Req 5.4.3

Title:

Master Twilightflat

Objective:

Determine master twilightflat frame, using observations of the twilight sky.

Twilightflat observations will be attempted for each passband that is observed during that night. If insufficient twilight time is available then the twilightflat observations are taken preferably in the previous or subsequent night. In addition twilightflats of the 4 keybands will be taken at least once a week.

In order to minimize the spatial gradient in the sky brightness, the observations need to be made on the solar circle, i.e. the great circle through the zenith and the sun's position, at a zenith distance of about 20° in the solar antidirection. Preferably, the field of view does not include stars brighter than TBD magnitude.

The consortium will deliver a list a targets for twilight flats.

Master twilight flats are obtained through an average with sigma rejection procedure on a stack of raw twilight flats, intended to remove any contamination (including stars) present on individual raw twilight flats and reduce photon shot noise.

Fulfilling or fulfilled by:

Selfstanding

When performed/frequency:

Evening twilight.

An attempt will be made to observe twilightflats for all bands observed during the night. For all observed bands, twilightflats will be observed within a maximum of 2 nights. In addition, twilightflats for the keybands will be obtained at least once a week, irrespective of whether keybands were used for science observations during that week.

Sources, observations, instrument configurations:

Twilightflat with 5 observations per filter at 'empty' sky, near 20° from zenith in solar anti-direction. It may be advisable to determine a standard target list of approximately 10-20 empty fields, equally spaced in right ascension, at 20° from zenith. A 'standard' target list will be provided by consortium at commissioning.

The telescope should be tracking. The maximum jitter offset allowed by the TSF should be used in order to be able to remove stars. Observations should approximate 20000 ADU. Exposure time should be based on skybrightness, typically it may vary from 0.1 sec to 300 sec, changing rapidly during the twilight period. No effects of shutter are expected.

The Tyson and Gal technique/formula for adapting exposure time to the variations of the twilight brightness is implemented in the observing template (similar to WFI@2.2m). A trend analysis of twilightflat can be used to calibrate this formula.

Each science/standard observation preferably has an associated twilightflat observed within 2 days. We expect a total of 10 twilightflats to be observed during each twilight period.

Inputs:

Raw twilight flatfields
CalFile– 541 *Master Bias frame*
CalFile– 522 *Hot pixel map*
CalFile– 523 *Gain*
CalFile– 535 *Cold pixel map*

Outputs:

CalFile– 543 *Master Twilightflat frame*

Required accuracy, constraints:

Mean levels should be approximately 20000 ADU.

Estimated time needed:

Observation: in total 25 min./night. Reduction: 3 min./CCD.

Priority:

essential

TSF:

Mode– Jitter N=5
TSF– OCAM_img_cal_skyflat
Use an appropriate (empty) field, exposure time determined by Tyson-Gal formula.

Recipe:

```
Twilight_Flat -i <raw_twilightflats> -b bias -g GAIN [-c cold] [-
h hot]
              [-oc OVERSCAN_CORRECTION]
```

```
raw_twilightflats   : raw twilight flatfields
bias                : master bias frame
cold                : cold pixel map
hot                 : hot pixel map
GAIN                : the gain value in e/ADU (float)
                      Range of allowed values: 0.1 - 10.0
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 prescan
in
                           the x-direction
                        6: use the per-row value of the overscan
in
                           the x-direction
```

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

Needed functionality:

image - processing (eclipse.trim_and_overscan)
image - arithmetic (eclipse.image_sub, eclipse.image_div, eclipse.image_pow, etc.)
image - cube median (eclipse.cube_median)
image - statistics (eclipse.stat)
image - mask (eclipse.image_threshold2pixelmap)
pixelmap - binary AND (eclipse.pixelmap_binary_AND)

CA:

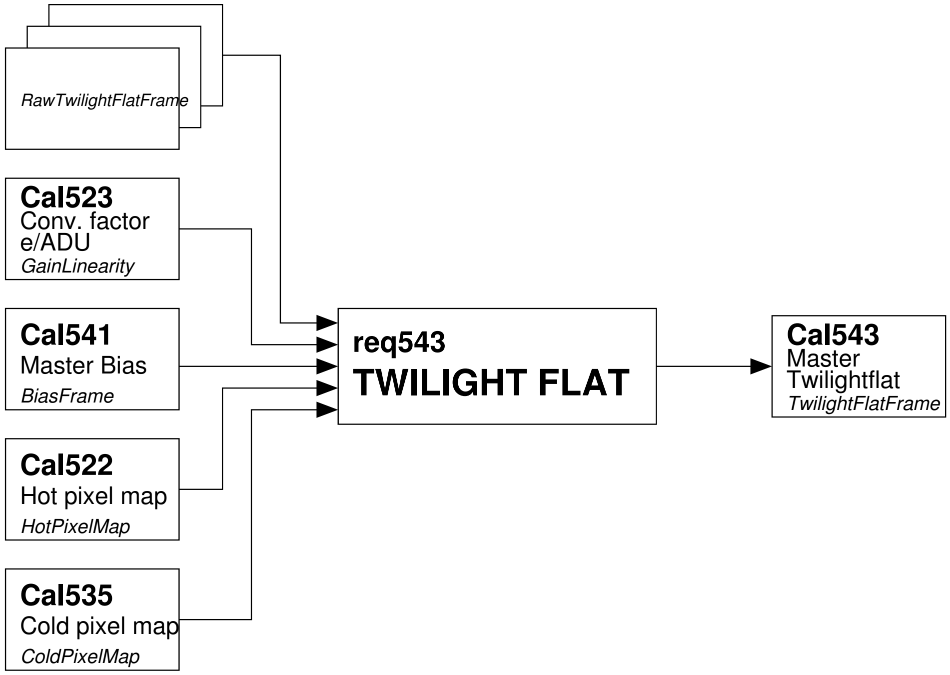

Fig 5.4.3 Dataflow and object class names (in small italic font) for req543

NOTE: The data reduction procedure is essentially the same as for dome flats (**req.**542).

Process (make):

1. Construct a mask from input hot and cold pixel maps.
2. Trim, overscan correct and de-bias the raw twilight frames.
3. Normalize frames to one illumination level (median pixel value 1.0).
4. Build a mean frame from the stack of input frames, rejecting 5σ outliers.
4.1 Compute the median frame of the stack of input frames.
4.2 Compute the error in each input frame ($\sigma = \sqrt{gain * counts}$).
4.3 Reject 5σ deviations in each input frame.
4.3 Compute the mean in the remaining data.

Verification (verify):
TBD

CAP:

`Constants:`

```
  SIGMA_CLIP    : the sigma clipping threshold

mask = eclipse.pixelmap_binary_AND(hot, cold)

scales = []
scaled_images = []

# Trim, overscan correct, de-bias and scale each input image
for raw_twilight in raw_twilightflats:
    reduced_twilight = eclipse.trim_and_overscan(raw_twilight)
    eclipse.image_sub_local(reduced_twilight, bias)
    stat = eclipse.image_stat_opts(reduced_twilight, pixelmap=mask)
    scaled_images.append(eclipse.image_mul(reduced_twilight, 1.0/stat.me
    scales.append(1.0/stat.median)

median_image = eclipse.cube_median(scaled_images)

# Record the good values
good_data  = eclipse.make_image(xsize, ysize)
good_count = eclipse.make_image(xsize, ysize)

# Loop over remaining data
i=0
for twilight in scaled_images:
    dev_image = eclipse.image_sub(median_image, twilight)
    err_image = eclipse.image_pow(twilight, 0.5))
    eclipse.image_div_local(dev_image, err_image)

    threshold = SIGMA_CLIP * sqrt(GAIN * scales[i])
    good = eclipse.threshold2pixelmap(dev_image, -threshold, thresh-
old)
    good = eclipse.pixelmap_2_image(good)

    eclipse.image_add_local(good_data,
                            eclipse.image_mul(twilight, good))
    eclipse.image_add_local(good_count, good)
    i=i+1
```

```
#
twilightflat = eclipse.image_div(good_data, good_count)
```