STEP 4 - MAKING A LIST OF IMAGE SHIFTS

In order to "un-dither" the images so that they can be added "pixel by pixel" into a final image, we need a list of the shifts.

We will use a package called 'xdimsum' for this step, which contains a command xdshifts.

epar xdshifts inlist = @on.list refim = on1 shiftlis = shifts.list cradius = 5

These settings will use all our "on frames" (from the list on.list) to calculate shifts. Since 'refim = on1', the first on frame is set as a reference (i.e. no shifts) and all other frames have shifts relative to this one. By setting 'cradius = 5' the reference star will be fitted (with sub-pixel accuracy) with a maximum distance of 5 pixels from the selected pixel.

The output list, here named 'shifts.list', will contain the names of the on frames and their two dimensional shifts.

Some instructions are in place, to explain the way this interactive command (xdshifts) works:

- The first thing that happens is that one should use the keys 'p' (previous image) and 'n' (next image) to step through all images just to find a suitable reference star that is well placed (close to the centre and bright) in all images. When you've found a good reference star, just press 'q' and move on to the next step.
- 2. Now, the 'fun' begins in the form of marking the reference star in all images (only one star / image). Locate the star in the first image, press 'a' to mark it, then 'n' to move on to the next image. This continues until you've done all images and get back to the first one. Now press 'q' to move on...
- 3. The reference frame (on1) is now displayed. Just mark the reference star one more time with 'a' and press 'q'

If you've completed these steps correctly (not marking more than one star or the same star several times in the same image) you will get an output list of the shifts called 'shifts.list'.

You can look at the list using e.g. less shifts.list (press 'q' to get out).

STEP 5 - MAKING A BAD PIXEL MAP

As can be seen in the masterflat, NOTCam contains a lot of dead pixels. The amount changes between about 2-5%, since dead pixels comes and goes each time NOTCam is heated to room temperatured and opened up to check/fix something.

Making a map of bad pixels is a trial and error thing, so one often needs to experiment a bit with the different parameters for a command called makemask. It is also possible to block out bad regions manually (not found with makemask).

epar makemask

inlist = masterflat_Ks outlist = badpixels subsamp = 1 checkli = no filtsiz = 15 nsmooth = 1 nsigrej = 5 thresht = nsigma nsigthr = 3 constth = 1

The above settings works OK for the example data, except for a horizontal line in the middle of the detector and a region at the top. This doesn't matter for our exercise, but for serious reductions these can be blocked manually by measuring their extent using imexa (approximating regions with squares) and changing the corresponding values in badpixels.pl

The resulting bad pixel map is called badpixels.pl and has the same size as the masterflat_Ks frame used to make the map. 0's and 1's are used to mark good and bad pixels, but since 'xnregistar' (which we will use in the next step) has the opposite definition of how a bad pixel map should be defined, we need to do the following: