

Req 5.2.2

**Title:**

CCD bad/hot pixel map

**Objective:**

Determine CCD bad/hot pixels.

$5\sigma$  outliers in the master bias frame are bad-hot pixels. These pixels should be recorded and ignored (assigned a weight of 0) in dedithering and dejittering, as well as source extraction. For this purpose the bad/hot pixel map is used to assign a weight of zero to the affected pixels in the weight map (**seq.- 633**). The search for hot pixels would also identify traps.

**Fulfilling or fulfilled by:**

Additional data reduction of **CalFile- 541** *Master Bias frame* to determine cold pixels

**When performed/frequency:**

daytime- Commissioning, in RP twice per week.

**Inputs:**

**CalFile- 541** *Master Bias frame*

**CalFile- 522** *Bad/hot pixel map* previous version

**Outputs:**

**CalFile- 522** *Bad/hot pixel map, number of hot pixels*

**Required accuracy, constraints:**

Number of hot pixels to be determined by experience/lab values.

The total number of bad pixels (hot pixels + cold pixels) is less than 80000 (checked in **req. 535** *Cold pixels*)

Difference in number of hot pixels w.r.t. reference value, less than 100.

**Estimated time needed:**

Observation: None. Reduction: < 20 sec./CCD.

**Priority:**

essential

**TSF:**

Use master bias (**req.541**)

**Recipe:**

Hot\_Pixels -i master\_bias [-max MAXIMUM\_ITERATIONS]

[-rej REJECTION\_THRESHOLD]

master\_bias : master bias image  
MAXIMUM\_ITERATIONS : maximum number of iterations for statistics  
measurement (integer).  
Range of allowed values: 2 - 10. Default:  
5  
REJECTION\_THRESHOLD : rejection threshold for bad pixels in sigma  
(float).  
Range of allowed values: 1.0 - 10.0. Default:  
5.0

**Needed functionality:**

image - statistics (eclipse.iter\_stat)  
image - mask (eclipse.image\_threshold2pixelmap)

**CA:**

Processing (make):

1. Iteratively estimate statistics ( $mean, \sigma$ ) of the input bias.
2. Construct a pixelmap using thresholds  $(-\infty, mean + 5\sigma)$ .
3. Count the number of bad pixels in the pixelmap.

Verification (verify):

1. The number of bad pixel should be less than TBD

Trend Analysis (compare):

1. The difference in number of bad pixels should be less than 100

**CAP:**

```
stats = eclipse.iter_stat(bias,  
                          MAXIMUM_ITERATIONS,  
                          REJECTION_THRESHOLD)  
threshold = stats.avg_pix + REJECTION_THRESHOLD * stats.stdev  
pixelmap = eclipse.threshold2pixelmap(bias, -1E20, threshold)  
  
# eclipse pixelmap counts good pixels  
count = pixelmap.lx * pixelmap.ly - pixelmap.count  
if abs(count-previous.count) > MAXIMUM_HOTPIXELCOUNT_DIFFERENCE:  
    HOTPIXELCOUNT_DIFFERENCE_TOO_LARGE = 1
```