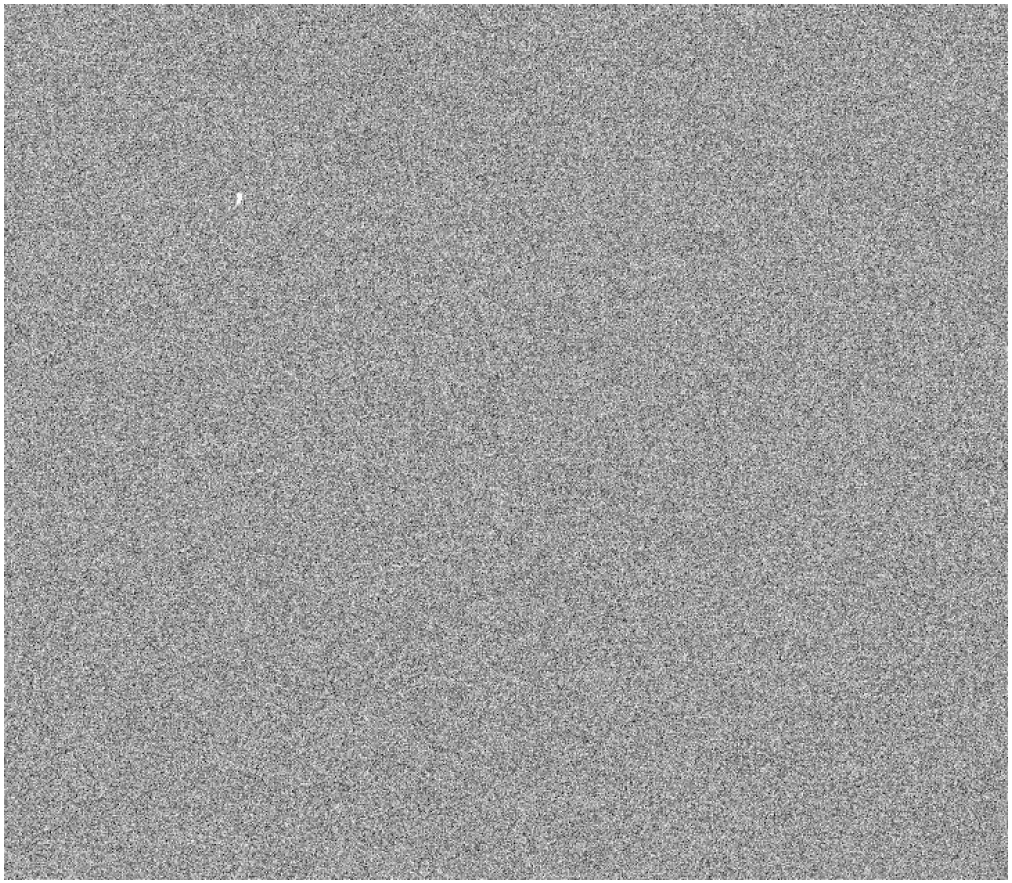
Review - bias

This is the bias file from last week. It's a decent representation of the zero correction that must be done, but its nowhere near perfect. Notice it is not uniform (most easily seen is the cosmic ray hit during the zero second exposure and subsequent read-out). A better statistical representation of this can be found by taking multiple exposures and combining them.

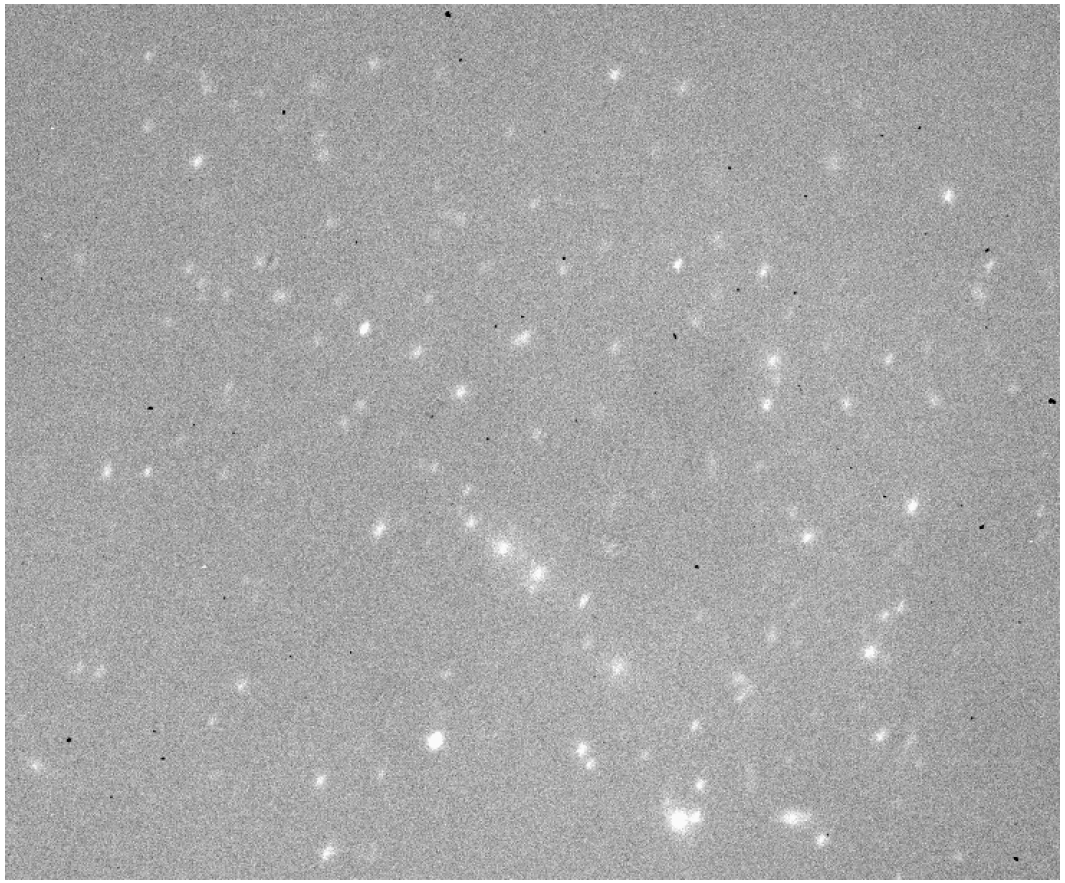


Review - flat field

Similarly, this is the flat field image from last weeks tutorial. This works well enough to account for varying sensitivity, but like the bias frame, there will be cosmic rays in this image too that can easily be removed by combining multiple exposures. Notice all the dark spots. These are dead pixels. At this stage those are a defect we can't correct.

Review - image

Finally, the image from last week. Even with the two basic corrections, again, dead pixels and cosmic rays contaminate the image. Multiple exposures alone won't fix both of these problems. The images also have to be dithered. Basically, this means the telescope needs to be moved slightly with each exposure. Therefore, the light from stars, galaxy, and the background fall onto different pixels of the detector.



Exercise: Lists

- We will be working out of the top level lectureTwo directory
- Lists will be used often in IRAF because we will most of the time be dealing with many files simulataneously
- Some cases, list generation will be trivial, like in the example to the right
- When dealing with raw files, or output lists, the list needs to be modified from this basic form

```
lm4-flspa-08:lectureTwo ckirkpat$ ls
bias            image           rawimage_i      skyflat
lm4-flspa-08:lectureTwo ckirkpat$ ls bias/*.fits > bias.list
lm4-flspa-08:lectureTwo ckirkpat$ ls
bias            bias.list       image           rawimage_i      skyflat
lm4-flspa-08:lectureTwo ckirkpat$ cat bias.list
bias/bias1_raw.fits
bias/bias2_raw.fits
bias/bias3_raw.fits
bias/bias4_raw.fits
bias/bias5_raw.fits
bias/bias6_raw.fits
bias/bias7_raw.fits
bias/bias8_raw.fits
bias/bias9_raw.fits
lm4-flspa-08:lectureTwo ckirkpat$
```

Create bias list of files

- For raw files (multi-dimensional), you must specify the extension for IRAF by appending it to the end
- In the top directory, run these simple (unix) commands
  - $ ls bias/*.fits > bias.list
  - For Mac: $ sed -i - 's/.fits/.fits[1]/g' bias.list
  - For Linux: $ sed -i 's/.fits/.fits[1]/g' bias.list
- There is no need for an output list because the output will only be a single file.
- Make a list file for the images (image.list) and the flat fields (flat.list) as well

Create output lists for bias subtraction

Mac

- $ ls image/*.fits > image_o.list
- $ sed -i - 's/_raw.fits/.fits/g' image_o.list
- $ sed -i - 's/AL/z/g' image_o.list
- $ rm *.list-

Linux

- $ ls image/*.fits > image_o.list
- $ sed -i 's/_raw.fits/.fits/g' image_o.list
- $ sed -i 's/AL/z/g' image_o.list

Repeat for flat_o.list

Exercise: Master Bias

zerocombine

- Location: noao -> imred -> ccdred
- Paramters to be edited: > epar zerocombine
  - input: @bias.list
  - (combine): median
  - (ccdtype): set this field to blank
  - (rdnoise): RDNOISE or 4.2
  - (gain): GAIN or 0.33
- Run zerocombine and a master bias should be created called ‘Zero.fits’

Remove bias with ccdproc

- > epar ccdproc
- First step, set all Yes/No questions to NO
- Parameters to edit:
  - images: @image.list
  - (output): @image_o.list
  - (ccdtype): set this field to blank
  - (trim): Yes
  - (zerocor): Yes
  - (trimsec): [58:2096,2:2036]
  - (zero): Zero
- Repeat for sky flats

Bias removed

- Each folder should now contain a new version of the file with the bias removed and image trimmed
- Do not override previous files, often you find you have to go back and redo steps or make comparisons!
- Next, we will make the flat correction

combine

- Location: noao > imred > ccdred
- Parameters to be edited: epar combine
  - input: @flat_o.list
  - output: Flat
  - ccdtype: set this field to blank
  - (combine): median
  - (reject): crreject
  - (rdnoise): RDNOISE or 4.2
  - (gain): GAIN or 0.33
- Run combine and a master flat image should be created called ‘Flat.fits’

Correct flat field with ccdproc

- One last list is needed for the flat field output
  - Follow the previous example, but this time create a list (fimage.list) where the final files will have ”fzAL…” at the beginning, signifying they are flat-fielded, bias subtracted images
- Again, set all Yes/No questions to NO
- Edit parameters: epar ccdproc
  - images: @image_o.list
  - (output):@fimage.list
  - ccdtype: set this field to blank
  - (flatcor): Yes
  - (flat): Flat

Results

- After running ccdproc, there should be three images with (some what) uniformity
- Why don’t the images look very good?

Exercise: Super Sky-flats

Making better flat-field corrections

- The sky-flats work well in r’ band (or bluer), but i’ band data was not successful.
- Sky-flats are incapable of correcting the fringe pattern that occurs when observing during the night.
- The theory for super sky-flats is if you stack every image you take, the random pointings on the sky will average to a blank field

Correct for fringe pattern

- The images contain a constant fringe pattern that needs to be removed before the flat field correction.
- The approach is to stack as many images as possible, then subtract out the large scale variations to leave behind only the fringing signal.
- This fringe pattern can then be removed from all images for the given filter.
- A second round of combining generates the final sky-flat that is fringe free

Getting started

- The rest of this exercise will take place in the rawimage_i/ directory
  - $ cd rawimage_i
- Create a new list of the images for input and output (image.list and image_o.list)
  - Use the previously learned methods, remember to add the proper extension for raw files
- Create a new instance of IRAF in this folder
- Remember to make the zero correction using the already created master bias
  - (zero): ../Zero
- Finally, create a new list of zero corrected images (zimage.list)

objmasks

- Package: > nproto
- Black box routine for masking out objects in our images before combining
- For optimization of this routine, first edit the hidden task objmasks1 (> epar objmasks1)
  - (fitstep): 10
  - (fitxorder): 1
  - (fityorder): 1
- Save & Quit, do not execute

> epar objmasks

- Create a new list for the object mask names. This has to be done separately because they use a special extension
  - $ ls *.fits > om.list
  - Use your knowledge of the ‘sed’ command to add “om” to the front of the file, and also to remove the .fits extension and replace it with just [pl]
- Specify your input and output lists
  - images: @zimage.list
  - objmasks: @om.list
  - (skys): sky//@zimage.list

sflatcombine

- Found in package mscred
- > epar sflatcombine
  - input: @zimage.list
  - (reject): ccdclip
  - (ccdtype): set this field to blank
  - (masktype): !objmask
  - (rdnoise): RDNOISE or 4.2
  - (gain): GAIN or 0.33
- After running should have a final image “Sflat” or “Sflat1” that is devoid of objects (I don’t know why it appends the 1 sometimes).

mscmedian

- In the same package should be the routine mscmedian, which will run on the new combined image to generate a template median frame
- > epar mscmedian
  - input: Sflat
  - output: MedianTemp
  - xwindow: 129
  - ywindow: 129
- This should now have a file MedianTemp.fits that only contains the large scale variations.

Fringe frame

- Using imarith, a task for preforming simple arithmetic on images, subtract the median frame from the combined frame.
- > epar imarith
  - operand1: Sflat
  - op: -
  - operand2: MedianTemp
  - result: Fringe
- You should now have the fringe only frame

Remove fringe pattern

- The fringe pattern can be removed from all images at once.
- Again, in mscred, use rmfringe for this purpose
- > epar rmfringe
  - input: @zimage.list
  - output: d//@zimage.list
  - fringe: Fringe
  - masks: @om.list
  - (background): sky//@zimage.list
- Inspect random files to makes sure the pattern is gone

Make Super Sky-flat

- These new, defringed images will be combined to get the final sky-flat
- Again, > epar sflatcombine (parameters from last run should be saved)
  - input: d//@zimage.list
  - output: Sflat2
- This will either generate a file called ‘Sflat2’ or ‘Sflat21’

Flat field correct images

- Apply the super flat field correction in the same way as with the sky flats
- Use ccdred version of ccdproc, not the mscred version
  - Type: noao > imred > ccdred
  - Mscred version of ccdproc has never worked for me
- Edit parameters: epar ccdproc
  - images: d//@zimage.list
  - (output): sd//@zimage.list
  - ccdtype: set this field to blank
  - (flatcor): Yes (All others to No)
  - (flat): Sflat2
- Check the new output files for smoothness of corrections. Do they look better than the sky-flat corrected images?

Homework: IRAF Scripting

- Now that you have mastered the basics of image reduction, there is no need to ever do any of this again… by hand
- Simple scripts of almost the exact same commands can run automatically and take only a few minutes to complete
- Create your own separate scripts for generating master bias, flat field, and super sky-flat corrected images
- Send the script to me: charles.kirkpatrick@helsinki.fi

Example bias.cl

```
# RUNS INSIDE BIAS DIRECTORY!!
#
# Assume file directory:
#   bias/
#   rawimage_(filter)
#

# Generate bias file list
!ls *.fits > bias.list
!sed -i - 's/.fits/.fits[1]/g' bias.list
!rm *.list-

# Open needed packages
noao
imred
ccdred

# Run zerocombine
unlearn zerocombine
zerocombine.combine = 'median'
zerocombine.ccdtype = ' '
zerocombine.rdnoise = 4.2
zerocombine.gain = 0.33
zerocombine input='@bias.list'
```

```
# Generate input and output file lists
!ls ../*_*/*.fits > input.list
!cp input.list output.list
!sed -i - 's/.fits/.fits[1]/g' input.list
!sed -i - 's/AL/zAL/g' output.list
!rm *.list-

# Remove bias with ccdproc
unlearn ccdproc
ccdproc.output = '@output.list'
ccdproc.ccdtype = ' '
ccdproc.fixpix = no
ccdproc.overscan = no
ccdproc.trim = yes
ccdproc.zerocor = yes
ccdproc.darkcor = no
ccdproc.flatcor = no
ccdproc.trimsec = '[58:2096,2:2036]'
ccdproc.zero = 'Zero'
ccdproc images='@input.list'
```