Req 6.3.3

Title:

Construct individual weight maps.

Objective:

Construct individual weight maps.

In addition to the effects of hot and cold pixels, individual images are contaminated by saturated pixels, cosmic ray events, and may be affected by satellite tracks. For purposes of subsequent analysis and image combination, pixels affected by these effects need to be assigned a weight of zero in weight maps that are unique to each image.

Saturated pixels are pixels whose counts exceed a certain threshold. In addition, saturation of a pixel may lead to 'dead' neighbouring pixels, whose counts lie below a lower threshold.

Cosmic-ray events can be detected using special source detection filters (retina filters), with Sextractor. These are essentially neural networks, trained to recognize cosmic rays, taking a set of neighbouring pixels as input (see Section 7.1 for further details).

Satellite tracks can be discovered by a line-detection algorithm such as the Hough transform, where significant signal along a line produces a 'peak' in the transformed image. This peak can be clipped, and transformed back into a pixelmap that masks the track (see Section 7.1 for further details).

Since the variance is inversely proportional to the Gain, which is proportional to the flatfield, the weight is given by:

$$W_{ij} = G_{ij} P_{hot} P_{cold} P_{saturated} P_{cosmic} P_{satellite}$$

where P are binary maps where good pixels have a value of 1 and bad pixels have a value of 0.

To take into account the illumination correction, as described in **req.**548, the weight should be divided by the illumination image.

Inputs:

SeqFile– 631 *Raw science frame* (for saturated pixelmap)
SeqFile– 632 *Reduced science frame* (for cosmic and satellite)
CalFile– 522 *Hot pixel map*
CalFile– 535 *Cold pixel map*
CalFile– 546 *Master flatfield*
CalFile– 548 *Illumination correction*

Outputs:

SeqFile– 633A *Saturated pixel map*
SeqFile– 633C *Cosmic pixel map*
SeqFile– 633S *Satellite track map*
SeqFile– 633 *Weight image*

Estimated time needed:

Approximately 1 min/CCD

Recipe:

This is covered by **Recipe– Reduce** which is documented in **seq.– 632**.

Needed functionality:

image - cosmic ray detection; image - mask; image - satellite detection; image - arithmetic; image - statistics

CA:

Saturated Pixel Map:

Process (make):

1. Create a pixelmap by applying a threshold to the raw data

Cosmic Map:

Process (make):

1. Construct a weight map from the flatfield, saturated, hot and cold pixelmaps.
2. Subtract the background from the reduced science image.
3. Multiply by the square-root of the weight to produce a SNR image.
4. Run Sextractor with a cosmic retina to produce a segmentation image and a catalogue of of cosmic-ray detections.
5. Use the segmentation image to produce a cosmic pixelmap.
6. Use the catalogue to count the number of cosmic ray events.

Satellite Map:

Process (make):

1. Construct a weight map from the flatfield, saturated, hot and cold pixelmaps.
2. Subtract the background from the reduced science image.
3. Multiply by the square-root of the weight to produce a SNR image.
4. Create a Hough-transformed image for pixels $> 5\sigma$.

5. Create a pixelmap by inverting the clipped Hough image.

Weight Image:

Process (make):

1. Combine the hot, cold, saturated, cosmic and satellite pixelmaps (logical AND).
2. Multiply the combined pixelmap with the master flatfield.
3. Divide the weight by the illumination correction.

CAP:

Saturated Pixel Map:

```
saturated = eclipse.threshold2pixelmap(raw, THRESHOLD_LOW, THRESHOLD_HIGH)
```

Cosmic Pixel Map:

```
Constants:
  COSMIC_CONFIG : SExtractor configuration parameters and their
values
                in case they differ from the SExtractor defaults.
          CHECKIMAGE_TYPE = 'SEGMENTATION'
          ANALYSIS_THRESH = 3.0
          CATALOG_TYPE    = 'ASCII'
          DETECT_MINAREA  = 1
          DETECT_THRESH   = 5.0
          GAIN            = 1
          MEMORY_PIXSTACK = 100000
          BACKPHOTO_THICK = 24
          BACK_TYPE       = 'AUTO'
          BACK_VALUE      = 0.0, 0.0
          FILTER          = 'Y'
          FILTER_NAME     = 'cosmic.ret'    # Retina filter for
SExtractor
          CHECKIMAGE_NAME = 'cosmic.fits'   # Image with cosmic
ray events

mask   = eclipse.pixelmap_2_image(hot & cold & saturated)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
```

```
eclipse.image_pow_local(weight, 0.5)
snr_image = eclipse.image_mul(reduced, weight)

sextractor(snr_image, COSMIC_CONFIG)

cosmic = eclipse.threshold2pixelmap(cosmic, -0.5, 0.5)
```

Satellite Pixel Map:

```
mask   = eclipse.pixelmap_2_image(hot & cold & saturated)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
eclipse.image_pow_local(weight, 0.5)
snr_image = eclipse.image_mul(reduced, weight)

stat = eclipse.stat_iter(snr_image)

hough = eclipse.hough_transform(snr_image,
                                stat.stdev*DETECTION_THRESHOLD)

satellite = eclipse.inverse_hough_transform(hough, HOUGH_THRESHOLD)
```

Weight image

```
mask   = eclipse.pixelmap_2_image(hot & cold & saturated & cos-
mic & satellite)
weight = eclipse.image_mul(flat, mask)
weight = eclipse.image_div_local(weight, illumination)
```