Req 5.4.5

Title:

Flat-field - fringing

Objective:

Determine the fringe pattern of the background.

Fringing due to variable strength of several skylines, mostly apparent at the long wavelengths, requires a different approach to background subtraction. Normally, after flatfielding, the background can be expected to be flat over the entire image, and a median of the image, excluding 5σ outliers, would in principle be sufficient to subtract the background.

In images that suffer from fringing we have to deal with a background that is variable on small ($<< 1'$) scales within the image, and can not be distinguished from sources. The image itself can, therefore, not be used to determine the background. However, given the fact that most observations are taken in jitter or dither mode, the information of several images can be combined to determine a background. This average should include enough observations to properly exclude contamination from sources, and, because the standard jitter/dither patterns only include 5 pointings, one background computation per jitter/dither is probably not sufficiently accurate. On the other hand, because the fringing pattern may vary with time and telescope position, a straight mean (the supersky) over an entire nights worth of data may also not be usable.

A suitable strategy to construct a fringed background image, usable for subtraction, thereby removing the fringe pattern, remains to be determined. If the fringe pattern is stable over the night, a decomposition of the night-sky flat in an additive and multiplicative term is feasable. The assumption that the high-frequency spatial component in the night-sky flat are fringes, while the lowest frequency components represent gain variations has been used with reasonable success.

Fulfilling or fulfilled by:

Data reduction of science and photometric standard observations

When performed/frequency:

Commissioning and when long wave science frames are taken.

Sources, observations, instrument configurations:

Use science and standard data to determine background

Inputs:

Raw science images

CalFile– 541 *Master Bias frame*
CalFile– 546 *Master flatfield*
CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*

Outputs:

CalFile– 545 *ff-fringe*

Estimated time needed:

Observation: None. Reduction: 5 min./CCD/filter for 15 science frames.

Priority:

very important

TSF:

Use same data as for night sky flat (**req.**544)

Recipe:

```
Fringe_Flat -i <raw_science_frames> -b bias -f flat [-c cold] [-
h hot]
            [-oc OVERSCAN_CORRECTION]

raw_science_frames  : the raw science images
bias                : the master bias image
flat                : the master flat image
cold                : the cold pixel map
hot                 : the hot pixel map
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 science` option to split the raw multi-extension FITS input files.

Needed functionality:

image - processing (eclipse.trim_and_overscan)
image - arithmetic (eclipse.image_sub_local, eclipse.image_div_local)
image - statistics (eclipse.stat, eclipse.iter_stat)
image - stacking (eclipse.cube)
cube - median average (eclipse.cube_avg_median)
pixelmap - binary AND (eclipse.pixelmap_binary_AND)

CA:

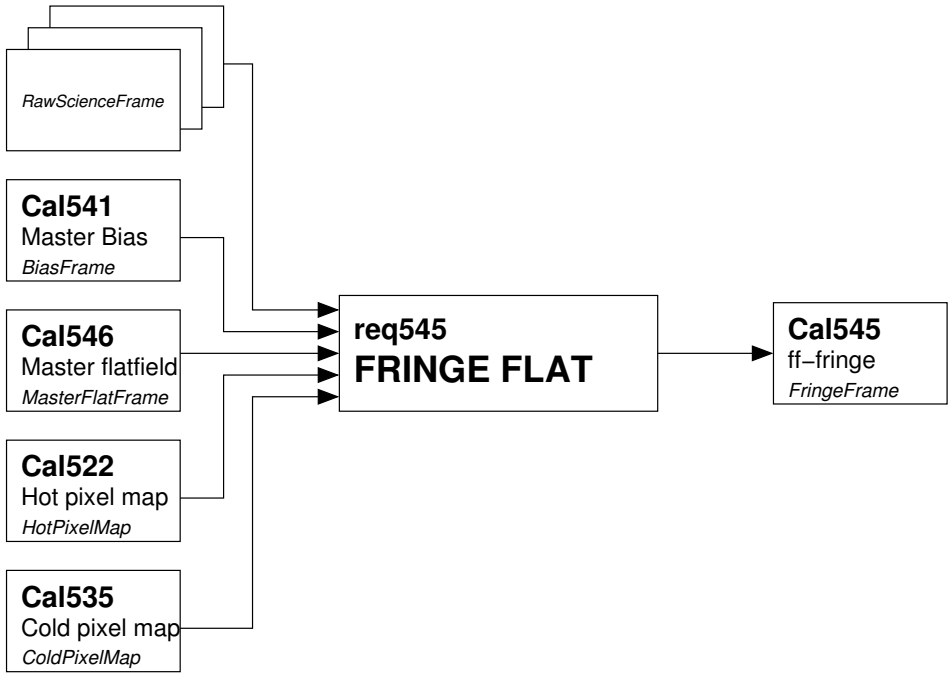

Fig 5.4.5 Dataflow and object class names (in small italic font) for req545

Process (make):

1. Check that at least three input science images are given; this is necessary

to calculate a median average of a cube of these.
2. Construct a mask from input hot and cold pixel maps.
3. Trim, overscan correct, de-bias and flat-field the input science images.
4. Iteratively estimate the statistics of the resulting science images.
5. Normalize each image by dividing by its median pixel value as determined above.
6. Stack these science images in a cube.
7. Determine the median average of the cube (this is an image).
8. Normalize the fringe map obtained in 7. (use mask obtained in 2.).
9. Subtract 1.0 from the normalized fringe map.
10. Assign bad pixels a value of 0 (use mask obtained in 2.).

Verification (verify):
TBD

CAP:

```
mask = eclipse.pixelmap_binary_AND(hot, cold)

cube = []
for frame in raw_science_frames:
    frame = eclipse.trim_and_overscan(frame)
    frame = eclipse.image_sub(frame, bias)
    frame = eclipse.image_div(frame, flat)
    stats = eclipse.iter_stat(frame)
    frame = eclipse.image_div(frame, stats.median)
    cube.append(frame)

fringemap = eclipse.cube_avg_median(fringemap)
stats = eclipse.stat(fringemap, pixmap=mask)
fringemap = eclipse.image_div(fringemap, stats.avg_pix)
fringemap = eclipse.image_sub(fringemap, 1.0)
fringemap = eclipse.image_mul(fringemap, mask)
```