Req 5.3.3

Title:

CCD Linearity

Objective:

Characterize the linearity of the system over the full dynamic range of the A/D converter.

Both the overall absolute linearity of the system and the pixel-to-pixel variation in linearity are of interest.

The overall linearity of the system can be obtained by measuring the counts as a function of exposure time for a series of dome flats. The data to use for this can be the same as for the measurement of the Gain (**req.**523, q.v.)

The pixel-to-pixel variation in the linearity is obtained by dividing a flatfield with a mean exposure level of more than 30000 ADU by a flatfield with an exposure level of less than 1000 ADU. Pixels that deviate more than 5σ from the mean, in this divided image, have an anomalously high nonlinearity. This map of nonlinear pixels may be used in conjunction with the hot and cold pixel maps to produce a map of bad pixels. One can use suitable long and short exposures from the measurements of the Gain (**req.**523, q.v.) for this.

In addition, during a cloudy night, once per year, the linearity will be checked by taking various exposures with a variety of exposure times on the dome screen. These data will be subject to interactive analysis.

Fulfilling or fulfilled by:

Data reduction of **req. 523** *CCD Gain*

When performed/frequency:

daytime- Commissioning, in RP once per month, dark dome test once/year

Inputs:

Raw dome flatfields
CalFile– 541 *Master Bias frame*

Outputs:

CalFile– 533P *CCD Linearity Plot* A measure of the overall nonlinearity can be obtained from this plot

CalFile– 533M *CCD Linearity map* A map of (anomalously) non-linear pixels can be used in conjunction with hot and cold pixel maps.

Required accuracy, constraints:

better than 1% on the photometric scale

Estimated time needed:

Observation: None

Priority:

essential

TSF:

Use same raw data as for **req. 523** *CCD gain*

Recipe:

Recipe– Gain, see **req. 523** *CCD gain*

Needed functionality:

image - processing (eclipse.trim_and_overscan())
image - arithmetic (eclipse.image_sub_local())
image - statistics (eclipse.iter_stat())
image - mask (eclipse.image_threshold2pixelmap)

CA:

The gain recipe (**req.**523, q.v.) takes a set of raw dome flats with exposure times of 2, 60, 50, 4, 8, 40, 30, 1, 16, 24, 24, 16, 1, 30, 40, 8, 4, 50, 60 and 2 seconds, and records an average of the median for each pairs of images with the same exposure time. One can plot this median vs. the exposure time to asses the overall linearity of the CCD.

For the linearity map (**CalFile– 533M**) use:

Process (make)

1. Assume a long and short raw dome flat exposure not affected by global non-linearity issues (for example: numbers 6 and 4 of the sequence).
2. Trim, overscan correct and de-bias the long and short exposure.
3. Divide the long exposure by the short expoure.
4. Iteratively estimate statistics ($mean$, σ).
5. Construct a pixelmap using thresholds ($mean - 5\sigma$, $mean + 5\sigma$).
6. Count the number of bad pixels in the pixelmap.

CAP:

```
Constants:
  MAXIMUM_ITERATIONS  : statistics measurement (integer)
                        Value: 3
  REJECTION_THRESHOLD : sigma rejection for bad pixels (float)
                        Value: 5.0

# div = (long-bias)/(short-bias)
div = eclipse.trim_and_overscan(long)
eclipse.image_sub_local(div, bias)
short = eclipse.trim_and_overscan(short)
eclipse.image_sub_local(short, bias)
eclipse.image_div_local(div, short)

stats = eclipse.iter_stat(div, MAXIMUM_ITERATIONS, REJECTION_THRESHOLD)
thresh_lo = stats.avg_pix - REJECTION_THRESHOLD*stats.stdev
thresh_hi = stats.avg_pix + REJECTION_THRESHOLD*stats.stdev

pixelmap = eclipse.image_threshold2pixelmap(div, thresh_lo, thresh_hi)

# eclipse pixelmap counts good pixels
count = pixelmap.lx * pixelmap.ly - pixelmap.count
```