UCF Physics: AST 5765/4762: (Advanced) Astronomical Data Analysis Survey
Homework 9:
Due Wednesday Nov 5, 2:30pm.

This assignment continues the correction of the data in from the previous 2 homework assignments.

1. (**5 points**) Use your terminal to make a new folder hw9_yourname under your ast4762/ homework folder. For this homework your main homework file is a Python file named hw9_<username>.py (or a hw9_<username>.ipynb but remember to save it as a .py before submission). Save it in your homework folder. Remember to commit your files and push to GitHub. Start the file with the usual header. Copy the routines and homework files from the previous homework into the directory for this assignment (hw9_yourname). Correct any errors you may have had, making a comment that says "# FIXED: " and giving the date.

2. (**10 points**) Make a table that will hold your photometric results.

You will start by initializing a 3D data array as shown in the example below (not using np.zero()!). The axes of your array will be (star number, frame number, parameters).
Each row should contain:
(a) the approximate y location of the star,
(b) the approximate x location of the star,
(c) the width of the star,
(d) the fitted y location of the star,
(e) the fitted x location of the star,
(f) the star's flux level, and
(g) the average sky level

(see below).

You will start with a header comment for the parameters and use np.nan to indicate that you haven't calculated a given value yet. You may copy-paste this table to your HW file:

```
# Stellar photometry:
photometry = np.array( [ # yguess, xguess, width, cy, cx, star, sky
# star 0
[ [ np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan] ], # frame 2
# star 1
[ [ np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],         # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan], # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan] ] , # frame 2
# star 2
[ [ np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],          # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]]  # frame 2
], dtype=float)
```

In this problem you will do photometry on three stars in each of the first three observation frames (files `stars_13s_1`, `stars_13s_2` and `stars_13s_3`).

Start by filling in the position of the three target stars in the first frame as shown here:

```
# Stellar photometry:
photometry = np.array( [ # yguess, xguess, width, cy, cx, star, sky
# star 0
[ [ 698, 512, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan] ], # frame 2
# star 1
[ [ 668, 520, np.nan, np.nan, np.nan, np.nan, np.nan],         # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan], # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan] ] , # frame 2
# star 2
[ [ 568, 283, np.nan, np.nan, np.nan, np.nan, np.nan],          # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]]  # frame 2
], dtype=float)
```

If you study the images, you will notice that systematics like telescope motion between observing the different frames result in a slightly different position of your stars from one frame to another. Fill in the array with the position of star 0 in the other two frames. Your array will look like this (note numbers are of course wrong, they are just for display):

```
# Stellar photometry:
photometry = np.array( [ # yguess, xguess, width, cy, cx, star, sky
# star 0
```

```
[ [ 698, 512, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 0
  [999, 999, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [999, 999, np.nan, np.nan, np.nan, np.nan, np.nan] ], # frame 2
# star 1
[ [ 668, 520, np.nan, np.nan, np.nan, np.nan, np.nan],        # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan], # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan] ] , # frame 2
# star 2
[ [ 568, 283, np.nan, np.nan, np.nan, np.nan, np.nan],         # frame 0
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan],  # frame 1
  [np.nan, np.nan, np.nan, np.nan, np.nan, np.nan, np.nan]]  # frame 2
], dtype=float)
```

Calculate the offset for star 0 between frames 0 (stars_13s_1) and 1 (stars_13s_2) and frames 0 (stars_13s_1) and 2 (stars_13s_3).

Add the appropriate offset to the positions of stars 1 and 2 in frame 0 to get their positions in frames 1 and 2, and update the table. You can use a loop and find the numbers of stars and frames by querying the table.

Note: For small datasets like this one, this offset procedure seems like extra work since guessing the positions of a few stars isn't hard. For big datasets, this is the only way to go, and "big" here is only a few dozen stars. For big datasets, you would allocate the table with np.zeros(), fill with np.nan, and set the few known values in assignments rather than quoting the whole table as done here.

3. (**20 points total**)

a. (**10 points**) Use the N-dimensional Gaussian fitting function `gaussian.py` from our class demos to estimate the 1σ widths and location of each star from your table. Record the average of the two width values for each star in the table, along with the fitted y and x positions.

To avoid getting `np.nan`, other stars, and bad pixels, make a copy of a subarray around each star, subtract the subarray median value to get an approximate zero background, and do your fit in that image. Note that you should do this in a copy of the subimage, rather than the original slice of the full image (why?).

Remember to adjust your initial guess `y` and `x` values to be relative to the subarray, and then to adjust the positions you get from the subarray fits back to the coordinates of the full array. Fill these last positions in your table from problem 2.

Print the table and copy-and-paste it to a comment in your homework file. Check whether your final positions are consistent with your guesses.

b. (**10 points**) Take the average of all widths to get a single width that you will apply to the whole dataset.
Set variables for the photometry aperture radius, two sky annulus radii, and subimage size as multiples of your fitted width. The sub-image size is the size of the sub-image that will be copied from each image for photometry. Justify your choices of multipliers for each new variable in a few comment sentences in your main homework file.

4. (**5 points**) Prepare and submit your homework. Remember to follow all necessary steps please. Don't forget to commit and push to GitHub. At all points, explain what you did in your log. Make a screenshot that shows your logging history, showing that you committed the file and add it to your `hw9_<yourname>` folder.