Req 6.3.2

Title:

Trim, de-bias, flatfield.

Objective:

Trim, de-bias, and flatfield the raw data.

This second step of the image part of the Pipeline applies a number of calibration files to a raw science image after trimming and overscan correcting the image. This includes bias subtraction and flat-fielding, as well as fringe removal.

Several methods for overscan correction are possible, provided the calibration files have been derived using the same method (see Section 7.1 for further details).

Fringe removal is done using an automatic fringe scaling algorithm. This method assumes the standard deviation of the pixel values in the reduced science image is a function of a fringe map scale factor. Pixel statistics can then be used to calculate a minimum for this function. Masks and sigma thresholding are used to exclude bad pixels, stars and focus on pixels in fringe minima and maxima.

Due to problems with non-uniform illumination, as described in **req.**548, the zeropoint is not uniform over the image. Therefore an illumination correction must be applied. Because this corrects for a non-uniform illumination, the background would not be flat anymore after this correction. Therefore, the background is subtracted prior to multiplying the data with the illumination correction image.

Inputs:

SeqFile– 631 *Raw science frame*
CalFile– 541 *Master Bias frame*
CalFile– 546 *Master flatfield*
CalFile– 545 *ff-fringe*
CalFile– 548 *Illumination correction*

Outputs:

SeqFile– 632 *Reduced science frame*
Updated image statistics (before and after background removal)

Estimated time needed:

Approximately 30 seconds/CCD

Recipe:

Note that the recipe below performs step **seq.– 632** through **seq.– 635** and the regridding part of **seq.– 636.**The target grid, defined by the input parameters ra, dec and pixelscale, has to be the **same** for all images that are to be coadded later by **Recipe– Coadd.**

```
Reduce -i raw -b bias -f flat -p photom -g GAIN -ra RA -dec DEC
      -ps PIXELSCALE [-fr fringe] [-c cold] [-h hot]
      [-il illumination] [-sl THRESHOLD_LOW]
      [-sh THRESHOLD_HIGH] [-oc OVERSCAN_CORRECTION]
      [-ft IMAGE_THRESHOLD] [-fl FRINGE_THRESHOLD_LOW]
      [-fh FRINGE_THRESHOLD_HIGH] [-ct DETECTION_THRESHOLD]
      [-sit DETECTION_THRESHOLD] [-sht HOUGH_THRESHOLD]
      [-dt ASTROM_DETECTION_THRESHOLD]


raw           : raw science frame
bias          : master bias frame
flat          : master flat frame
photom        : photometric parameters
GAIN          : gain (float)
                Range of allowed values: 0.1 - 10.0.
RA            : right ascension of the center of the output image
                (decimal degrees, float)
                Range of allowed vales: 0.0 - 360.0
DEC           : declination of the center of the output image
                (decimal degrees, float)
                Range of allowed values: -90.0 - 90.0
PIXELSCALE    : pixel scale of output image (arcsec, float)
                Range of allowed values: 0.05 - 1.0
fringe        : fringe map
cold          : cold pixel map
hot           : hot pixel map
illumination : illumination correction frame
THRESHOLD_LOW         : saturation cutoff low (float)
                        Range of allowed values: 0.0 - 200.0. De-
fault 50.0.
THRESHOLD_HIGH        : saturation cutoff high (float)
                        Range of allowed values: 40000.0 - 70000.0.
```

```
                        Default 50000.0.
OVERSCAN_CORRECTION   : overscan correction mode (integer).
                        Description of allowed values:
                          0: apply no overscan correction (default)
                          1: use median of the prescan in the
                             x-direction
                          2: use median of the overscan in the
                             x-direction
                          3: use median of the prescan in the
                             y-direction
                          4: use median of the overscan in the
                             y-direction
                          5: use the per-row value of the pres-
can in
                             the x-direction
                          6: use the per-row value of the over-
scan in
                             the x-direction
IMAGE_THRESHOLD       : (for fringe map scaling) do not use pix-
els from
                        science image outside this threshold (float)
                        Range of allowed values: 2.0 - 10.0. De-
fault: 5.0
FRINGE_THRESHOLD_LOW  : do not use pixels from fringe map below
this
                        threshold (float)
                        Range of allowed values: 1.0 - 3.0. De-
fault: 1.5.
FRINGE_THRESHOLD_HIGH : do not use pixels from fringe map above
this
                        threshold (float)
                        Range of allowed values: 3.0 - 10.0. De-
fault: 5.0.
DETECTION_THRESHOLD   : Sextractor detection threshold used for
cosmic ray
                        detection (float)
                        Range of allowed values: 2.0 - 20.0. De-
fault: 5.0.
```

```
DETECTION_THRESHOLD(2) : Sextractor detection threshold used for
satellite
                         track detection (float)
                         Range of allowed values:  Default: 5.0.
HOUGH_THRESHOLD        : Theshold for satellite tracks in Hough
image
                         (float)
                         Range of allowed values: TBD. Default 1000.0.
ASTROM_DETECTION_THRESHOLD : Sextractor detection threshold used
for
                             extracting astrometric catalog (float)
                             Range of allowed values: 5.0 - 15.0.
                             Default: 10.0.
```

Needed functionality:

image trim (eclipse.image.extract_region)
image arithmetic (eclipse.image_add/sub/div/mul)
image statistics (eclipse.image.stat_opts)
image statistics (eclipse.iter_stat)

CA:

1. Perform an overscan correction and trim the science image. 2. Subtract the master bias. 3. Divide by the master flatfield.
4. Check for filters that may require fringe subtraction. Scale and subtract a fringemap if so. The scaling is done by assuming that the standard deviation of the pixel values in the reduced science image depends on the scale factor used for the fringe map, with a minimum σ for the best scale factor. If the pixel values in the resulting de-fringed image are given by $x_{ij} = I_{ij} - aF_{ij}$, (I is the science image, and F the fringe map) and $\sigma = \frac{1}{N-1}\sum_{ij}(x_{ij} - \mu)^2$, then assume $\frac{d\sigma}{da} = 0$ for the best scale factor a. This results in the following formula for a:

$$a = \frac{\sum(IF) - N\mu_I\mu_F}{\sum(F^2) - N\mu_F\mu_F},$$

where I is the (flat-fielded, de-biased) science image, F the fringe map, N the number of pixels used in the statistics (i $\times$ j, which is not equal to NAXIS1 $\times$ NAXIS2 if there are any masked pixels), and μ_I and μ_F the average of the

pixel values in I and F respectively. Pixel maps (sigma thresholding) are used to exclude bright stars and select mostly pixels in fringe minima/maxima.
5. Compute image statistics.
6. Subtract the background.
7. Multiply the image with the illumination correction frame.
8. Compute the image statistics again.

CAP:

```
Constants:
  BACKGROUND_CONFIG : SExtractor configuration parameters and their
values
                      in case they differ from the SExtractor de-
faults.
          CHECKIMAGE_TYPE = 'BACKGROUND'
          DETECT_THRESH   = 1000
          ANALYSIS_THRESH = 3.0
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'N'
          GAIN            = 1
          MEMORY_PIXSTACK = 100000

ima = eclipse.trim_and_overscan(raw)
ima = (ima-bias)/flat

if filter.has_fringes():
  FUNCTION make_mask():
    mask = ~(hotpixelmap & coldpixelmap & saturatedpixelmap)
    stat = ima.iter_stat()
    m, s = stat.median, stat.stdev
    mask &= ima.thresh_to_pixmap(m-IMAGE_THRESH*s, m+IMAGE_THRESH*s)
    fstat = fringe.iter_stat()
    fm, fs = fstat.median, fstat.stdev
    mask &= ( fringe.thresh_to_pixmap(fm-FRINGE_THRESH_HIGH*fs,
                                      fm-FRINGE_THRESH_LOW*fs) |
              fringe.thresh_to_pixmap(fm+FRINGE_THRESH_LOW*fs,
```

```
                                        fm+FRINGE_THRESH_HIGH*fs)
)
    return mask

  # calculate fringe map scale factor, scale and subtract from sci-
ence
  mask = make_mask()
  I = ima
  F = fringe
  mean_I = I.Stat(pixelmap=mask).mean
  mean_F = F.Stat(pixelmap=mask).mean
  N = mask.count()
  a = (sum(mask,I*F)-N*mean_I*mean_F) / (sum(mask,F*F)-N*mean_F*mean_F)
  ima = ima - a * fringe

  stats1 = ima.iter_stat()

  # compute background, and subtract
  sextractor(ima, BACKGROUND_CONFIG,
                  CHECKIMAGE_NAME=tmp_ima)
  ima = ima-tmp_ima

  # apply illumination correction
  ima = ima * illumination
  stats2 = ima.iter_stat()
```