Req 5.2.3

Title:

CCD gain determination

Objective:

Determine CCD gain and variation with time

Determine the conversion factor between the signal in ADU's supplied by the readout electronics and the detected number of photons (in units e^-/ADU) and monitor variations in time.

The gain factors are needed to convert ADU's in raw bias-corrected frames to the number of electrons, i.e. detected photons.
Take two series of 10 dome flatfield exposures with wide range of exposure times. Derive the rms of the differences of two exposures taken with similar exposure (integration time). Exposure differences of pairs should not exceed 4%. The regression of the square of these values with the median level yields the conversion factor in e^-/ADU (assuming noise dominated by photon shot noise).

Fulfilling or fulfilled by:

Selfstanding, also measures detector linearity (**req.**533).

When performed/frequency:

daytime- Commissioning, in RP once week.

Sources, observations, instrument configurations:

Dome flat field exposures on lamp, with t_{exp}= 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60 and 2 seconds. Use r' filter.

Inputs:

Number of raw domeflats such that for each exposure time there are two different raw domeflats. The total number of raw domeflats has to be larger than 4.
A raw domeflat with a short exposure time.
A raw domeflat with a long exposure time.
CalFile– 541 $Master\ Bias\ frame$
CalFile– 523 $Conversion\ factor\ e^-/ADU$ older versions.

Outputs:

CalFile– 523 $Conversion\ factor\ e^-/ADU$
The **CalFile– 523** corresponds to the QC parameter gain (single number).

Required accuracy, constraints:

Accuracy: In units of e^-/ADU, from lab values or found empirically. Variation

in time less than 1%.

Estimated time needed:

Observations: 1 hour. Reduction: 3 min./CCD.

Priority:

essential

TSF:

Mode– Stare N=20
(**TSF– OCAM_img_cal_domeflat,** t_{exp}= 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60, 2 s.).
= **TSF– OCAM_img_cal_gain**

Recipe:

```
Gain -i <raw_domeflat_list> -b master_bias -s short_raw_domeflat
                                           -l long_raw_domeflat
                                           [-oc OVERSCAN_CORRECTION]

raw_domeflat_list   : list of even number of raw dome flats
master_bias         : master bias frame
short_raw_domeflat  : raw dome flat - short exposure time
                      One of <raw_domeflat_list> with an exposure
level
                      of about 1000 ADU
long_raw_domeflat   : raw dome flat - long exposure time
                      One of <raw_domeflat_list> with exposure level
                      of about 20000 ADU
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 prescan
in
```

```
                                the x-direction
                            6: use the per-row value of the overscan
in
                                the x-direction
```

Also satisfies **req. 533** *CCD Linearity*. Before applying this recipe, use **Recipe– Split**—which is documented in **seq.– 631**—with the `-t dome` option to split the raw multi-extension FITS input files.

Needed functionality:

image - processing (eclipse.trim_and_overscan());
image - arithmetic (eclipse.image_add(), eclipse.image_sub(), eclipse.image_sub_local());

image - statistics (eclipse.image_getmedian(), eclipse.iter_stat());
misc - regression (minimal least-squares fit)

CA:

Assume bias corrected dome flats.

nb: monitoring in time not implemented in recipe

Process (make):
1. Divide the observations into pairs of equal exposure time
2. Trim, overscan-correct, and de-bias each exposure.
3. For each pair of exposures produce a sum and a difference image.
4. For each sum image measure the median
5. For each difference image measure the standard deviation (reject 5σ outliers).
6. Perform a minimum least-squares fit of the medians divided by 2 as a function of the squared standard deviations (variances).
7. The fitted slope is the gain. The offset should be equal to the read noise^2

Verification (verify):
1. The gain should be between TBD and TBD.

Interactive analysis:
1. A plot of exptimes vs. median_sum gives a measure of the linearity (**req.**533)

CAP:

```
domeflat_pairs = make_pairs(domeflats)
rms_diff = []
```

```
median_sum = []
exptimes = []
measurements = []

for pair in domeflat_pairs:
    exptime = eclipse.header(pair[0]).get_key('EXPTIME')
    exptimes.append(exptime)

    ima1 = eclipse.trim_and_overscan(pair[0])
    eclipse.image_sub_local(ima1, bias)

    ima2 = eclipse.trim_and_overscan(pair[1])
    eclipse.image_sub_local(ima2, bias)

    exptimes.append.eclipse

    sum = eclipse.image_add(ima1, ima2)
    median = eclipse.image_getmedian(sum)
    median_sum.append(median/2)

    diff = eclipse.image
    stats = eclipse.iter_stat(diff, MAXIMUM_ITERATIONS, REJECTION_THRESH
    rms_diff.append(stats.stdev)


    measurements.append(stats.stdev**2, median/2)

coeffs, errors = least_squares_fit(measurements)
gain   = coeffs[1]
```