Quality Control Tools

Astro-Wise Group, INAF-OAC

October 20, 2003

All the tools described in this document are implemented in C as an extension of the eclipse library (using image structure) and can be run in python scripts. All of them calculate a quality index (QI) which, at the end, is compared with a threshold in order to assign a binary value (0 or 1) to a quality flag (QF):
if QI > threshold $\Longrightarrow$ QF = 0 (means bad frame)
if QI < threshold $\Longrightarrow$ QF = 1 (means good frame)

1 SignTest

1.1 Description

The signTest algorithm is applied to bias frames in order to detect possible structures. In output it creates a quality index (defined below) and a flag image. The algorithm splits the input frame in a number n of subwindows (n=$n\_windows\_x \times n\_windows\_y$) and then counts, for each subwindow, the excess of elements greater than the median (respect to those smaller); if this number is not compatible with the binomial distribution, the subwindow is flagged. Finally, the number of bad windows is compared with the theoretical number of bad windows expected (following the Gaussian distribution at a fixed number of sigmas) and the quality index is calculated as:
$QI = \frac{N^{\circ}BAD - N^{\circ}EXPECTED}{\sigma}$

Input

- Image (input frame)
- n_windows_x (number of subwindows in x)
- n_windows_y (number of subwindows in y)
- n_sigma (number of sigmas)
- threshold

Algorithm

- Split the frame in ($n\_windows\_x * n\_windows\_y$) windows
- $\forall$ windows:
  $npos = \#elements > median$
  $nneg = \#elements < median$

- IF $ABS(npos - nneg) > 25$ THEN

  $LOWER = \frac{npos+nneg}{2} - threshold * \sqrt{\frac{npos+nneg}{4}}$

  ELSE

  $UPPER = \frac{npos+nneg}{2} + threshold * \sqrt{\frac{npos+nneg}{4}}$

- IF $npos < LOWER$ OR $npos > UPPER$ THEN

  This window is FLAGGED as BAD

Output

- Quality Flag
- FLAG Image [fits file with $(n\_windows\_x * n\_windows\_y)$ elements] showing the bad windows

1.2 Tests

The algorithm has been tested on bias images. It works fine with the laboratory OmegaCAM bias frames. On the other hand, with worst quality bias such as the WFI ones, it is even too sensitive; in this case it can be applied to the difference between one bias and another (raw minus master or raw minus reference).

2 QC DomeFlat

2.1 Description

The purpose of this tool is to measure the flatness of the dome flatfields and reject those having flatness larger than a fixed threshold (e.g. 1%). First the image is normalised (divided by its median value). Then the medians of a number of sub-windows of the normalized image are calculated. Finally, the minimum median is subtracted from the maximum median and the result, which represents the relative flatness, is compared with the given threshold.

Input

- Image (input frame)
- n_windows_x (number of subwindows in x)
- n_windows_y (number of subwindows in y)
- threshold

Algorithm

- Normalize image by its median
- Split the frame in $(n\_windows\_x * n\_windows\_y)$ windows
- $\forall$ window calculate $median_i$

- Max is the Maximum value of medians
- Min is the Minimum value of medians

Output

- Quality Flag
- Median Image [fits file with ($n\_windows\_x * n\_windows\_y$) elements] reporting the value of the median of each window

2.2 Tests

This algorithm has been tested on 96 WFI V-band dome flats: 48 were correctly flatfielded (i.e. divided by the V-band master dome flatfield) and 48 were divided, as a test, by the master dome of the B band. In the first case the result is close to zero (of the order of 10^{-4}); instead, in the second case, the value is always greater than 0.01 (see Figure 1).

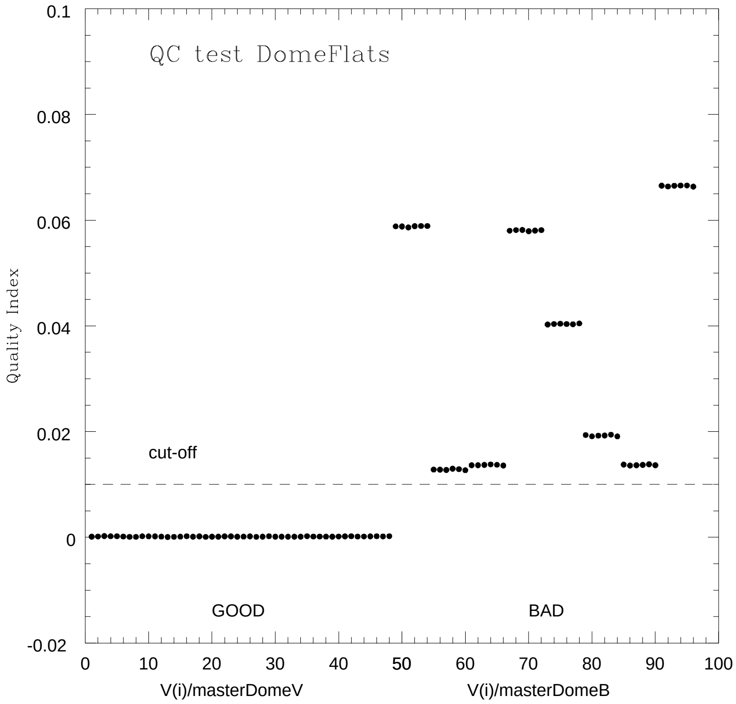

Figure 1: results of the QC Domeflat test. Left: good quality dome flats obtained dividing 48 WFI V-band dome flats by their master. Right: bad quality dome flats obtained dividing 48 WFI V-band dome flats by the B-band master; each sequence of 6 points corresponds to a different CCD.

3 QC Twilightflat

3.1 Description

The purpose of this tool is to measure the flatness of the twilight flatfields and reject those having flatness larger than a fixed threshold (e.g. 1%). In order to exclude the contribution of the stars (which can be present in the raw twilight flatfields), the flatness is calculated on a bicubic spline interpolation of the original frame (using the imsurfit tool). Then the procedure is exactely the same as in QC Domeflat.

Input

- Image (input frame)
- n_windows_x (number of subwindows in x)
- n_windows_y (number of subwindows in y)
- threshold

Algorithm

- Make a Bicubic Spline Interpolation on input frame
- Split the surfit frame in $(n\_windows\_x * n\_windows\_y)$ windows
- $\forall$ windows compute the median

Output

- Quality Flag
- Median Image [fits file with $(n\_windows\_x * n\_windows\_y)$ elements] reporting the value of the median of each window

3.2 Tests

This algorithm has been tested on 88 WFI V-band twilight flats: 48 were correctly flatfielded (i.e. divided by the V-band master twilight flatfield) and 40 were divided, as a test, by the master twilight of the B band. In the first case the result is close to zero (of the order of 10^{-4}); instead, in the second case, the value is always greater than 0.005 (see Figure 2).

4 QC Outlayers

4.1 Description

This tool counts the number of pixels having a value larger than a fixed number in absolute value, called as maximum_number. It can be used for calibration frames, in particular for dome and twilight flatfields, in parallel with the QC Dome and QC Twilight tools (which measure only the flatness).

Input

- Image (input frame)

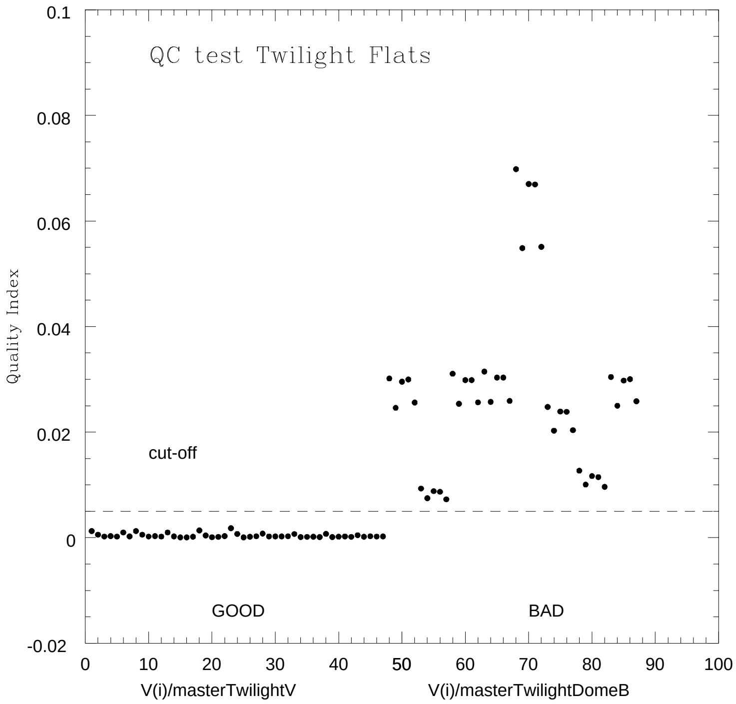

Figure 2: results of the QC Twilight test. Left: good quality twilight flats obtained dividing 48 WFI V-band twilight flats by their master. Right: bad quality twilight flats obtained dividing 40 WFI V-band twilight flats by the B-band master; each sequence of 5 points corresponds to a different CCD.

- Image (Bad Pixel Mask)
- maximum_number
- threshold

Algorithm

- $\forall$ pixels

  IF $(pixel_i > maximum\_number$ or $pixel_i < -maximum\_number)$
  and $(pixel_i\ is\ not\ flagged\ in\ the\ BadPixelMask)$ THEN
  Flag this pixel and count it

Output

- Quality Flag
- FLAG Image [fits file with same dimensions of the input frame]

4.2 Test

The algorithm has been tested in a few images and works correctly.