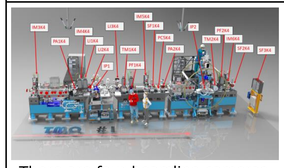
Checking Beamline Camera Alignment Using OpenCV

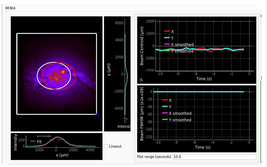
Andrew Lu

SLAC NATIONAL ACCELERATOR LABORATORY

Introduction



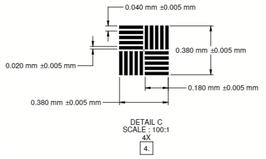
There are four beamline cameras in the TMO hutch, IM3K4-IM6K4



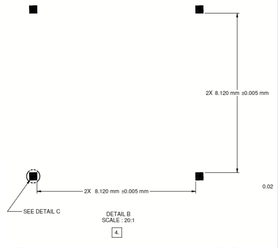
Screen used to help align the X-ray beam using pictures from a beamline camera.

Beamline cameras are used to align the X-ray beam. However, the cameras themselves can also be knocked out of their proper alignment. When checking out the beamline cameras, we can check that the cameras are still properly aligned if the position of the fiducials within the camera frame has not shifted. This process was previously done by eye, and by automating the process using OpenCV we save time and effort.

Fiducial Detection for PPM Imagers

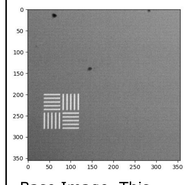


Drawing of fiducial used in PPM imagers

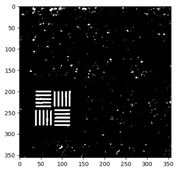


Fiducials are laid out in an 8.5mm square on the YAG crystal

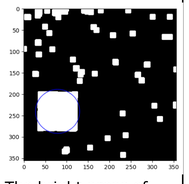
As the TMO hutch has PPM imagers, we first worked on how to detect the fiducials for those cameras.



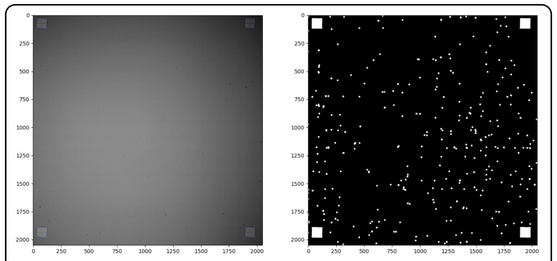
Base Image. This image is cropped from the original to give a better view of the fiducial



The image is thresholded, which makes every pixel either completely black or completely white

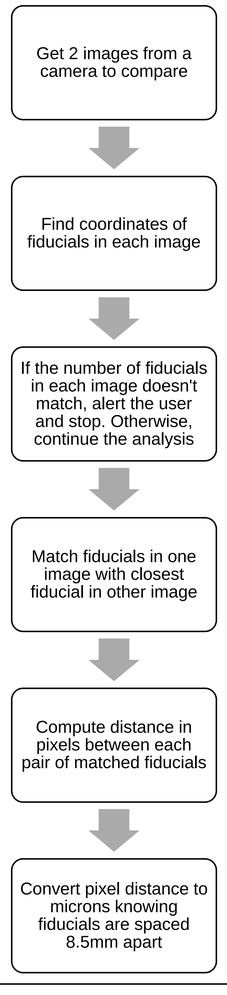


The bright areas of the image are then expanded, allowing the fiducial to be recognized using a blob detector. The blue circle indicates the location of the detected fiducial.

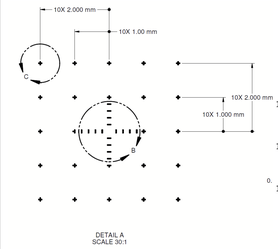


Left: Image before filtering, detected fiducials are circled in blue
Right: Image after filtering

Analyzing Camera Movement



Feature Matching: A More Generalized Approach



Layout of the XTES reticle

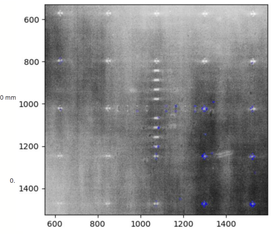
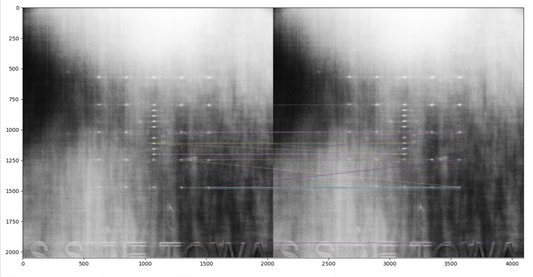
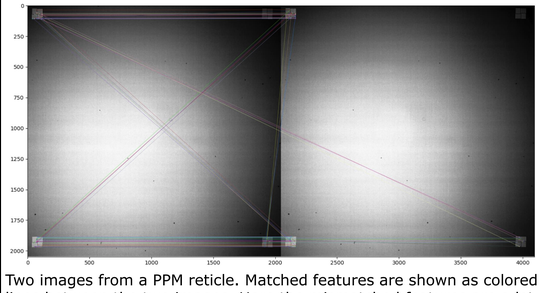


Image of an XTES reticle, with detected features circled in blue

While the blob detection approach worked well for the PPM imagers, the same approach could not be used with the XTES imagers. The PPM imagers have four relatively large fiducials, but the XTES imagers have a reticle composed of many small crosshairs, which would be much harder for the blob detector to differentiate from noise. Thus, we instead opted to use an ORB detector, which finds keypoints in the image and calculates descriptors for them. We can then match keypoints from the two images that have similar descriptors and calculate the distance between each pair of matched keypoints, like what was done before.



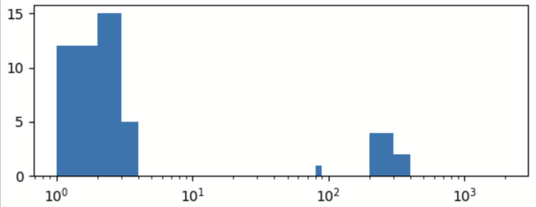
Two images from an XTES reticle. Matched features are shown as colored lines between the two images.



Two images from a PPM reticle. Matched features are shown as colored lines between the two images. Here the mismatched features are a lot more visible, as long diagonal lines between the two images.

Pros:
- Works with images from both PPM and XTES imagers
- Detecting individual features gives a lot more data points than detecting whole fiducials

Cons:
- Less control over what features get detected
- Cannot perform pixel to micron conversion
- Feature mismatch happens as feature descriptors from different fiducials may be similar



A histogram of distances between matched features. While most distances are between 1 and 4 pixels, there are some outliers due to incorrectly matched features.

Conclusions

Using OpenCV and traditional computer vision techniques, we were able to track movement of fiducials within the camera frame.
In the future, using more modern computer vision techniques, such as training a machine learning model to detect fiducials and match keypoints may lead to better results.

Acknowledgments

James Cryan and TMO Instrument Team
Tong Ju
Jing Yin