circumstances. Median filter could provide better filtering. But for backgrounds that contain non-wanted patterns across all areas, FFT could provide better filtering.

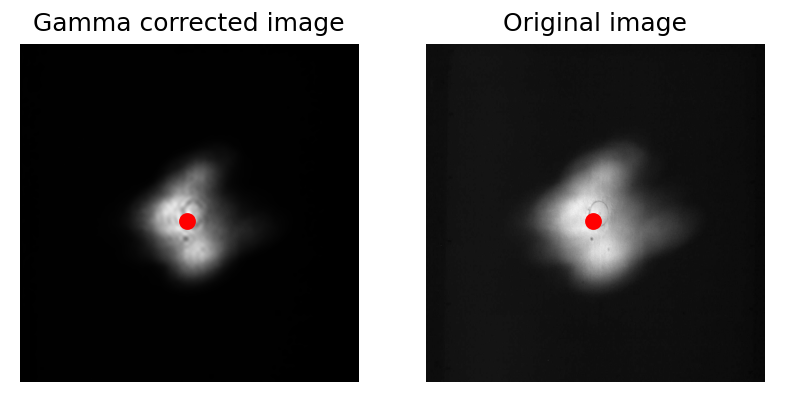


Figure . Illustration of background noise suppression with FFT low pass filtering. Image from Sandia. Left one shows the gamma corrected FFT low passed image, kept 2% of all low frequency components. Any more compression will reduce beam pattern feature that led to unreliable readings.

# Methods:

## Preprocessing:

Not all data can be directly processed by the main processing code directly due to data type incompatibility. The data type works with the rest of the system is uint8 type, single channel matrix. Reading TIF images is known to cause incompatibility, and right now is being handled by the code

## Edge and Corner Detection:

The edge and corner detection remains unchanged from the previous code. Image pixel adjustment with alpha and gamma correction, together with blurring and morphological operations are used to make the edge stands out more. Lastly, Hough transform is used to get the line, slope ranges looking for horizontal and vertical are applied to filter out some undesired edges. Intersections are used to find the corners.

This method will fail with some complex backgrounds. In the round roubin test, Sandia and PSA data does not need this feature.

## Target Rectification:

After finding the corners of the target, the area of interest is rectified and cropped out of the original image. The cropped image is them rectified into a 1000x1000 square with perspective transform

## FFT low pass filtering:

The FFT low pass filtering is design to reduce unwanted background features on the BCS target. This approach relies on the fact that the significant features of beam are on low frequency components. Most unwanted background noises, including ridges are high frequency components. The code provided keeps the coefficients related to signals with lowest 10% frequency.

## Median Filtering

Median filtering is an edge-preserving filtering technique that replaces pixels within a kernel with the median values of all pixels in the kernel. Since most of the background noises are slim lines, median filter could be effective in removing backgrounds.

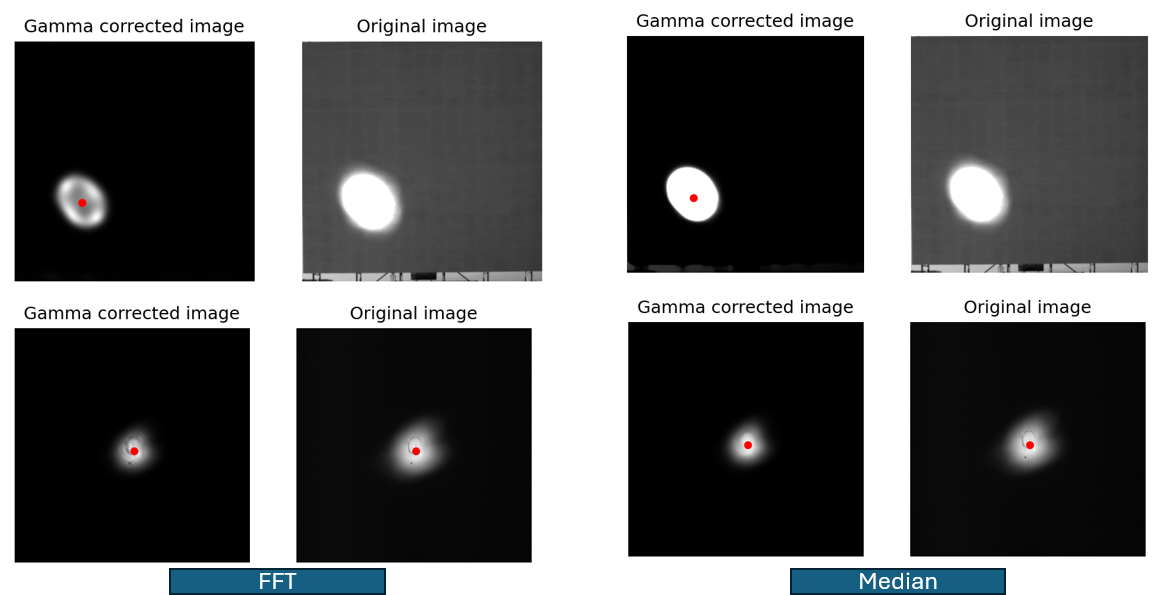


Figure . Comparison of FFT low pass filtering and median filter. FFT low pass filter may experience difficulties in removing some strong edges in the images, and may bring unexpected artifacts. Median filtering is more stable. However, median filtering could be less effective when target cropping have errors, as shown in the top row.

## Gamma Correction

Although the filtering has suppressed much of the unwanted features, backgrounds still posses non-zero values. Direct thresholding approaches could lead to unwanted features because of the inappropriate choice of threshold values. Therefore, non-linear scaling can produce more natural images, and gamma-correction was applied in this case. The gamma value was set to 5 in the provided code. This process suppressed the background noises by making the values of the black pixels higher.

## Beam Center

The beam center was calculated as the centroid of the image after gamma processing. Not only does this calculation consider beam shape, but it also considers intensity distributions. This could provide more accurate results compared with ellipse fitting.

## Tracking error calculation

The tracing error was calculated based on vector math, with dot product, specifically

Where the ideal is the vector pointing to the target center, and the actual is the vector pointing to the center calculated.

The following table summarizes the tracking error calculated with 1) FFT low pass processing, and 2) median filtering processing

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Image Source | PSA-1 | PSA-2 | PSA-3 | CENER | SANDIA-1 | SANDIA-2 | SANDIA-3 | SANDIA-4 |
| median filter | 0.59 | 1.83 | 0.79 | 17.03 | 2.84 | 3.08 | 2.71 | 1.49 |
| FFT | 0.59 | 1.83 | 0.79 | 16.63 | 2.80 | 3.09 | 2.71 | 1.49 |

Results from this table indicates that there is little to no difference between the two processing methods when the light spot is near the center. Even when the light spot is far away, plus some cropping errors, around 2.4% differences between these two approaches.

# Uncertainty Analysis:

The uncertainty values are obtained by disturbing the input images. Random Gaussian noises are added to the images so that edge detection, the calculated centroids are not deterministic and should follow a normal distribution. Confidence intervals can be taken from distribution to acquire an upper and lower limit. Any detection with overly large confidence intervals, i.e., variances, indicates that the input image is not suitable for the detection algorithm and the results should not be trusted.

It should be noted that it is very unlikely for the unprocessed uncertainties, or variances, to be calibrated straight out of the box [1]. The cumulative distribution function calibration used in [1] can be implemented to calibrate the uncertainty. To implement this approach, accurately annotated beam center is required. The accurate annotation could come from manual annotation.

[1]

V. Kuleshov, N. Fenner, and S. Ermon, ‘Accurate Uncertainties for Deep Learning Using Calibrated Regression’, in *Proceedings of the 35th International Conference on Machine Learning*, 10--15 Jul 2018, vol. 80, pp. 2796–2804.