The Advanced Radiographic Capability (ARC) Python phase correlation alignment algorithm

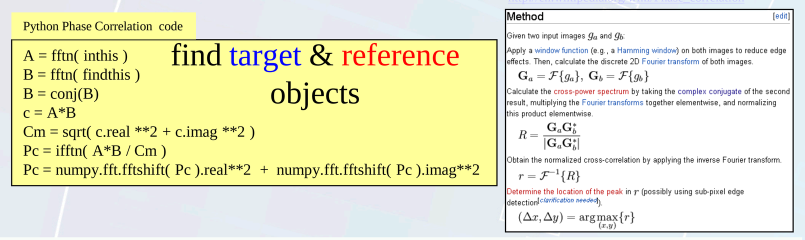
Mike Rushford and Eren Veziroglu
NIF—Photon Science and Applications
Saratoga High School

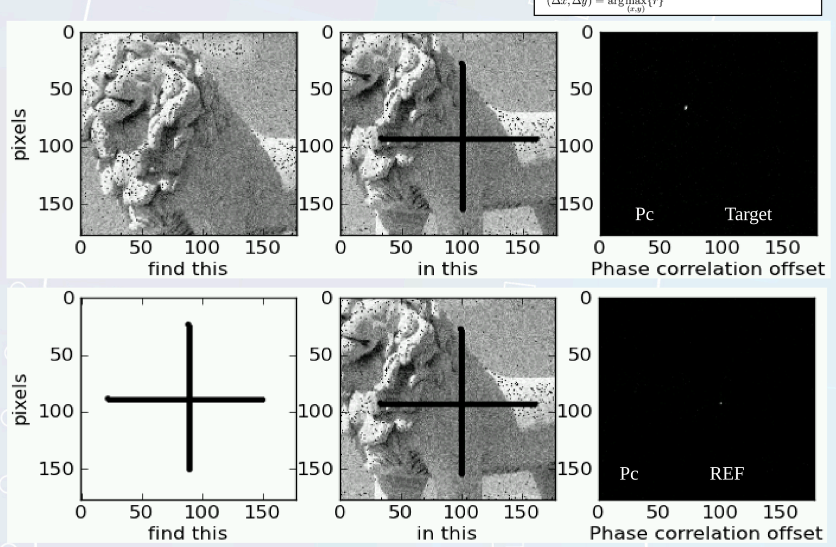
Alignment images from the ARC (Advanced Radiographic Capability) need a general purpose object location algorithm. Two objects make alignment, one of them can not be the camera pixel location as these camera are removed periodically. Thus ARC alignment images are an opaque reference object and a bright background beam pattern. Using FRED, we simulated images and used these as input to our algorithm trials.

Introduction:

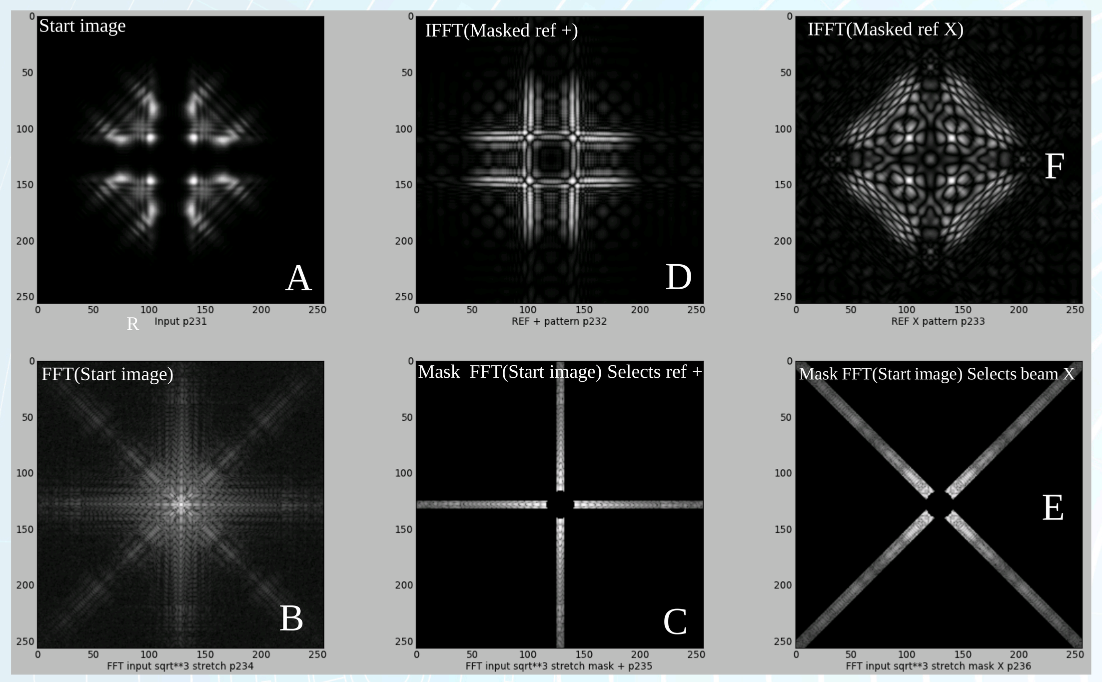
Finding two objects defines a line. Identifying those object requires training. We study Python phase correlation to train on and locate objects. FRED simulated ARC alignment images enable training and algorithm development. Masking the FFT or original image defines reference data for finding the location of each object pattern.

Begin the study using Mask applied to Wiki image





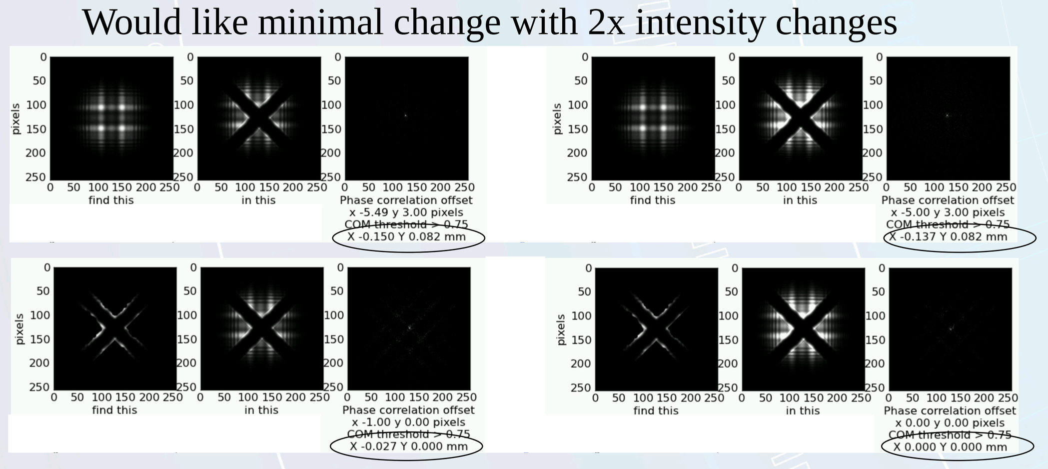
Example ARC image alignment task
Mask applied to FFT to separate Reference + and beam pattern x edges



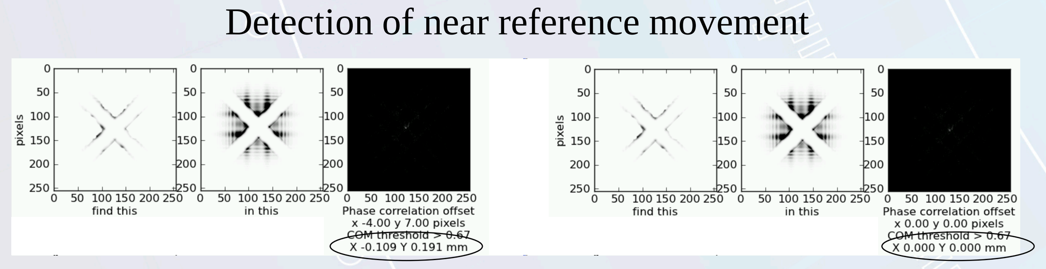
Results:

Simulated FRED images give us head start at developing our methodology to finding alignment information in ARC images. Web searches helped our Python development.

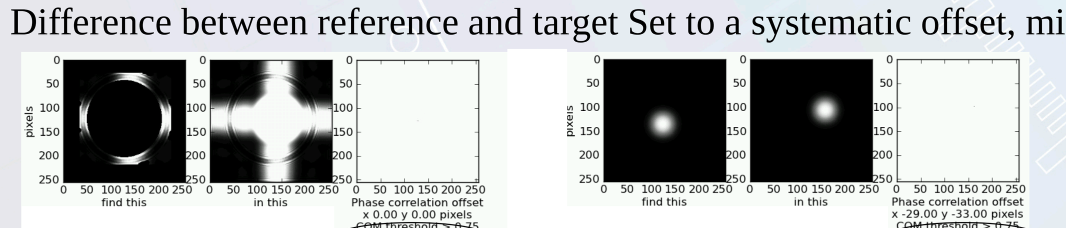
Would like minimal change with 2x intensity changes



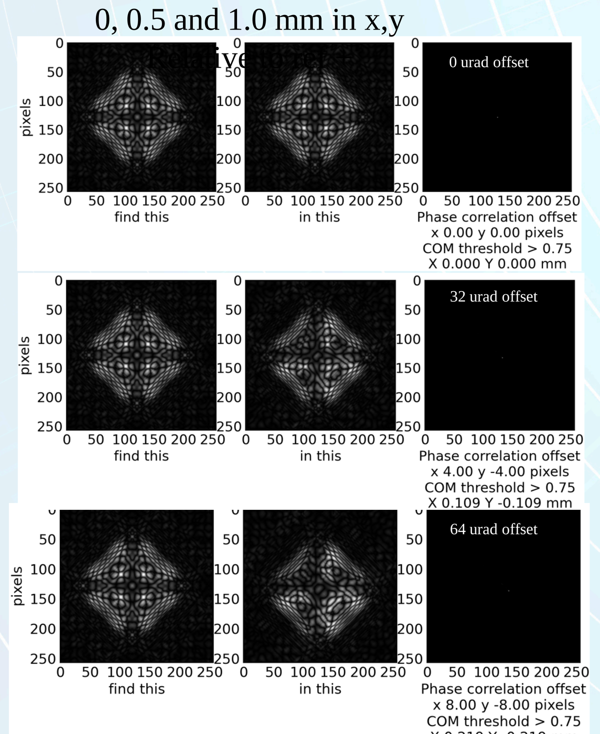
Detection of near reference movement



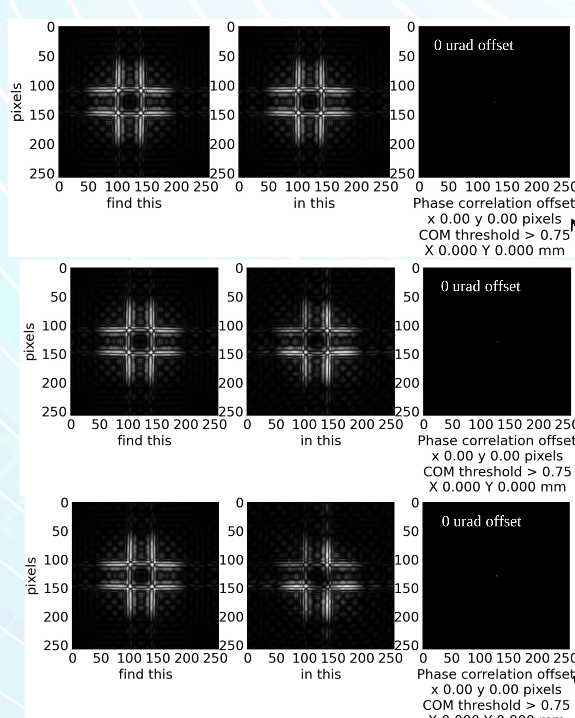
Difference between reference and target Set to a systematic offset, might be zero.



Find Beam X when moved 0, 0.5 and 1.0 mm in x,y



Find [0,0] ref + location is unaffected when target pattern moved 0, 0.5 and 1.0 both x,y

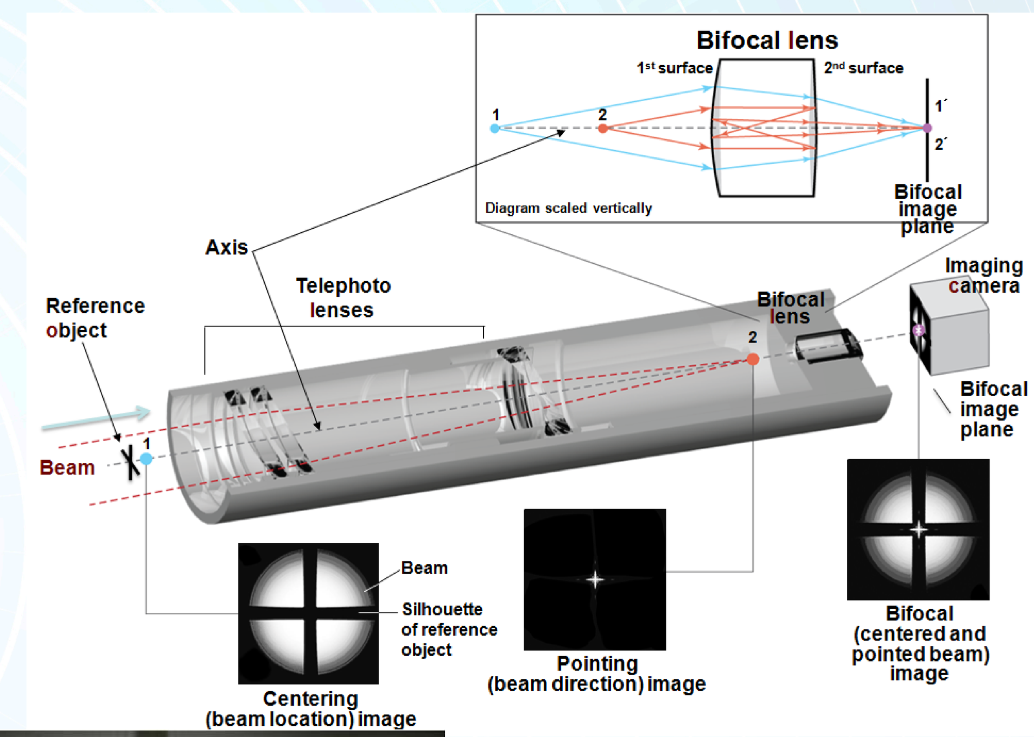


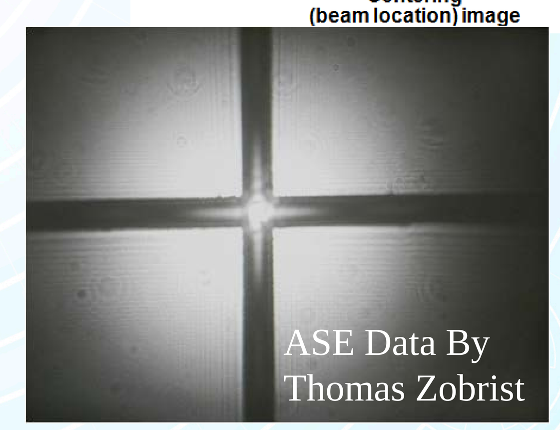
Check:
No change
in position

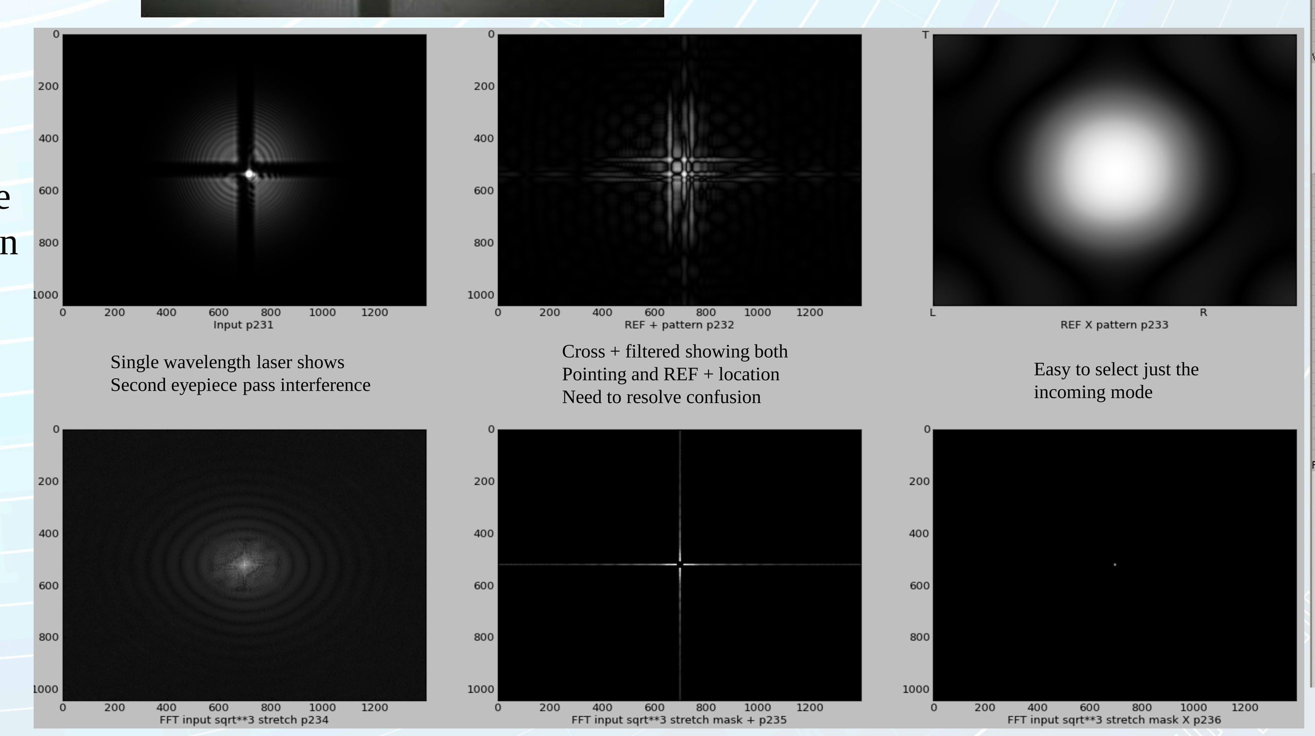
Methods and Future Work:

To distinguish between the near and far crosshair take advantage of specific diffraction patterns relevant to each object within the image. A FFT based phase correlation allows recognition and tracking of these patterns.

Test this Phase Correlations on
CAPS (Centering & Pointing System) principles of operation







Develop adaptable memory that is Training for changes.

Handle slight angular orientation changes as camera remounted, backup methods to help with learning.

Flat fielding to overcome sensor non uniformity.

Pixels go bad from nuclear radiation induced changes.

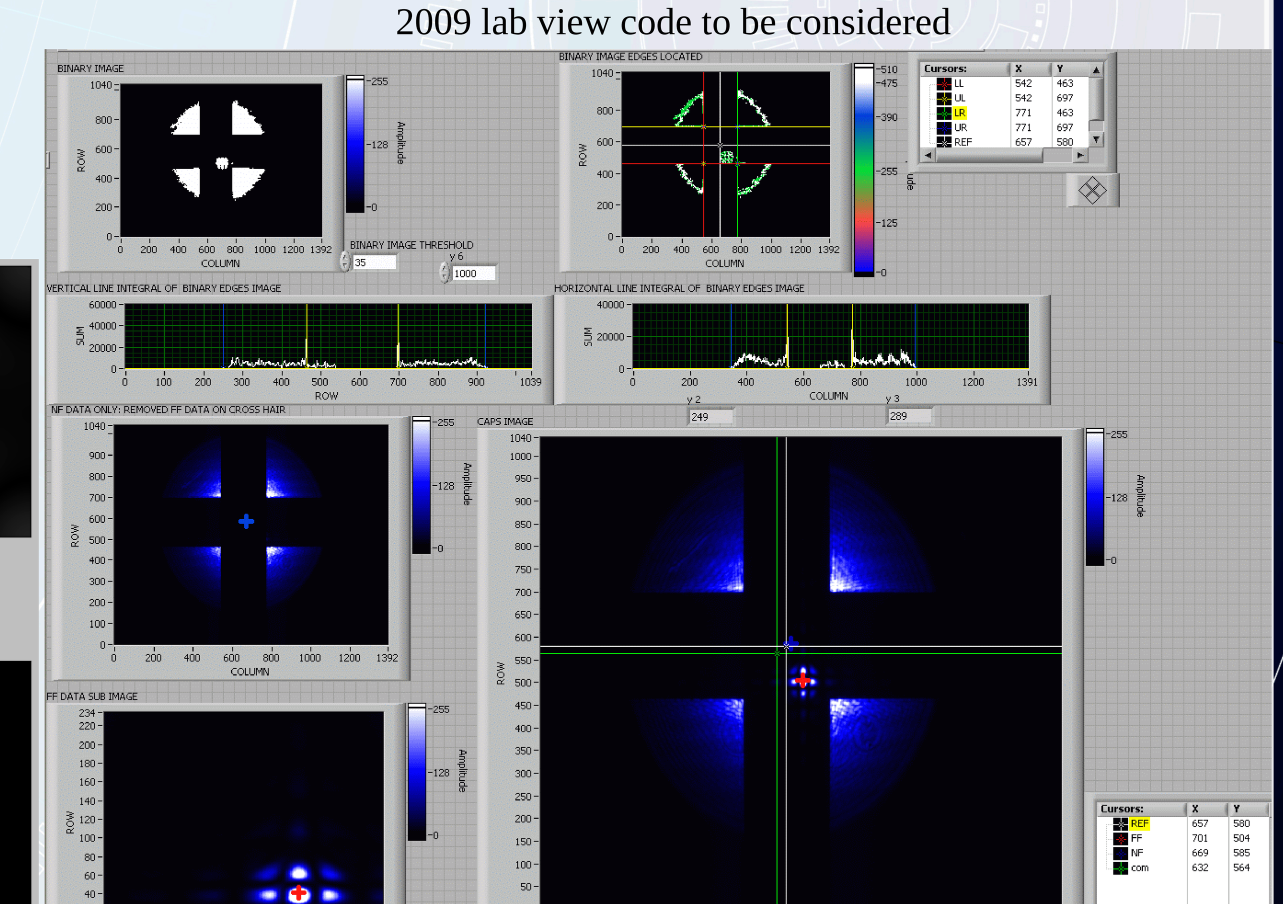
How to hand off PYTHON effort for use and testing by ICCS?

Can a web based client server machine be tasked with this methodology?

Can Python run on Android phones and their camera and network lower costs?

Develop GUI "ness" so to feel good to users

2009 lab view code to be considered



National Ignition Facility • Lawrence Livermore National Laboratory • Operated by the US Department of Energy
This work performed under the auspices of the U.S. Department of Energy by Lawrence Livermore National Laboratory under Contract DE-AC52-07NA27344.

IM Release Number: LLNL-POST-557857