**FGS Commissioning Tools: User Guide**

March 2018

POC: Keira Brooks (kbrooks@stsci.edu) or Lauren Chambers (lchambers@stsci.edu)

*Special notes for users:*

* Note that the IPython boxes are not necessarily meant to be copied word-for-word; users will need to change parameter values or file paths from what is documented here.
* However, also note that IPython string variable assignments that contain ‘…{}…’.format() use string formatting to auto-populate a file or directory name with a variable value. Please do copy *these* word-for-word, and make your life easier!
* If something is not working as is stated in these notes, please submit an issue on <https://grit.stsci.edu/wfsc/tools> (accessible from WFSC Guiding console – make sure you log in as yourself using your AD username and password) using the “Issues” button on the left-hand side of the page. Please specify that it is for this tool and be specific about the problem that you ran into.
* If the issue you are having is preventing you from completing your task, immediately contact Keira Brooks ([kbrooks@stsci.edu](mailto:kbrooks@stsci.edu), x6821) or Lauren Chambers ([lchambers@stsci.edu](mailto:lchambers@stsci.edu), x6517).

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

# Table of Contents

1. [Setting Up](#settingup)
2. [Getting the Input NIRCam Image](#gettingimage)
3. [Selecting Guide & Reference Stars for an Input NIRCam Image](#selectingstars)
4. [Opening DHAS](#openingdhas)
5. [Testing Selections in DHAS](#testingindhas)
6. [Contingency: Re-selecting Stas and Re-running DHAS](#reselectingstars)
7. [Writing the Segment Guiding Override File](#segmentguiding)
8. [Getting the Commissioning Tools on your Machine](#installingtherepo)
9. **Setting Up**
10. Open a Terminal
11. Type the following into the command line:

$ bash

1. Check to make sure that you have pysiaf and jwst-fgs-commissioning-tools installed

$ conda list | grep pysiaf

$ conda list | grep jwst-fgs-commissioning-tools

Note: conda search will not work here so you must grep the list of packages

1. If this search comes up empty for either package navigate to /Users/svc\_wssops/WFSC\_guiding/jwst\_siaf\_prototype/pysiaf or /Users/svc\_wssops/WFSC\_guiding/tools/fsg-commissioning (depending on what is missing) and type:

$ pip install –e .

1. Now make sure that you have the most up-to-date version of the tools

$ cd /Users/svc\_wssops/WFSC\_guiding/tools

$ git status

This command will let you know if there are any changes that you have made to the local master branch. If there are any untracked changes be sure to commit them or discard the changes.

$ git pull

1. **Getting the Input NIRCam Image**
2. To get the NIRCam image navigate to where you want the files saved (make sure you have already typed bash into the command line:

$ cd /Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ote{#}

1. $ scp\_cal\_sci “\*”

This copies all of the images from the DAN server to the current folder

1. **Selecting Guide & Reference Stars for an Input NIRCam Image**
2. Start IPython

$ ipython

1. Import the tool

In [1]: from jwst\_fgs\_commissioning\_tools import

run\_fgs\_commissioning\_tool

1. Define input parameters
   1. Required:
      1. **input\_image** - filepath for the input (NIRCam or FGS) image
      2. **guider** - number for guider 1 or guider 2
   2. Optional:
      1. **out\_dir** – if specified, determines where the out/ directory will be placed. If not, files will be saved to the directory in the main package directory (tools/fgs-commissioning/out/)
      2. **root -** if defined, will be used to create the output directory, out/{root}. If not provided, the root name will be derived from the input\_image filename.
      3. **nircam** (default True) - denotes if the input\_image is an FGS or NIRCam image. If True, the image will be converted to FGS format using the nircam\_to\_fgs.convert\_im function. The FGS-formatted image will be saved to out/{root}/FGS\_imgs/{input\_image}\_G{guider}.fits
      4. **global\_alignment** (default False)- set to True if the provided image is from global alignment, or the PSF-finding algorithm will go haywire.
      5. **nircam\_det** - is used to specify the detector of a provided NIRCam image. If left blank, the detector will be extracted from the header of the NIRCam FITS file.
      6. **fgs\_counts** and **jmag** - used to normalize the input NIRCam image, either to a desired J magnitude or to a desired number of FGS counts. The jmag parameter can also be used to normalize an FGS image.
      7. **in\_file** - a table of X/Y pixel coordinates and count rates that lists the desired guide star and reference stars; provided in the form of a filepath to a regfile.txt or .incat file. Providing this will bypass using the Star Selection GUI to choose the guide and reference stars.
      8. **bkgd\_stars** (default False)- a developing feature that includes a number of background stars in the final image

Example (*actual parameters might change depending on the data*):

Note: “ote\_number” is an integer stemming from the commissioning step numbers (e.g. OTE-07), added here for file naming reasons; it is not a parameter that gets passed to the tools

In [2]: ote\_number = #

In [3]: guider = 1

In [4]: input\_image = ‘/Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ ote{}/NIRCam\_filename.fits’.format(ote\_number)

In [5]: root = ‘GA\_obs1\_WFR’

In [6]: global\_alignment = True

In [7]: out\_dir = ‘/Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ ote{}/’.format(ote\_number)

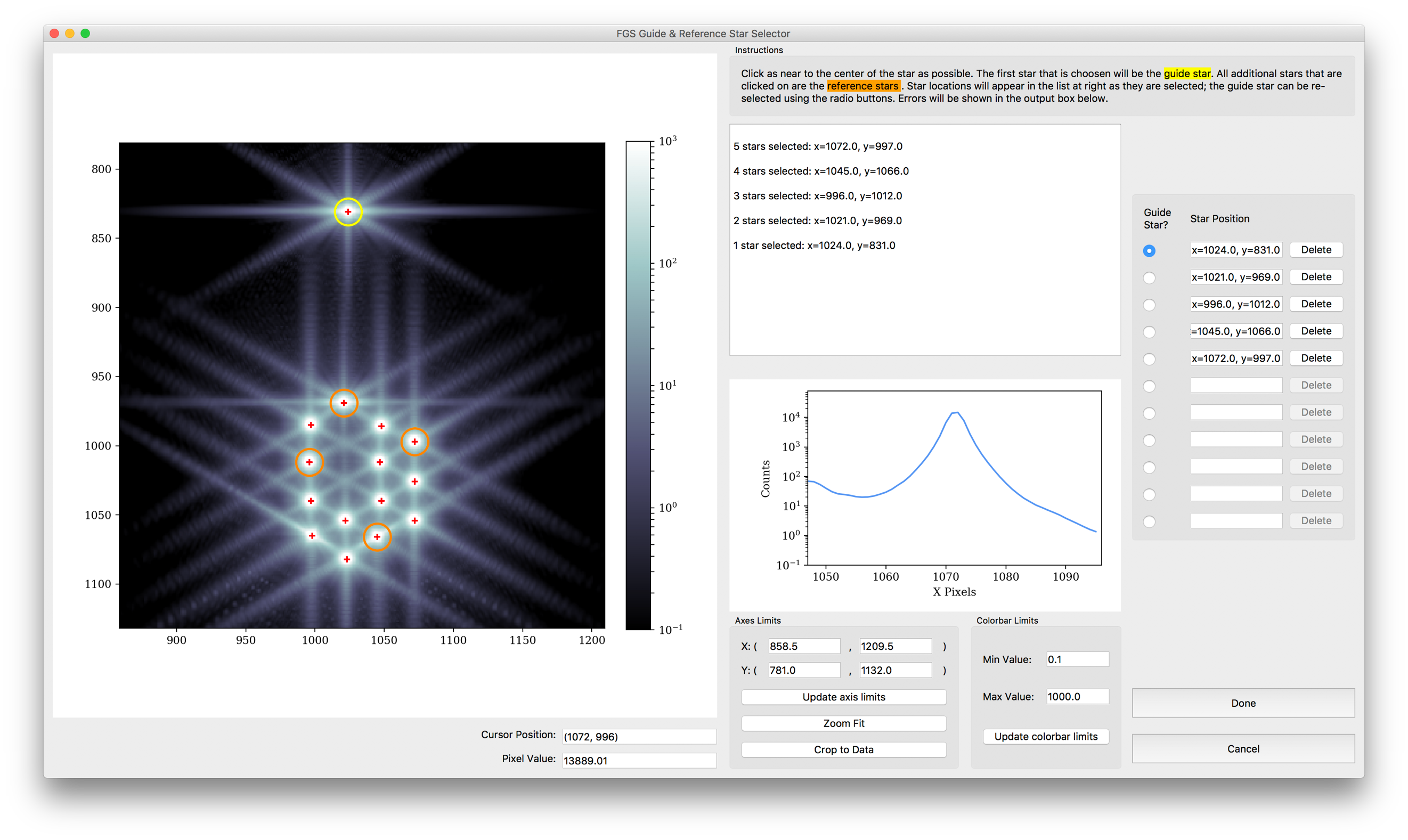
**Fun tip**: You can drag and drop a file or directory into the terminal and it will paste the path name, just be sure to put quotes around it and delete the following space.

1. Run the tool with the defined parameters

In [8]: run\_fgs\_commissioning\_tool.run\_all(input\_image, guider,

root=root, global\_alignment=global\_alignment, nircam=nircam, out\_dir=out\_dir)

1. After the tool is started, a log is written out to the command line (and to /Users/svc\_wssops/WFSC\_guiding/tools/fgs-commissioning/logs/) of what steps are underway. This is also where you will see if the tools are successful.
2. When the GUI appears:



* 1. Inspect the PSFs in the image by moving your cursor over different PSFs. Examine the profile plot to see the distribution of light.
  2. Select, by clicking, which PSFs will be the guide star and the reference stars. The first star selected will be the guide star, while any subsequent stars will be reference stars.

*For GA*: Choose two (2) reference stars

*For Coarse Phase*: Choose the unphased stacked PSFs as the guide star. No reference stars

* + 1. Select the more compact or “peaky” PSFs rather than diffuse, noncontiguous or “blobby” PSFs
    2. Try to select some reference stars that are close to the guide star and some that are further away
    3. For global alignment, try to select the guide and reference stars from the outside ring of PSFs
    4. Note: Do not choose reference or guide stars in regions where several PSFs are close together
  1. If you want to change your selections while in the tool, use the “Guide Star?” radio buttons to change the guide star, or use the “Delete” buttons to remove selections
  2. When you are happy with your selections, click “Done”

1. If you get the spinning-rainbow-ball-of-death, find the icon below on your dock, right-click, and click “Force Quit”. Then start again from step 3.



1. **Opening DHAS**
2. Find the MATLAB icon on the dock and double click to open MATLAB (or press cmd-space bar to open Spotlight and search for MATLAB)



1. In the MATLAB interface, if not already there, navigate to the DHAS, version 3.0 directory

$ cd /Users/svc\_wssops/Documents/MATLAB/dhas/v3.0/fitsdisp2\_v3p0/

1. To start the DHAS GUI type the following and then press return:

$ fitsdisp2

1. **Testing Selections in DHAS**



Figure 1: DHAS GUI

1. Load the IDstrips.fits file you just created:
   1. Click the blue “Load Multiple .FITS Files” button



* 1. Using the “Current Directory” drop-down menu (or by typing the path directly into the textbox), navigate to the out/{root} directory where the images have been saved. (In the above example, /Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ ote{#}/out/GA\_obs1\_WFR/)
  2. Go the “dhas” subdirectory (In the above example, /Users/svc\_wssops /WFSC\_guiding/WFRMarch2018/ote{#}/out/GA\_obs1\_WFR/dhas/)
  3. Check the “Show All Files” box
  4. Select the IDstrips.FITS file (In the above example, GA\_obs1\_WFR\_G1\_IDstrips.fits)
  5. Click “Add →”
  6. Click “Done”

1. Run the ID simulator
   1. Click the corresponding G1 or G2 button depending on if this is a guider 1 vs. guider 2 image at the top of the page



* 1. Click the small pink “ID-sim” button at the top of the page



* 1. A finder window will appear with the expectation that the user will choose the appropriate .prc file (no text indicates this). For this step (ID) open the ID.prc file corresponding to the IDstrips.fits file that you selected (in the above example, GA\_obs1\_WFR\_G1\_ID.prc)
  2. When the star\_catalog\_page dialog box appears:
     1. Set the Row and Column values to be 12 and 0 (in the bottom left corner of the window), respectively (the X-Angle and Y-Angle values will automatically change to reflect this):



* + 1. If you want to alter your guide & reference star selections, do so here by toggling the “GS” (guide star) and “RS” (reference star) buttons
    2. Click the “Done” button.
  1. When the ID Mode Setup dialog box appears, click “OK.”

1. Wait for the simulator to run
2. When the finder window pops up, press “Cancel”
3. DHAS results note: Commanded (CMD) reference stars are denoted by yellow triangles (∆). The commanded (CMD) guide star is denoted by a yellow cross/plus sign (+). The reference stars that the DHAS has found are denoted by blue x’s (x) and the guide star is denoted by a blue asterisk (\*). See the screen shot below for an example of a successful DHAS run.



Figure 2: Example of a successful DHAS run

1. Inspect the DHAS results
   1. Do the stars that DHAS found (in blue) match the stars you commanded it to find (in yellow) If not, DHAS has **failed**.
   2. For more detailed DHAS diagnosis:
      1. Does DHAS think it successfully found the guide star? (Does “Status” equal SUCCESS?)
      2. Did DHAS find all of the stars? (Does “# Candidates” equal 18 – for GA or image array steps?)
      3. If necessary, click the “Export” button to more closely examine DHAS’s output plot and/or save the image as a .PNG
2. If all goes well, click the “EXIT” button in the DHAS GUI to exit the GUI and then type “exit” into the Command Window to close MATLAB

**If DHAS fails, we need to try a different orientation of guide and reference stars until we find a successful one. See Part VI.**

If DHAS is successful, move on to Part VII.

1. **Contingency: Re-selecting stars and Re-running DHAS**

When re-selecting stars for testing in the DHAS, not all of the files generated by the commissioning tools need to be rewritten; only the files that specify which segments are the guide star and reference stars need to be changed (namely, the ID.prc and regfile.txt files). Thus, though completely re-running the tool would work to re-select the guide and reference star segments (see Part III), it is not necessary. We have developed a faster method for re-selection and file rewriting, detailed here.

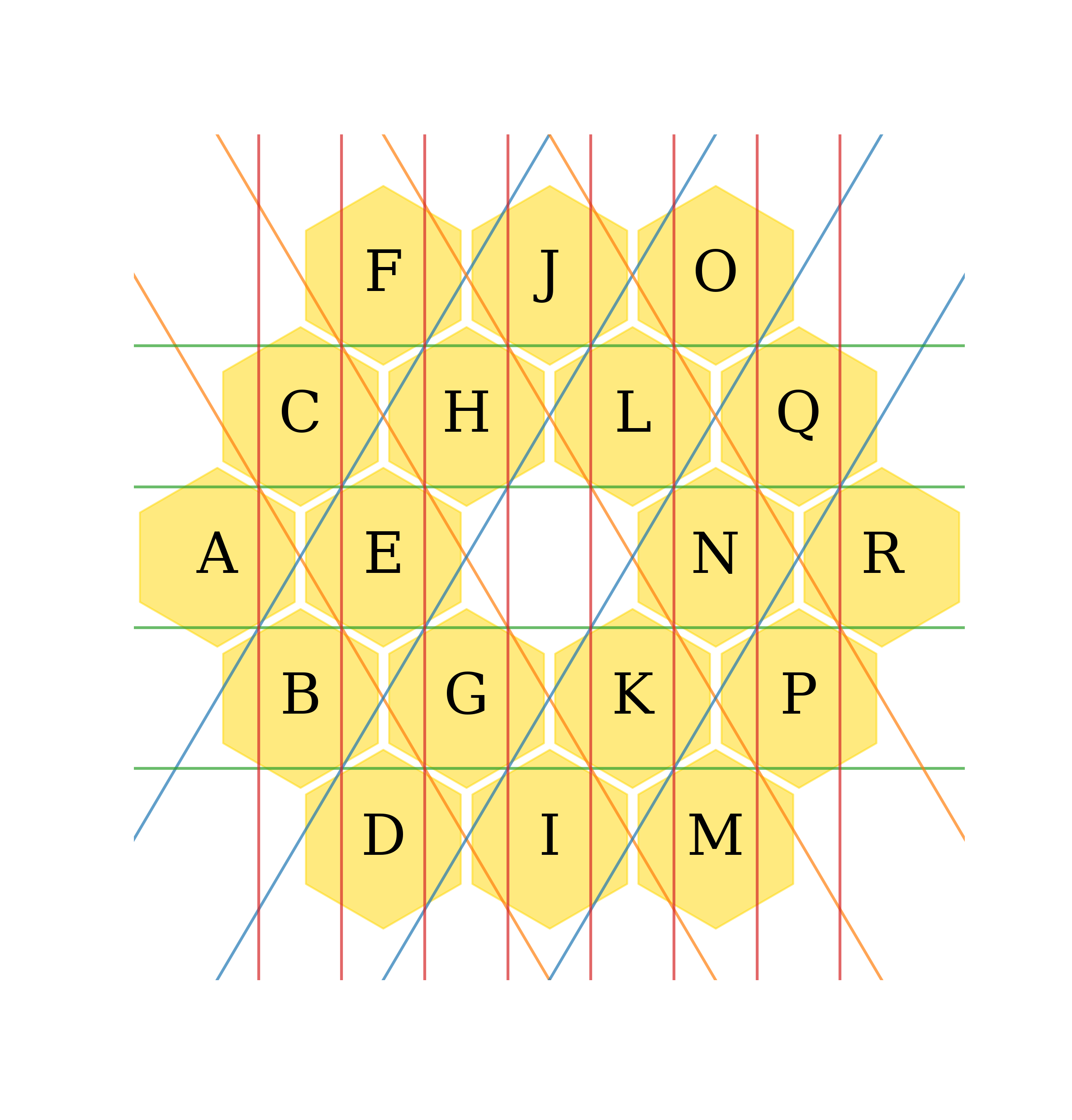
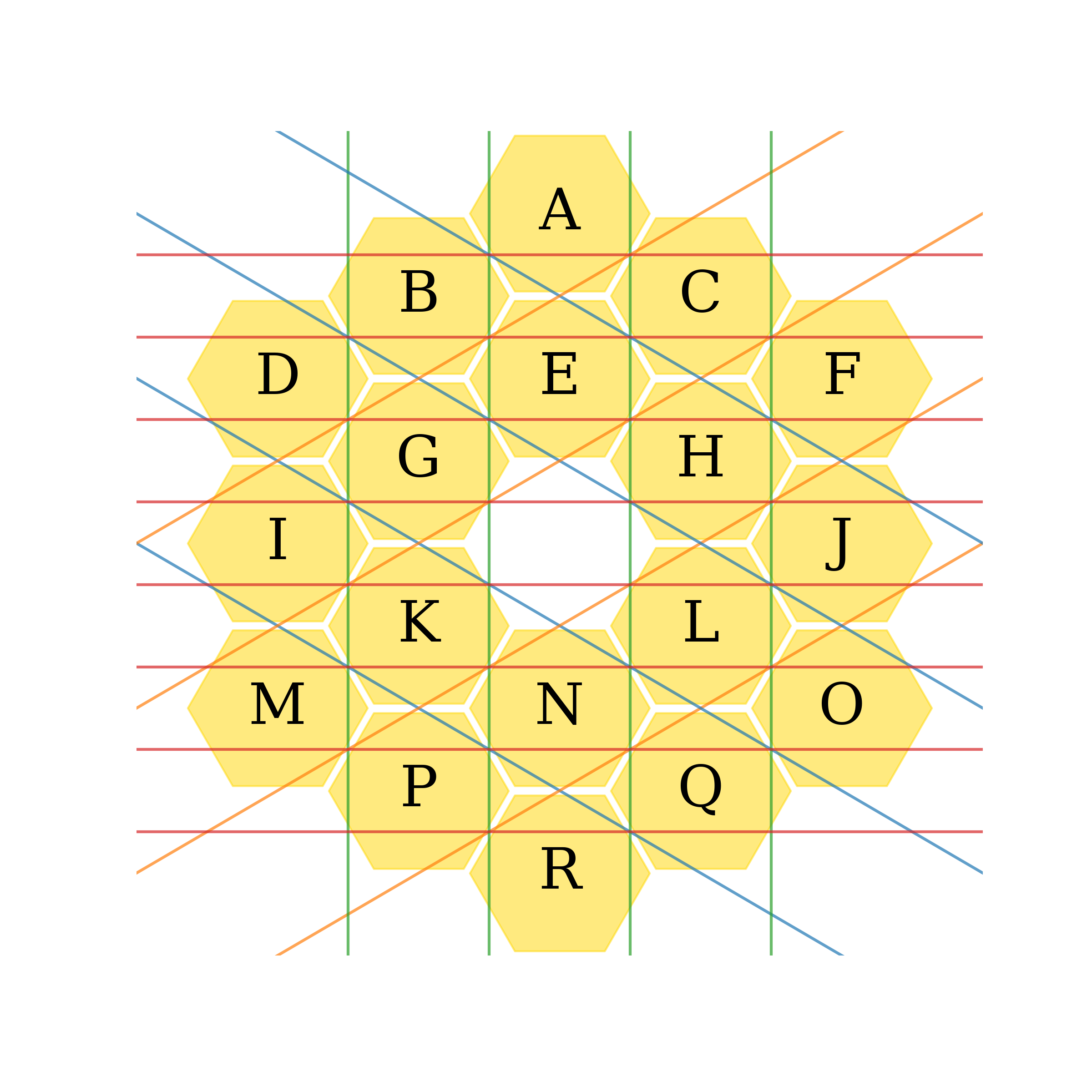
1. Examine your input image and decide what new guide/reference stars you will select. If you do this in DS9 (which you can do by typing $ ds9 name\_of\_image.fits into the command line or !ds9 name\_of\_image.fits from IPython) be sure to load the image in the folder /Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ote{#}/out/{root}/FGS\_imgs/, which is your FGS image. Since DS9 displays images differently than Python, go to the Zoom menu in the menu bar of DS9 and select “Invert Y”. Recall the suggested selection criteria:

*For GA*: Choose two (2) reference stars

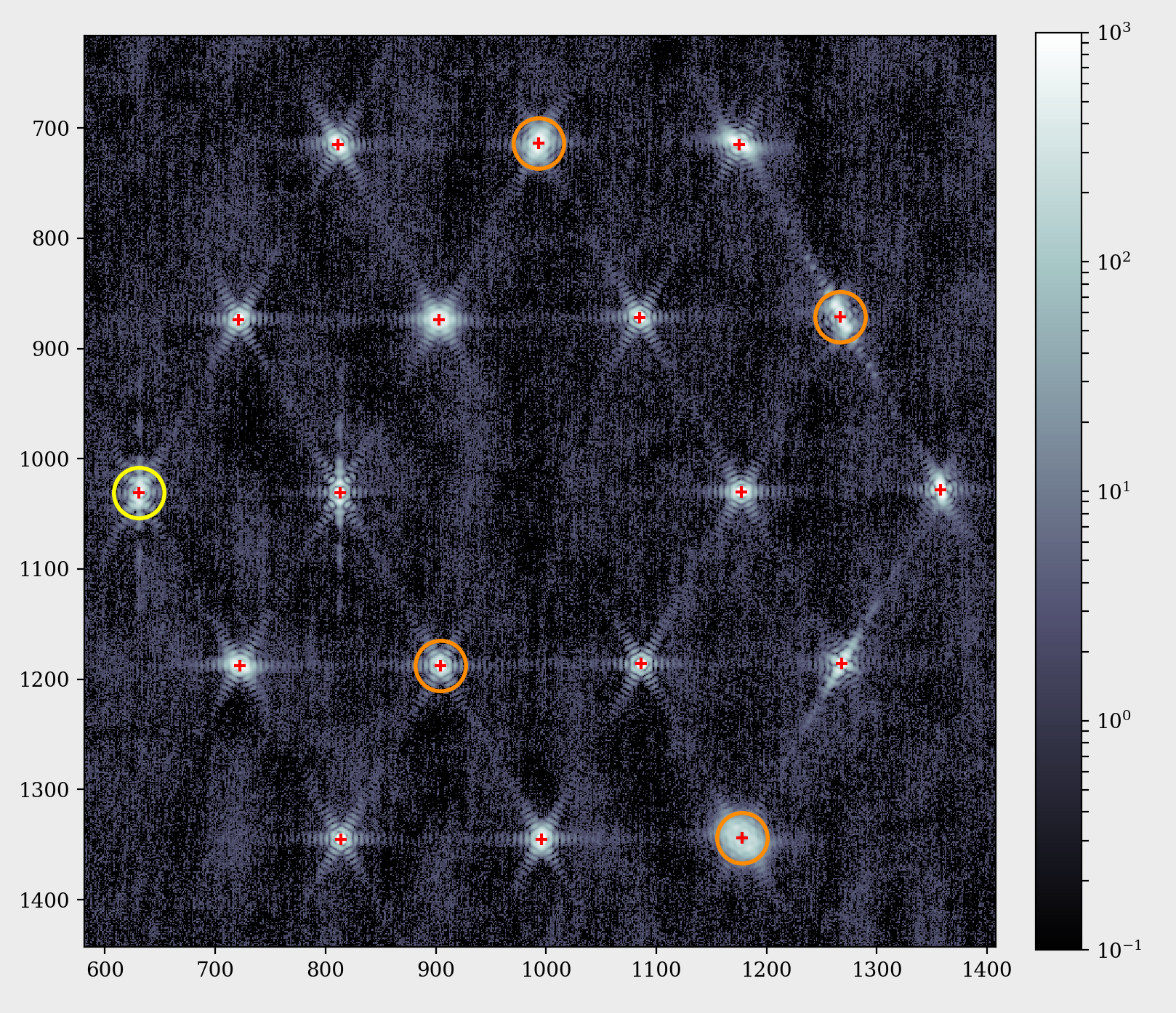
*For Coarse Phase*: Choose the unphased stacked PSFs as the guide star. No reference stars

* + 1. Select the more compact or “peaky” PSFs rather than diffuse, noncontiguous or “blobby” PSFs
    2. Try to select some reference stars that are close to the guide star and some that are further away
    3. For global alignment, try to select the guide and reference stars from the outside ring of PSFs
    4. Note: Do not choose reference or guide stars in regions where several PSFs are close together

1. Look at the following labeling schema (whichever orientation matches your data) and identify the letters corresponding to your desired segments:



Example: “AQGMJ” would correspond to segment A being the guide star, with segments Q, G, M, and J being four reference stars:



1. Import the rewrite\_prc function

In [10]: from jwst\_fgs\_commissioning\_tools.fsw\_file\_writer

import rewrite\_prc

1. Rewrite the ID.prc and regfile.txt files (here the out\_dir argument should provide the path/to/out\_dir/out/root/) with the desired star selection command.

Example (*actual parameters might change depending on the data*):

In [11]: command = ‘AQGMJ’

In [12]: out\_dir = ‘/Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ote{}/out/{}/’.format(ote\_number, root)

In [13]: rewrite\_prc.rewrite\_prc(command, guider, root,

out\_dir)

Unlike the run\_all function, you must specific the guider, root, and out\_dir. Guider and root will be what you used to run the previous steps, but out\_dir must be specified, ending in the root of your image.

This step will overwrite the existing ID.prc and regfile.txt files with the new star commands.

1. Re-run DHAS (Part V)

**If DHAS fails again, we still need to try a different orientation of guide and reference stars until we find a successful one. Repeat Part VI.**

If DHAS is successful, move on to Part VII.

1. **Writing the Segment Guiding Override File**
2. Import the segment guiding tool (you might get a warning here about the backend, you can ignore this warning)

In [14]: from jwst\_fgs\_commissioning\_tools.segment\_guiding

import segment\_guiding

1. Define the paths to the segment lists that were written after star selection in Part III, or re-written with rewrite\_prc in Part VI: ALLpsfs.txt and regfile.txt (In the above example, found in ~/WFSC/tools/fgs-commissioning/out/GA\_obs1\_WFR/)

In [15]: out\_file\_dir = ‘/Users/svc\_wssops/WFSC\_guiding/WFRMarch2018/ote{}/out/{}’.format(ote\_number, root)

In [16]: segment\_infile = ‘{}/{}\_G{}\_ALLpsfs.txt’.format(out\_file\_dir, root, guider)

In [17]: selected\_segs = ‘{}/{}\_G{}\_regfile.txt’.format(out\_file\_dir, root, guider)

In [18]: program\_id = {Program ID} \*\*See note below\*\*

In [19]: observation\_num = {Observation Number}

In [20]: visit\_num = {Visit Number}

**Note:** You may need to ask the Wavefront Ops for the program ID, observation number, and the visit number. *The program ID number need to be a string of the format ‘00###’*

1. Define the guide star parameters, including:
   * **fgsNum** – guider number (parsed from VSS GS report if provided)
   * **RA** – RA of the guide star (parsed from VSS GS report if provided)
   * **Dec** – Dec of the guide star (parsed from VSS GS report if provided)
   * **PA** – position angle of the guide star (parsed from VSS GS report if provided)
   * **V2Boff** – V2 boresight offset
   * **V3Boff** – V3 boresight offset
   * **segNum** – the segment number on which will be used to guide, i.e. which segment will be placed at the V2/V3 origin. If this is set to 0, then the average position of the segment array will be used as the center/origin. Should be 0 for all cases where we are guiding on a usual image array (e.g. global alignment); might need to be changed for cases where we are guiding on a clump of segments
   1. If we have been provided a guide star report (GS\_report.txt) from VSS, point the tool to this file

In [18]: vss\_infile = ‘path/to/GS\_report.txt’

* 1. If we have **not** received a VSS file, get the following parameters from the following places:
     1. Guide star **RA** & **Dec**: Retrieve the guide star ID (e.g. N13I000017) from the APT file, under “Special Requirements” for the current observation. Look up this ID in the guide star catalog to determine the RA and Dec (<http://gsss.stsci.edu/webservices/GSC2/WebForm.aspx>; submit the guide star ID in the “HST ID” field).
     2. **PA**: Ask the S&OC for the visit position angle.
  2. Get the current **boresight offset** (V2Boff and V3Boff) from the wavefront scientist.
  3. Provide a dictionary containing guide star parameters

Example (*actual parameters will change depending on the guide star, OTE step, and telemetry*):

In [19]: GS\_params\_dict = {'V2Boff': 0.,

'V3Boff': 0.,

'fgsNum': 1,

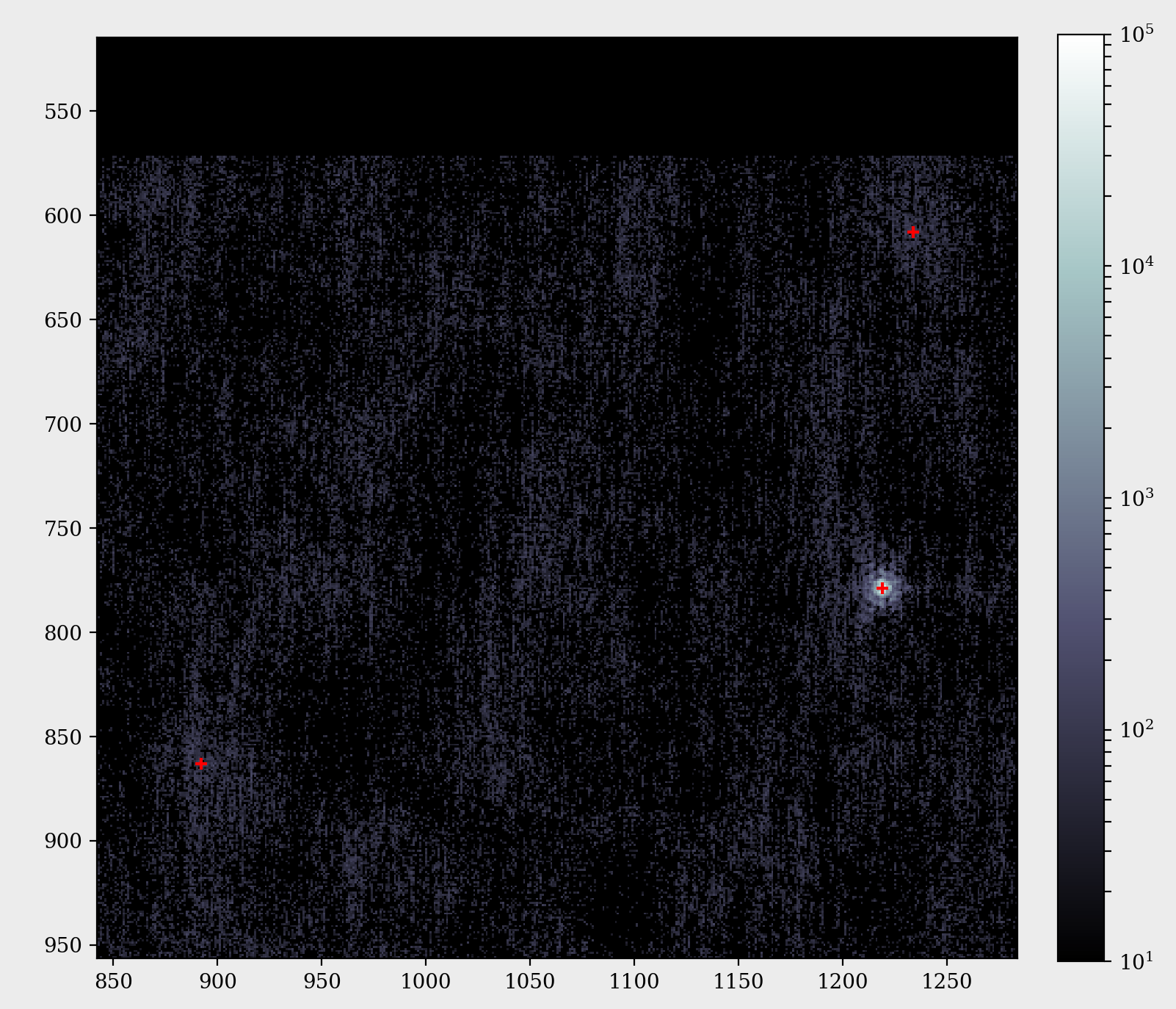
'RA': 273.18584587,

'Dec': 65.53650820,

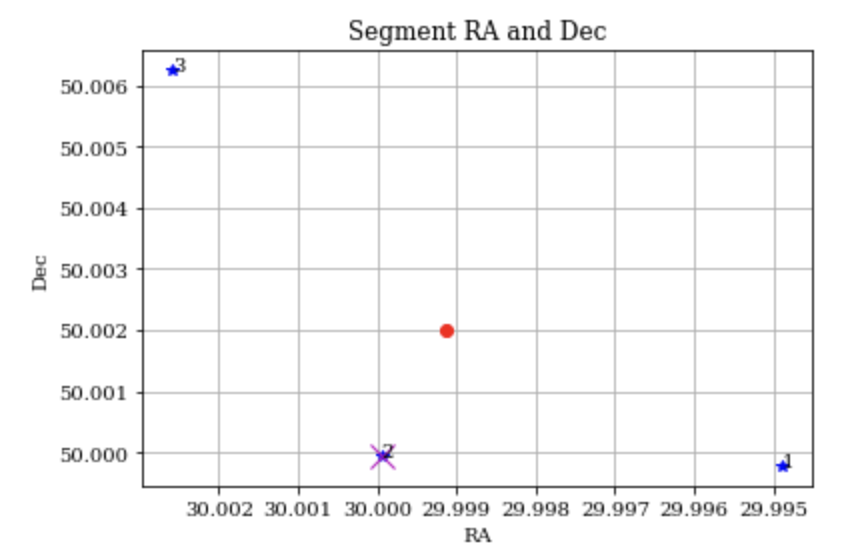
'PA': 13.25759450,

'segNum': 0}

NOTE FOR WFR: When testing coarse phasing data (OTE-12) that includes just one diffuse PSF, be sure to set the segNum parameter to match the segment number of the desired PSF, rather than segNum = 0. To determine the correct PSF number, you can run the segment guiding tool once (steps 1-4), observe the output plots (.png files) in the out/{root} directory ({root}\_RADecsegments.png and {root}\_V2V3segments.png) to determine the appropriate number, and run the segment guiding tool again (steps 1-5) to write out the files correctly.



Example: in this coarse phasing data, the star selector has incorrectly identified two non-sources, or “noise islands,” as sources. We don’t want to write these out to the segment guiding override file, and we don’t want the centroid of all 3 ‘sources’ to be where the telescope points.



If segNum is commanded to 0, the pointing will center on the red dot ⚫. Instead, set the segNum to 2, so the pointing centers on the real segment, shown with a purple **X**.

1. Run the tool

In [20]: segment\_guiding.run\_tool(segment\_infile, program\_id,

observation\_num, visit\_num, selected\_segs=selected\_segs, GS\_params\_dict=GS\_params\_dict, vss\_infile=vss\_infile, out\_dir=out\_file\_dir)

1. Copy the output file to central storage directory where it can be accessed by PPS:

$ cp path/to/gs-override.txt /data/jwst/wss/gs\_selection\_override/

Ensure that all people have read permissions!

1. **Getting the Commissioning Tools on your Machine**

**The repository is currently installed on the service SOGs account so should not need this section if you are using that account and the environment has been properly set up. If you are using your own machine, however, you will need to follow these steps. Note: These will not be the same steps for installing the tools on your SOGs account.**

1. Go to <https://grit.stsci.edu/wfsc/tools>
2. In a terminal, in the directory where you want keep this repository:

$ git clone [git@grit.stsci.edu:wfsc/tools.git](mailto:git@grit.stsci.edu:wfsc/tools.git)

1. Note: from here on out you can follow the instructions for installing the repo in the README or you can continue with these instructions.
2. For the tools repository you also will need the pysiaf repository, so go to <https://grit.stsci.edu/ins-tel/jwst_siaf_prototype>
3. In the terminal, in the directory where you want keep this repository (I recommend using the same directory as where you put the tools repository) :

$ git clone [git@grit.stsci.edu:ins-tel/jwst\_siaf\_prototype.git](mailto:git@grit.stsci.edu:ins-tel/jwst_siaf_prototype.git)

$ cd jwst\_siaf\_prototype/pysiaf

$ pip install –e .

1. Note that the last instruction there is not what is recommended on the grit page for the pysiaf repository, but has caused fewer installation issues. If you run into a problem with liblxml2.2.dylib, got to <https://grit.stsci.edu/ins-tel/jwst_siaf_prototype> to the “KNOWN INSTALLATION ISSUE” section for how to solve this issue
2. Now return to the tools directory:

$ cd tools/fgs-commissioning

$ pip install –e .

1. You now should have the tool up and running!
2. Note that since this is a Grit repository that is still in development, whenever you go to use the tools, be sure to first do the following to get the latest version of the tool:

$ cd tools/fgs-commissioning

$ git pull origin master

1. You should be running this in an environment with Astroconda installed with Python 3+. For installing Astroconda see: <http://stsci-env.readthedocs.io/en/latest/installing_anaconda.html>
2. Additionally you will need to install photutils. If, for example, this environment is named “astroconda”:

$ source activate astroconda

$ conda install photutils