AST 381: Planetary Astrophysics

Homework #2, Due Oct 1

Reminder: Show your work! Each part is worth equal weight, and the assignment is worth 100 points in all. **Solutions should be typeset with LaTeX. Hand-written solutions will not be accepted. All code should be posted on github, in such a way that I can access it, and the link should be given in the TeX.**

The goal of this assignment is to build the basic infrastructure that one could use in analyzing coronagraphic AO imaging data from a typical ground-based telescope. We won't get to the point of implementing LOCI, PCA, or other such algorithms, but I suggest that implementing them (and maybe even testing them against each other) could be an interesting term project. Note that I'm open to people maybe using HR 8799 for this assignment instead. The 2009 K' dataset is supposed to be particularly good.

Part 1: Go to the Keck website and search for observations of ROXs 42B and ROXs 12 with the NIRC2 camera (the facility adaptive optics imager). You specifically want the images from 20110623 taken with the corona300 coronagraph. Download the calibrated images, of which there should be about 30 for each target. These are standard "angular differential imaging" datasets, where the telescope is fixed so that the zenith is up (or at least at some constant angle), while the sky rotates in the field of view (generally rotating around whichever star is the AO guide star).

Part 2: Write a code in your language of choice that, for each image, will measure the x/y pixel position of the star underneath the coronagraph, and write the file name and x/y coordinates to a text file. Those x/y coordinates will be useful in subsequent steps when "registering" images.

I suggest that the best way to do this is either to place a box down and measure the flux-weighted centroid, or fit a 2D Gaussian (where you can probably fix the width and just fit for x/y) to it. Either way, be careful of the area where flux leaks in around the edges of the coronagraph - you don't want this influencing your position measurement too much. Note that you can simplify life considerably by using the fact that the coronagraph itself is always within a pixel or so of the same position, and hence if the observer did a good job of centering the star underneath the coronagraph, then you can start with an initial guess of the position that's pretty good.

Part 3: Write a code that, for all images of a star, will "register" them (shift them all so that the star is at the same x/y position in them) and then produce output images with the sum and the median of each stack. It's fine to call an outside routine to do the actual shift, such as the IDL routine fshift. Note that since the sky is rotating, this has the effect of creating an average representation of the point spread function while simultaneously blurring any companions in the tangential direction. Produce sum and median images for both stars.

Part 4: Write a modified version of the last code that, while stacking the images, will rotate each individual image around the position of the primary star so that north is up. It's fine to use outside code for doing the rotation, such as the IDL routine rot. This has the effect of causing the flux of real companions to add up, while the stellar PSF blurs tangentially. Again, produce sum and median images for both stars. Note that you can get the PA of the +y axis from the FITS headers, using a combination of parameters: PA = PARANG + ROTPPOSN - EL - INSTANGL

Part 5: Write a code that, for each image, calculates the brightness profile as a function of distance away from the star (that is, calculates the azimuthal median in concentric rings) and subtracts it off of that image. If you want to game through the best way to do this, stop by and we can chat. Do that for each individual image, and then use your code from steps 3 and 4 to produce corresponding stacked sum and median images.

Part 6: Write a code that, for each image, subtracts off the median-combined image of the PSF (from part 3). This is the classical definition of "angular differential imaging". After producing these ADI-subtracted images, register and stack them (via sum and median) as in Part 4.

One way to do this might be to register the science frame against the calibrator (median-PSF) frame, find the median ratio between the pixel values in some broad ring around the science target, use that to rescale the calibrator brightness to the science target brightness, and subtract it off.

Part 7: Write a code that, for each image of a given star, compares it to all the images of the other star and finds the one that has the most similar PSF, then subtracts that calibrator image off. (I suggest that you output the name of that best-matching image to a text file, to allow for ease of grading.) After producing these best-PSF-subtracted images, register and stack them (via sum and median) as in Part 4.

One way to do this might be to register the science frame against a potential calibrator frame, find the median ratio between the pixel values in some broad ring around the science target, use that to rescale the calibrator brightness to the science target brightness, then measure the χ^2 of the residuals. The best-fit image should be the one with the lowest χ^2.

Part 8: For each median image that you have produced in parts 4 through 7, pick out what appear to be the real objects and measure their positions in the same way as you did for the primary star. Using the pixel scale for NIRC2, which you can look up online or in the literature in various ways, measure the position angle and projected separation for each object. (For now, don't worry about distortion.)