

Req 6.3.4

Title:

Astrometry

Objective:

Determine the astrometric solution for the science image. Also determine the seeing.

Produce a catalog of the science image. Use the catalog to determine the astrometric solution and the seeing, and record the solution in the header of the image. The astrometrically calibrated image is used when building the dedithered image (**seq.– 636**).

Next to the basic scientific requirement of an accuracy of the absolute astrometry better than 0.1 arcsec rms, an upper limit on the astrometric uncertainties is determined by the requirement that astrometric inaccuracies do not degrade the seeing of the dedithered image (**seq.– 636**).

The full astrometric process solves simultaneously the focal plane assembly parameters and the telescope light-path parameters. A series of these solutions should result in a definition of the telescopes current focal plane structure, which in normal circumstances should be constant, and a definition of the light path deformation, which in normal circumstances can be variable. The empirical determination of the full telescope plate deformation can be used as a-priori knowledge in the pairing phase for another determination.

The default astrometric solution is obtained on a 'per chip' basis. Alternatively, one can choose to use the overlap that results from a repeated observation of the same area of sky in a jittered or dithered observation to improve on this solution. Improved accuracy is obtained if one uses the overlap information for a single chip. Maximum accuracy is obtained if one also uses the overlap between different chips in dithered observations. The latter method breaks the inherent parallelism of processing individual chips.

The US-NAVAL Observatory A2.0 catalog provides astrometric reference objects with a position accuracy of 0.3 arcsec rms. Since there are > 10 reference objects per CCD, even the default astrometric solution will give a position error $< 0.3/\sqrt{10} < 0.1$ arcsec rms.

The large CCD's are created by stitching smaller masks together. Between the stitches errors in the position of pixels with respect to a truly regular grid may occur. The method described above may be applied to individual stitches for each CCD, therefore increasing the number of free parameters of the solution.

The small stitches are generally too small to yield enough pairs for high resolution solutions, but an ensemble of observations would. Determination of the stitching topology is a do-once operation.

Preliminary inspection of **OmegaCAM** CCDs has shown that the position error introduced by stitching is approximately 0.02 arcsec. Correction for stitching is therefore currently not considered.

Inputs:

SeqFile– 632 *Reduced science frame*

SeqFile– 633 *Individual weight image*

CalFile– in1 *US-NAVAL Observatory A2.0*

Outputs:

SeqFile– 634 *Astrometrically calibrated science image with astrometry solution inserted into the descriptors*

Astrometric quality parameters

Seeing **SeqFile– 634C** *Astrometrically calibrated catalog - not further used*

SeqFile– 634R *Astrometric residual map - not further used*

Estimated time needed:

Approximately 30 sec./CCD.

Recipe:

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

Needed functionality:

catalog - source extraction; catalog - association; catalog - astrometric solution; catalog - seeing;

CA:

Process (make):

1. Run `sextractor` to obtain input catalog.
2. Run `LDAC-prestrom` to produce preliminary calibrated catalog and preliminary local reference catalog.
3. Run `LDAC-aplastrom`, `LDAC-associate` and `LDAC-make_ssc` to create a pairs catalog of the preliminary calibrated catalog and the local reference catalog.
4. Run `LDAC.astrom` to determine the full astrometric solution from the preliminary calibrated catalog and the pairs catalog.
5. Apply the astrometric calibration and determine the `wcs`-parameters that describe the distortion.
6. Select stellar objects (no flag, bright but not too bright, and approximately the right size) from the catalog.

7. Use the median of the half-light radius (FLUX_RADIUS) as a measure of the seeing.

CAP:

Constants:

APLASTROM_CONFIG : LDAC.aplastrom configuration parameters and their values

do NOT differ from the LDAC defaults.

ASSOCIATE_CONFIG : LDAC.associate configuration parameters and their values

in case they differ from the LDAC defaults.

INTER_COLOR_TOL = 2.5

ISO_COLOR_TOL = 1.5

PAIR_COLS = 'NO'

ASTROM_CONFIG : LDAC.astrom configuration parameters and their values

in case they differ from the LDAC defaults.

NITER = 5

XMIN = 0.0

XMAX = 32767.0

YMIN = 0.0

YMAX = 32767.0

MAKE_DISTORT_CONFIG : LDAC.make_distort configuration parameters and their

values do NOT differ from the LDAC defaults.

MAKE_SSC_CONFIG : LDAC.make_ssc configuration parameters and their values

in case they differ from the LDAC defaults.

MAKE_PAIRS = 'YES'

PREASTROM_CONFIG : LDAC.preastrom configuration parameters and their values

in case they differ from the LDAC defaults.

MAX_OFFSET = 150.0

PHOT = 'FLUX_ISO'

RMS_TOL = 2.57

XMIN = -32768.0

YMIN = -32768.0

SEXTRACTOR_CONFIG : SExtractor configuration parameters and their values

in case they differ from the SExtractor defaults.

```
ANALYSIS_THRESH = 3.0
BACKPHOTO_THICK = 24
BACK_TYPE       = 'AUTO'
BACK_VALUE      = 0.0, 0.0
GAIN            = 1
MEMORY_PIXSTACK = 100000
FILTER          = 'N'
WEIGHT_TYPE     = 'MAP_WEIGHT'
```

```
incat = sextractor(image, SEXTRACTOR_CONF)
```

```
LDAC.prestrom(incat, outcat1, inref, outref1, PREASTROM_CONFIG)
```

```
LDAC.aplastrom(outcat1, outcat2)
```

```
LDAC.associate([outcat2, outref1], [outcat3, outref2], ASSOCIATE_CONFIG)
```

```
LDAC.make_ssc([outcat3, outref2], outref3, MAKE_SSC_CONFIG)
```

```
LDAC.astrom(outcat1, astcat1, outref3, ASTROM_CONFIG)
```

```
LDAC.aplastrom(astcat1, astcat2)
```

```
LDAC.make_distort(astcat2, astcat3)
```

```
wcs-parameters = LDAC.get_records(astcat3, table='DISTORT')
```

```
# determine seeing
```

```
data = LDAC.get_records(atscat3, table='OBJECTS')
```

```
stellar_radii = []
```

```
for object in data:
```

```
    if (object['Flag'] == 0 and
        peak_min < object['MaxVal'] < peak_max and
        radius_min < object['FLUX_RADIUS'] < radius_max):
        stellar_radii.append(object['FLUX_RADIUS'])
```

```
SEEING = median(stellar_radii)
```