Req 5.3.5

Title:

CCD cold pixels

Objective:

Identify cold pixels.

Cold pixels should be recorded and ignored (assigned a weight of 0) in dedithering/dejittering and source extraction. For this purpose the bad/cold pixel map, together with the hot pixel map (**req.**522) is used to assign a weight of zero to the affected pixels in the weight map (**seq.– 633**).

Cold pixel maps are constructed from reduced dome (**req.**542) or twilight (**req.**543) flats. The flatfield is smoothed. The smoothed flat is used to flatten the flat. In this flatfielded image, pixels that are outside a given range (0.96-1.04) are taken to be cold pixels. Note that this invalidates any pixel whose gain differs significantly from its immediate neighbours. In particular, this also identifies pixels that are bright relative to their neighbours as "cold". Note, that pixels above the threshold are formally not "cold", but are flagged annyway. In the end, hot plus cold pixel map are combined in weightmap.

Fulfilling or fulfilled by:

Requires data reduction of master domeflat (**req.**542) or twilight (**req.**543) flat frames.

When performed/frequency:

daytime- Commissioning, in RP once per 3 months.

Sources, observations, instrument configurations:

Use master dome- or twilight flat. Use r' filter.

Inputs:

CalFile– 535 $Cold\ pixel\ map$ previous version
CalFile– 542 $Dome\ flat$ or **CalFile– 543** $Twilight\ flat$

Outputs:

CalFile– 535 $Cold\ pixel\ map$

Required accuracy, constraints:

Quality Check: 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. Difference in number of cold pixels w.r.t. reference version less than 100.

Estimated time needed:

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

Priority:

very important

TSF:

Use master dome- (**req.**542) or twilight flat (**req.**543)

Recipe:

```
Cold_Pixels -i flat [-low THRESHOLD_LOW] [-high THRESHOLD_HIGH]

flat           : the master dome- or twilight flat
THRESHOLD_LOW  : the lower flagging threshold (float)
                 Range of allowed values: 0.90 - 1.00. Default:
0.96
THRESHOLD_HIGH : the higher flagging threshold (float)
                 Range of allowed values: 1.00 - 1.10. Default:
1.04
```

(could be incorporated in **Recipe– Dome_Flat** or **Recipe– Twilight_Flat**)

Needed functionality:

image - arithmetic (eclipse.image_div)
image - mask (eclipse.image_threshold2pixelmap)
sextractor - background

CA:

1. Make a smoothed flatfield using sextractor.
2. Divide the flatfield by the smoothed flatfield.
3. Construct a pixelmap using thresholding.
4. 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

Note that we use sextractor to produce the smoothed image. Sextractor uses a robust algorithm to estimate the background on a grid and interpolate between

these grid points. By measuring this background for the flatfield we essentially have a fast smoothing algorihm with a large kernel, that is relatively insensitive to bad pixels.

CAP:

```
# make a smoothed 'background'
sextractor(flat, DETECT_THRESH=1000,
                 CHECKIMAGE_TYPE='BACKGROUND',
                 CHECKIMAGE_name=smoothed_flat)

# flatten the flat
normalized = eclipse.image_div(flat, smoothed_flat)

# threshold the bad pixels
pixelmap = eclipse.threshold2pixelmap(normalized,
                                      THRESHOLD_LOW,
                                      THRESHOLD_HIGH)

# eclipse pixelmap counts good pixels
count = pixelmap.lx * pixelmap.ly - pixelmap.count
if count > MAXIMUM_COLDPIXELCOUNT:
    COLDPIXELCOUNT_TOO_LARGE = 1
```