Malcolm J. Currie
Brad Cavanagh
Joint Astronomy Centre, Hilo, Hawaii

2004 June

ORAC-DR – imaging data reduction 4.1
User Guide

Abstract

ORAC-DR is a general-purpose automatic data-reduction pipeline environment. This document describes its use to reduce imaging data collected at the United Kingdom Infrared Telescope (UKIRT) with the UFTI, UIST, IRCAM, and Michelle instruments; at the Anglo-Australian Telescope (AAT) with the IRIS2 instrument; at the Very Large Telescope with ISAAC and NACO; from Magellan's Classic Cam, at Gemini with NIRI, and from the Isaac Newton Group using INGRID. It outlines the algorithms used and how to make minor modifications to them, and how to correct for errors made at the telescope.

Contents

1 Introduction

ORAC-DR is a data-reduction pipeline operating at UKIRT, JCMT, the AAT. It is part of the ORAC system. The pipeline reduces and displays multi-frame observations soon after they are read from the detector. This allows observers to assess the quality and suitability of their data in near real time. Yet ORAC-DR is capable of producing publication-quality results.

ORAC-DR is suitable for 'offline' data reduction at your home institution too. There are many reasons why you may wish to use ORAC-DR in this fashion. For instance, you may have come back from UKIRT with only the raw observations; or there was an error in a telescope sequence (formerly an 'exec') mixing the groups of observations; or some data were reduced with a basic algorithm for speed at the telescope, and now you want to do a more-careful job. ORAC-DR is capable of reducing data from instruments not running the pipeline at their respective telescopes. Hence ORAC-DR is available on Starlink.

SUN/230 presents an overview of ORAC-DR, general facilities like its display system, and it explains the differences between a pipeline and a traditional reduction package. Put briefly, ORAC-DR uses a few data headers to direct the data reduction. Amongst these headers is the name of a *recipe*. A recipe is a series of high-level instructions such as "make a mosaic" or "divide by a flat" that reduces an *observation* comprising one or more data frames. The implementation of each of these instructions is through a Perl script—called a primitive—which calls Starlink packages such as CCDPACK and KAPPA, to actually do the processing of the bulk data.

This document describes how to use ORAC-DR software on Starlink to reduce data from the UKIRT imaging instruments: UFTI, UIST, IRCAM, and Michelle; the AAT imaging instrument IRIS2; the ISAAC and NACO instruments on the Very Large Telescope (VLT); Classic Cam from Magellan; NIRI from Gemini, and INGRID from the Isaac Newton Group on La Palma. It outlines the various algorithms used in the recipes, and includes detailed recipe documentation in the appendix. Besides the standard reduction recipes, this manual describes how you can customise recipes to suit your preferences, and how to correct errors in the headers of your data frames.

There are complementary documents: SUN/236 describes the ORAC-DR for spectroscopy from CGS4, Michelle, and UIST; SUN/246 describes the ORAC-DR for integral field spectroscopy from UIST; and SUN/231 addresses the reduction of SCUBA data with ORAC-DR.

Those wishing wishing to write their own recipes from scratch, or wanting to apply ORAC-DR to new instruments should consult SUN/233.

2 Using the pipeline

2.1 Setting up ORAC-DR

Before you can run the pipeline you have to tell ORAC-DR for which instrument you wish to reduce data, the observation date, and the directory containing the raw data, and wher you want the processed data to be written. There are two options.

- The first needs your data to conform to the directory-naming convention of the instrument at UKIRT. This will be the case if you simply unpack the archive written by the **uktape** utility. In this case enter

  ```
  % setenv ORAC_DATA_ROOT <root_data_directory>
  % oracdr_<instrument> <date>
  ```

  where <root_data_directory> is the directory in which you unpacked the data from the tape, <instrument> is either ufti or ircam, and <date> is the UT date in the format YYYYMMDD. Note that each % represents the UNIX shell's prompt, which you do not type. The commands must be entered in the above order.

  For example, the standard location for raw UFTI data is raw/ufti/YYYYMMDD/, and reduced/ufti/YYYYMMDD, for the corresponding reduced data. So if your data are stored in /home/users/abc/data/UKIRT/raw/ufti/2 you should enter the following.

  ```
  % setenv ORAC_DATA_ROOT  /home/users/abc/data/UKIRT/
  % oracdr_ufti 20001108
  ```

  to enable the pipeline for UFTI data taken on 2000 November 8.

  Data taken from the AAT is handled differently, as there is no unified directory structure for both raw and reduced data directories. For IRIS2, INGRID, ISAAC, NACO, NIRI, or Classic Cam data the best option is specifying where the raw and reduced data directories are, as shown below. Those with ISAAC and NACO data should see

  Section 2.2.1 for a necessary preliminary naming conversion step for each instrument. Classic Cam users need to read

  Section 2.2.2 concerning renaming raw data files for the respective instruments.

- The second option is where your raw and reduced data are to be in arbitrary directories. Type the following

  ```
  % oracdr_<instrument> <date>
  % setenv ORAC_DATA_IN <raw_data_directory>
  % setenv ORAC_DATA_OUT <reduced_data_directory>
  ```

  The directories can either be given as full paths, or as relative paths to the current working directory. Here is an example for IRCAM data using full paths.

  ```
  % oracdr_ircam 19990328
  % setenv ORAC_DATA_IN /export/data/mjc/asteroid/night1
  % setenv ORAC_DATA_OUT /home/scratch/mjc/reduced
  ```

In the first case $ORAC_DATA_IN and $ORAC_DATA_OUT are still defined, but in terms of the root directory. For instance, re-using the earlier example with UFTI for UT date 2000 November 8, $ORAC_DATA_IN points to $ORAC_DATA_ROOT/raw/ufti/20001108/.

ORAC-DR operates in $ORAC_DATA_OUT, irrespective of what your current directory is when you invoke it. Your current directory remains unchanged.

It is highly recommended to work in directories on discs local to the computer running the pipeline. Processing over NFS-served drives can many times slower and degrades the performance seen by other users. Running ORAC-DR on a Linux computer over NFS-served drives can also lead to erroneous results, crashing of the pipeline, or computer lockups.

2.2   Raw Data Formats and Conversions

Raw data take the form of multiple NDFs within an HDS container file for UKIRT data, or individual FITS files for AAT, INGRID, ISAAC, NACO, NIRI, and Classic Cam data. For UFTI, UIST, and IRCAM they comprise one NDF for the data array and dynamic headers, such as the start time of the exposure, and another for static headers. Each container file is converted to a single NDF in $ORAC_DATA_OUT with a merged set of headers.

The Michelle HDS container file also has NDFs for the individual chop beams. However, these cannot be merged until the data variance is calculated from the individual beams. Michelle reduced chopped data become simple NDFs once the recipe takes the difference of the two beams.

ORAC-DR automatically converts AAT FITS files into single NDFs in $ORAC_DATA_OUT which retain the original FITS headers. For INGRID and NIRI, ORAC-DR converts a multi-extension FITS file into a multi-NDF HDS container file following UKIRT conventions.

2.2.1   ISAAC and NACO Preliminary Conversion

Since ORAC-DR as yet cannot cope with ESO file naming, which uses the UT epoch instead of a sequence number, there is a special C-shell script which must be invoked once, normally before the first ORAC-DR initialisation. If you enter

```
% isaac2oracdr
```

in a directory containing ISAAC FITS files, the command converts them into NDFs with names adhering to the UKIRT convention. The prefix is `isaac`. The earliest file has observation number 1, and the observation number increments for each FITS file in time order. The script copes with files names in either the raw or archive nomenclature. It also writes observation and group number headers to assist ORAC-DR. It copes with data from more than one night in a given directory, assigning each night its own sequence of observation numbers; and it uses a common UT date for observations in a single night spanning midnight UT. You should put all both the calibration and target files for a given night in the same directory.

Likewise in a directory of NACO FITS files, you should first enter

```
% naco2oracdr
```

to create a set of NDF files whose names adhere to the UKIRT convention with a `naco` prefix.

2.2.2   Classic Cam Preliminary Conversion

The Magellan Classic Cam raw FITS data have a sequence number but no UT date in their names. There is a preprocessor C-shell script which must be invoked once, normally before the first ORAC-DR initialisation. If you enter

```
% cc2oracdr
```

in a directory containing Classic Cam FITS files, the command converts them into NDFs with names adhering to the UKIRT convention. The prefix is `cc`. The earliest file has observation number 100 (a Classic Cam convention), and the observation number increments for each FITS file in sequence-number order.

It also writes the observation and group numbers, and the number of offsets in the grouped observation into the headers, to assist ORAC-DR.

2.3   Running the pipeline

To run the pipeline, you use the **oracdr** command. This has a number of qualifiers described fully in SUN/230. There is online help too; enter

```
% oracdr -h
```

for a list of the options.

Unlike using ORAC-DR at UKIRT, you are unlikely to need the looping (`-loop` option) for offline processing, as all the data exist. Thus the most important qualifiers are `-list` and `-from`, which specify the frames to process; and the recipe name.

```
% oracdr -from 42
```

will process frames f20001108_00042 until the end of the night's data (assuming the earlier **oracdr_ufti** command), running the recipes given by each frame's header (RECIPE keyword). More likely is that you provide a list of selected observations. The following example

```
% oracdr -list 41:49,51:59 JITTER_SELF_FLAT
```

processes frames from 41 to 49 inclusive and 51 to 59 inclusive, invoking the JITTER_SELF_FLAT recipe, and overriding the RECIPE header.

```
% oracdr -list 5:7,23,33
```

would reduce the frames 5, 6, 7, 23, and 33. This is most likely to be applicable to a series of dark frames.

There is a hazard with the `-list` option. Take care to select a complete set of frames associated with an observation. A common error is to include accidently a dark frame not part of the group. Check the log in the raw data directory; it has file extension `.nightlog`. If you do not have a log, it is easy to create one.

```
% oracdr -noeng -from 1 -skip -nodisplay NIGHT_LOG
```

This will create a log called $ORAC_DATA_IN/<date>.nightlog for the current UT date. For multi-mode instruments such as Michelle, UIST, IRIS2, NACO or ISAAC, the log will be called

`$ORAC_DATA_IN/<date>_im.nightlog`.

2.4   Graphical initialisation and operation

You may prefer the ORAC-DR graphical interface called **xoracdr**. (See SUN/230.) It allows you to configure ORAC-DR: set the instrument, UT date, raw and reduced directories; and to run the pipeline with the various options. It permits monitoring of the primitives during execution of a recipe. **xoracdr** offers access to other facilities like display control and recipe editing. The in-built documentation does not pertain to the GUI itself but to general ORAC-DR information, however, **xoracdr** is straightforward to use and explore. While **xoracdr** has some rough edges, it is popular with many users. To try it, enter

```
% xoracdr &
```

Once the tool appears, you should select an instrument from the menu on the left, a UT date in the top centre, and raw and reduced directories to the lower right. The `From:` and `To:` refer to the observation numbers to process. When you are ready to reduce data, click on the `Start ORAC-DR` button.

2.5   Display

ORAC-DR optionally lets you inspect the raw frames, and the processed data as they are created. There is a variety of graphical methods available, including histograms and contour plots, if you choose a KAPPA GWM widget. Most people prefer a simple scaled image display with GAIA. This offers facilities to inspect and analyse the data, and both pixel and sky co-ordinates of the cursor position are presented. The selection of frame types to display, where they should appear, and how they are scaled are configurable using a simple text file or a special GUI tool **oracdisp**. See SUN/230 for details and examples.

Processing offline, there is less need to see the data displayed in real time. If you wish to accelerate the processing switch off the display option.

```
% oracdr -nodisplay ...
```

If you do want to display a recommendation is to create two GAIA windows displaying images using autoscaled limits. This first could be for raw and flat-fielded data, and the second for the mosaics. You are likely to want to interact with the latter, using GAIA's toolboxes. Your `$ORAC_DATA_OUT/disp.dat` could look like this.

```
# Send raw frame to first GAIA window
num type=image tool=gaia region=0 window=0 autoscale=1 zautoscale=1

# Send flatfielded frame to first GAIA window.
ff  type=image tool=gaia region=0 window=0 autoscale=1 zautoscale=1

# Send mosaic frame to second GAIA window.
mos type=image tool=gaia region=0 window=1 autoscale=1 zautoscale=1
```

2.6 Calibration Information

ORAC-DR records calibration information, such as dark frames, flat fields, and the read noise, within index files, one for each type of calibration information. When the pipeline needs a calibration frame it searches the index file for the best matching entry subject to a set of rules. Each recipe reports the calibrations it has used. If no suitable calibration exists, the pipeline exits with an error message stating this fact. For further details see SUN/230.

Section 4.3 has an example of an index file.

You can also select a specific calibration using the `-calib` command-line option, provided the chosen calibration has an entry in the appropriate index file. See the section on calibration options in SUN/230 for details and examples.

2.6.1 Available calibration methods

The following calibration methods are available for imaging recipes.

- baseshift — Use the given comma-separated doublet (*i.e.* 0,0) as the frame's base position. This is used to locate faint sources in the mid-infra-red data where centroiding fails when there is some telescope pointing error (such as incorrect instrument apertures). It is calibrated within NOD_CHOP_APHOT on a bright standard, and used by NOD_CHOP_FAINT. For a well-tuned system, the baseshift is expected to be near 0,0, so the centre of the detector is at the reference position (derived from the FITS headers).

- bias — Use the given bias frame.

- dark — Use the given dark frame.

- fpcentre — Use the given Cartesian pixel centre of the Fabry-Perot transmitted region (UFTI only).

- flat — Use the given flat frame.

- mask — Use the given bad-pixel mask.

- polrefang — Use the specified polarimetry calibration angle, converting measured polarisation vector orientations into position angles. It is measured in degrees, applied clockwise to the vector direction to allow for the orientation of the analyser with respect to north.

- readnoise — Use the given value for the detector readnoise in electrons.

- referenceoffset — Use the comma-separated doublet (*i.e.* 0,0) as the frame's reference offset, which is difference between the frame centre and the reference pixel derived from the FITS headers.

- rotation — Use the given frame as a rotation matrix. This is no longer used in the imaging recipes.

- sky — Use the given sky frame.

2.7   Log files

In addition to presenting the progressing data reduction to an ORAC-DR X-window, ORAC-DR, by default, retains a copy of the processing steps and errors in a log file. These logs are important if something has gone wrong, and you have exited the X-window. Information from the applications software can be included if you run the pipeline with the `-verbose` command-line option. Logs also serve as a record of the data processing. Yet the log files are often overlooked because they are hidden. The log file is called `$ORAC_DATA_OUT/.oracdr_`<number>, where <number> is the current process identification. The `-log f` option to the **oracdr** command enables log-file creation.

See SUN/230 for details of the logging options.

3   Features of the Primitives

Primitives are the Perl scripts which actually call the applications to do most of the data processing. All of the imaging recipes are, in principle, independent of the instrument. However, some recipes are inappropriate; for example, the NOD_CHOP family of recipes are intended for mid-IR imaging with Michelle, but not the JITTER_SELF_FLAT family. The generality comes in part from translations of relevant header information into a generic form used by ORAC-DR.

Not all the following steps apply to all recipes. Consult the reference section to see summaries for each recipe. The steps are presented in normal order of appearance.

The main primitives pertinent to each step are listed in bracketed italics, should you wish to tailor the recipes. These are found in `$ORAC_DIR/imaging` tree, unless they start with *general*/ when they are located in `$ORAC_DIR/general`. Note the some may be instrument-specific variants, either given explicitly or with the <*instrument*> token, which means substitute the instrument name in uppercase.

While those listed form the bulk of the primitives, there are many not listed here, mostly those for

recipe initialisation called <*recipe*>_HELLO_ (see Section 4.2.1 for more information), and for

recipe steering which control when to perform certain operations, called <*recipe*>_STEER_ (see Section 4.2.2). The first of these is normally left unchanged unless there is a need to add more steering parameters passed to the steering primitive. The steering primitive itself is only modified for such new steering parameters, and if the observing pattern or sequence is different from what the recipe is programmed to expect. Other primitives not mentioned here are tied closely with single recipes, usually to create and file calibrations.

3.1   Preparation of Single Frames

[_IMAGING_HELLO_, <instrument>/_INSTRUMENT_HELLO_,
_PREPARE_SINGLE_FRAME_]

3.1.1 Manipulation of Raw Data

The first stage makes an NDF, or multiple NDFs stored in an HDS container file, located in `$ORAC_DATA_OUT`, the actual operation depending on the instrument. In some cases like UFTI, this is dealt by the ORAC-DR infrastructure, including conversion from FITS for early UFTI data.
[<instrument>/_CREATE_RAW_FRAME_]

For IRCAM the recipes are merely copying the raw NDF, because the original raw files are normally write protected.

UIST and Michelle both copy all the components of multi-NDF container file. These container files have a HEADER NDF of the observation-wide FITS headers, and NDFs of chopped beams or integrations. Recipes check that two integrations are present for a chopped observations or one integration for other modes, exiting with an error message if either test fails. A single integration NDF is merged with the HEADER NDF into a simple NDF.

3.1.2 Preliminaries

There are a few operations applied to all frames. First the raw frame may be displayed.
[general/_DISPLAY_FRAME_IMAGE_]

Recipes remove any AXIS and blank TITLE components; the latter to preserve the object name when NDFs are exported to FITS. Next they set the origin of the frame so that frame pixels retain the detector pixel indices. It then becomes possible to use a full-sized bad-pixel mask or flat field on any subset of a detector's pixel grid.
[_REMOVE_AXES_, _REMOVE_BLANK_TITLE_, _SET_ORIGIN_]

Then recipes determine the displacements of the reference pixel with respect to the centre of the frame and stores the displacements in the referenceoffset calibration system. The reference pixel is where a star would be placed for photometry or the centre of a chopped and nodded pattern.
[_SET_REFERENCE_PIXEL_OFFSET_, _GET_FRAME_CENTRE_]

The next step is to switch on history recording. It is recommended to leave this enabled, since it provides a record of the processing steps of your final mosaics. Otherwise the pipeline becomes something of a black box. Use the KAPPA command **hislist** to list the history records.
[_TURN_ON_HISTORY_]

For Michelle there is a validation check of the waveform used, comparing the waveform name given in the headers with other metadata, and recipes issue a warning if there is an inconsistency. Also the data range is validated to be between 25000–48000.
[MICHELLE/_CHECK_WAVEFORM_, MICHELLE/_VALIDATE_RAW_DATA_]

For UFTI there is data validation, such that a warning is issued if the clipped mean sky level is below 24 counts per second in *K* band and 32 counts per second in *H*.
[UFTI/_CHECK_SKY_COUNTS_, UFTI/_VALIDATE_RAW_DATA_]

For UIST data taken before 2002 December 2 and Michelle, raw data units are converted from ADU per second to the UKIRT standard of total ADU per exposure.
[<instrument>/_INSTRUMENT_HELLO_, UIST/_DATA_UNITS_TO_ADU_]

A night log is created or appended in `$ORAC_DATA_OUT` for each frame processed. This tabulates the main parameters of the observation having first corrected defective or undefined headers.
[_NIGHT_LOG_, UIST/_NIGHT_LOG_, <instrument>/_FIX_HEADERS_ for Classic Cam, IRCAM, MICHELLE, NIRI, and UFTI]

3.1.3   Non-linearity Correction

Detector non-linearity corrections are applied to Classic Cam, IRCAM, and INGRID. For IRCAM the correction is 3.3×10^{-6} times the square of the bias-subtracted signal. For INGRID the measured-to-actual counts is given by the expression $1.0 - 1.2247 \times 10^{-6} * M - 7.68045 \times 10^{-11} * M^2$, where M is the measured ADU count, although it is actually applied after the pre-exposure readout is subtracted from the post-exposure integration. For Classic Cam the correction is $1.625 \times 10^{-6} * (1 + \text{overhead}/\text{exposure\_time})$, where the overhead is $2 * \text{speed} *$ number of pre-exposure reads $+$ (speed $+$ readout time) $*$ number of post-exposure reads. The various parameters in this expression are either read from the headers or taken verbatim from an IRAF script cited in the user manual.
[CLASSICCAM/_CORRECT_NONLINEARITY_,
IRCAM/_CORRECT_NONLINEARITY_,
INGRID/_CORRECT_NONLINEARITY_PRE_POST_]

3.1.4   Electronic Ghosting

For ISAAC there is a correction applied for electronic ghosting. The ghosts consist "of an additional signal, which, on one row, is proportional to the sum of the intensity along this row and the row 512 rows away." according to the *ISAAC Data Reduction Guide*. Recipes combine the two halves adding the bottom half to the top and vice versa, collapse this image along rows, scale by the documented correction factor of 1.35E-5 to form the ghost flux per row, which is then subtracted from the original image.
[ISAAC/_REMOVE_ELECTRONIC_GHOSTING_]

3.1.5   Bad pixels

The recipes apply a predetermined bad-pixel mask with the aim of removing the bulk of 'hot' and 'cold' pixels. The approximate percentages of pixels masked for each instrument is as follows: Classic Cam1.1, INGRID 1.0, IRCAM 0.1, IRIS2 0.1, ISAAC 0.4, UFTI 0.7, and UIST 0.4.

The Michelle bad pixel mask is time dependent; initially 12 aberrant pixels were identified, but when the new detector returned on UKIRT in 2004, the mask merely ignored the top sixth of the array.

For the test data available, NACO had large numbers of very noisy pixels to the foot and sides of the detector, hence the default bad-pixel mask has 9.9% bad pixels. If your NACO data do not show this, you can form your own mask from a long-exposure dark frame, looking for the highly deviant pixels. See

the notes "Creating a bad-pixel mask" below for suggestions.
[_MASK_BAD_PIXELS_]

There are two problems. First, the pre-calculated mask only accounts for 95% of UFTI's problem pixels. The other 5% are occasionally deviant on timescales of days. The variability of IRCAM, Michelle, UIST, and IRIS2 bad pixels is unknown at the time of writing. In addition the bad-pixel masks have not been regularly monitored prior to 2000 August. The result is that non-physical values could appear in the processed data, some as extreme as -10^{-31} causing automatic registration and image display to go awry.

Therefore, after dark subtraction, recipes apply thresholding which flags non-physical values as bad, meaning undefined. This is just augmenting the bad-pixel mask, and no valid data are lost.

The upper limit is above the nominal saturation levels: 16000 for Classic Cam; 30000 for INGRID; 20000 for IRCAM in STARE or NDSTARE mode, and 33000 using the Deepwell; 200000 for IRIS2 and ISAAC; 100000 for Michelle; 4300 for NACO in FowlerNsamp and Double_RdRstRd modes, but 12400 for Uncorrelated reads except for the M' band where it is 28000; 17000 times the number of coadds for NIRI; 15000 for UFTI; and 20000 for UIST. The lower limit is the 2-, 3-, 3-σ-clipped mean, approximating to the mode, less five times the clipped standard deviation, σ. While a positive threshold looks attractive, small negative values, while appearing non-physical, can arise through noise. Therefore, to avoid a bias (mainly in the *J* band), a further constraint is that the lower limit lies in the range −100 to 1.
[_SUBTRACT_DARK_, <instrument>/_GET_SATURATION_LEVEL_]

Recipes issue warnings if the dark-subtracted frame's mode is negative, allowing for the error in the mode. It aborts with an error message if the modal dark-subtracted signal is more than one standard deviation negative. These states usually arise because of an aberrant dark.

Creating a bad-pixel mask   The easiest way to create your own bad-pixel mask for use with the calibration system, is to run the MAKE_BPM recipe on a long-exposure dark (at least 20 seconds integration). It is possible to change the symmetric σ-clipping bounds in the recipe (see primitive `_MAKE_BPM_BY_SIGMA_THRESHOLDING_`). You can tailor this primitive if you want more control, say to have asymmetric rejection or more sophisticated definitions.
[_MAKE_BPM_BY_SIGMA_THRESHOLDING_]

For better results, use the average of long dark frames taken across two or three nights. First, produce QUICK_LOOK versions of the long-exposure dark to flatten the NDF structure or convert the FITS file. Flag all pixels that are 5 standard deviations (σ) above and below the 3-σ clipped mean of the dark as "bad", then multiply the resulting frame by zero so that the resulting bad-pixel mask has data values of 0 and bad only. You can choose your own thresholds. Here is an example, using data from two nights of UFTI data and Starlink software.

```
% oracdr_ufti 20010101
% setenv ORAC_DATA_OUT 'pwd'
% oracdr -list 4:4 QUICK_LOOK -nodisplay
% oracdr_ufti 20010102
% setenv ORAC_DATA_OUT 'pwd'
% oracdr -list 4:4 QUICK_LOOK -nodisplay

% kappa
% add f20010101_00004_raw f20010102_00004_raw add_darks
% cmult add_darks 0.5 av_darks
% stats av_dark clip=3
% thresh av_darks av_darks_thresh -49 58 bad bad
% cmult av_darks_thresh 0 avbpm title=\"UFTI bpm, January 2001\"
```

In the above example the 3-σ clipped mean was 4.27 and the standard deviation was 10.727, resulting in −49 and 58 as the lower and upper thresholds.

Then you specify the bad-pixel mask on the command line.

```
% oracdr -calib mask=avbpm ...
```

UIST has its own slightly different formula; see DARK_AND_BPM for details.

[UIST/_FIND_BAD_PIXELS_, UIST/_FILE_BAD_PIXELS_, _FILE_MASK_]

3.1.6   Data Variance

There is optional data variance creation for all instruments. By default only the polarimetry and mid-infra-red NOD_CHOP family of recipes have variance processing enabled. To switch on variance calculations for the other recipes is a simple edit of the recipe. See Section 4.2.1 for instructions.

The initial variance is calculated as follows.

- First the readnoise variance for the instrument is applied. It is currently a constant for all pixels, which takes the gain, the number of exposures, and the number of array reads per exposure into account. For more than one read, n, the noise scales by the following factor,

$$\sqrt{(n(n+1))/(12(n-1))}$$

  which is derived from linear-regression theory. The read noise reduces by the factor $1/\sqrt{\text{number of exposures}}$.
  [_ADD_READNOISE_VARIANCE_]

  The raw read noise in units of electrons comes from the readnoise calibration (set within the ARRAY_TESTS) recipe, or should one not exist, ORAC-DR selects a representative default. The UFTI and IRCAM defaults come from the instrument Web pages such as

  `http://www.ukirt.hawaii.edu/instruments/ufti/PARAMETERS.html` and depend on speed, gain, and read type. Typical values are as follows: Classic Cam $40e^-$, INGRID $25e^-$, IRCAM $47e^-$, IRIS2 $15e^-$, ISAAC $18.5e^-$, NACO $53e^-$, NIRI $50e^-$, UFTI $26e^-$, and UIST $40e^-$. Michelle reductions merely use an estimate of $1000e^-$.
  [<instrument>/_GET_READNOISE_]

  The gain—the number of electrons per analogue-to-digital unit—comes from the data headers. If for some unusual reason, the header is absent, the recipe subsitutes suitable time-dependent defaults. The gain is $7.5e^-$ for Classic Cam, $4.1e^-$ for INGRID, $\sim 6e^-$ for UFTI and IRCAM, $5.2e^-$ for IRIS2, $4.6e^-$ for ISAAC, $500e^-$ for Michelle, $10e^-$ for NACO, $12.3e^-$ for NIRI, and $15e^-$ for UIST.
  [<instrument>/_GET_GAIN_]

- The second step is to add the Poisson variance. This simply adds the data array to the variance component taking into account the gain of the detector and the number of exposures.

  For read modes where the bias level is not removed, such as CHOP or STARE, a bias calibration frame must be subtracted first. If no suitable bias is found, the bias is deemed to be zero; ORAC-DR issues a warning that the variance is wrong, likely overestimated.

  The calculations also derive the ratio of the read noise to the Poisson noise, and reports the percentage of background-limited pixels, *i.e.* those where the Poisson noise is greater than the read noise.
  [_ADD_POISSON_VARIANCE_]

3.1.7   Bias subtraction

In most cases there is no bias to subtract. Recipes attempt to remove a bias frame only if the data have variance information and were taken using a non-ND mode (*i.e.* where the bias has not

already been subtracted in the instrument data system), whereupon a bias frame, if available, is subtracted; if, however, there is no bias calibration the recipe issues a warning issued that the computed variances may be wrong. This step occurs between the creation of the readnoise variance and adding the Poisson variance.
[_REMOVE_BIAS_]

3.1.8   Bias creation

The Michelle ARRAY_TESTS recipe averages two bias frames, sets the observation type to BIAS for the result, and files the mean bias frame in the calibration system. The filed frame includes the observation number in its name.
[MICHELLE/_ARRAY_TEST_STATISTICS_, _FILE_BIAS_]

Likewise the UIST ARRAY_TESTS recipe averages a group of bias frames and sets its variance to a population variance estimate. The mean bias frame filed with the calibration system includes the group number in its name.
[UIST/_BIAS_GROUP_]

3.1.9   Chopping

In the thermal and mid-infra-red regimes the sky is varying so rapidly normal reduction methods are inappropriate. Instead sky subtraction is achieved either by frequently oscillating the secondary mirror between two beams (mid-infra-red), called A and B; or moving the telescope offsets (thermal) after a short exposure. The generic term is *chopping*. The former are reduced by the NOD_CHOP recipes, and the latter by the NOD_SELF_FLAT_NO_MASK recipes.

Both methods produce frames with the target image at different positions on the detector. The aforementioned recipes difference these pairs of frames, so that the result has both a positive and negative image, and a background level close to zero. The sense of the subtraction is always the same. ORAC-DR subtracts the B beam from the A beam, and the normal sequence is ABBA. For the thermal data, the chopped beam is only notional, but the same terminology and subtraction sense is used.
[_DIFFERENCE_PAIR_, _DIFFERENCE_PAIR_SIMPLE_,
_DIFFERENCE_CHOP_BEAMS_]

If the telescope is further offset (nodded), the final mosaic of the differenced frames can have two positive and two negative representations of the source. In practice the thermal reductions register and co-add the nodded frames to compensate for flat-field errors in IRCAM.

3.1.10   Post-pre subtraction

Raw INGRID data comprise a pre-exposure image and a post-exposure integration. The pipeline subtracts the former from the latter to give the measured signal.
[INGRID/_DIFFERENCE_PRE_POST_]

3.1.11   FITS headers

For historical reasons, IRCAM headers prior to ORAC were somewhat jumbled, disorganised, and some violated the FITS standard. The pipeline corrects, orders and structures the headers to help the human reader locate information quickly, and allow complete conversion to FITS.

For IRCAM, Michelle, and UFTI, even since ORAC came online, the raw headers do not provide
a sky co-ordinate system. IRIS2 also has an incomplete sky co-ordinate system supplied in the
raw headers. Using information in the raw headers, the pipeline creates a FITS world co-ordinate
system using a tangent plane projection in the AIPS convention—this is quite adequate given
the small fields of view—which it imports into the NDF's WCS component. Thus GAIA and
KAPPA can display these co-ordinates on overlay grids and axes. For all these, such as KAPPA's
display with axes, you may need to select the appropriate astrometric Frame like this.

```
% wcsframe <your_NDF> frame=sky
```

where you substitute your NDF's name for <your_NDF>. The mosaics from the pipeline should
already have the sky domain set. INGRID headers are also revised by the pipeline into an AIPS
system.
[<instrument>/_CREATE_WCS_, <instrument>/_GET_PLATE_SCALE_]

Note that the reference pixel of the equatorial co-ordinates in the raw headers was until recently
only known to a few arcseconds, but now should be of the order of 0.5 arcseconds. The pipeline
sets empirical reference pixels now matched by the telescope control system; see primitive
_CREATE_WCS_ for details. For critical work, you should tie in your frames with online
catalogues, as available through GAIA.

3.2   Dark subtraction

This is as simple as it sounds. The dark frame selected through the calibration system must
have the same exposure time and read type as the object frames. There is no dark subtraction
for chopped data as processed by the NOD_CHOP collection of recipes, as the differencing of
nodded pairs of frames makes the operation unnecessary.
[_SUBTRACT_DARK_, _GET_DARK_NAME_]

3.2.1   Dark creation

After the preliminary steps including addition of variance, the dark is merely filed in the
calibration system, using the frame number and the exposure time in the name.

When multiple darks of the same exposure time form part of the same grouped observation,
these darks are averaged before filing with the calibration system. Such averaged darks have
the group number instead of the observation number in their names.
[_AVERAGE_DARKS_ and IRCAM, UFTI, and UIST variants; _GET_DARK_NAME_]

3.3   Flat-fielding

Some recipes create their own flats from the observations themselves, called a self flat, or sky
frames within a sequence that dithers to sky, or use a separate observation of a jitter on sky.
(There is no support for internal calibrations or dome flats in the recipes.) Whenever a flat
is required, recipes access the flat calibration index to find the most recent flat matching the
required attributes such as filter.

The current recipes for Michelle mid-infra-red observations do not create or use flat fields.

Creating a flat can be an iterative procedure, involving cleaning, making a first guess, object masking, proper normalisation, and making an improved flat. There are some primitives to bundle the operations including division by the flat field.
[_FLAT_FIELD_MASKED_GROUP_, _FLAT_FIELD_QUADRANT_JITTER_, _FLAT_FIELD_NCOLOUR_, _DIVIDE_BY_FLAT_, _DIVIDE_BY_FLAT_FROM_GROUP_]

There are variants for certain families of recipes, which marshall the various required subgroups of frames before dividing by the flat field.
[_DIVIDE_BY_FLAT_CHOP_SKY_, _DIVIDE_BY_FLAT_FROM_EXTENDED_, _DIVIDE_BY_FLAT_NOD_PAIRS_]

3.3.1   Flat creation

Frames are optionally cleaned to remove extreme outliers (± 3 or 6σ about the mean in 15×15-pixel neighbourhood) iterated three times, except for thermal recipes NOD_SELF_FLAT_NO_-MASK

and variants, and NOD_SKY_FLAT_THERMAL. The data are then normalised, combined pixel by pixel using a broadened median, and the combined array normalised to a have a mean of one. It is possible to use another statistic for the combination, such as a clipped median.
[_MAKE_FLAT_FROM_GROUP_, _MAKE_FLAT_FROM_NORMALISED_GROUP_, _MASK_DEVIANTS_]

There are variants for certain families of recipes, which marshall the various required subgroups of frames.
[_MAKE_FLAT_CHOP_SKY_, _MAKE_FLAT_FROM_NORMALISED_CHOP_SKY_, _MAKE_FLAT_EXTENDED_, _MAKE_FLAT_FROM_NORMALISED_EXTENDED_, _MAKE_FLAT_QUADRANT_JITTER_]

Flats are filed with the calibration system.
[_FILE_FLAT_]

3.3.2   Object masking

In recipes which make a flat using the frames taken of the targets, the so-called self flat, any sources present can bias the flat field, and result in blotchy mosaics. The full versions of such recipes, as opposed to the _BASIC versions, and SKY_FLAT_MASKED greatly reduce these artifacts using the following algorithm. After the application of the approximate self-flat field, an EXTRACTOR inventory is made of objects having at least 12 connected pixels above 1.0 σ above sky. (The thresholds can be changed in $ORAC_DATA_CAL/extractor_mask.sex through the DETECT_MINAREA and DETECT_THRESH parameters.) The locations, shapes, orientations and sizes are used to make a mask. The mask is applied to the dark-subtracted frames and a new flat created. As the outer parts of bright objects often leave residual unmasked blobs, a circular central occulting mask is used. The diameter is normally 7 arcseconds, but it can be adjusted through the OCCULT argument of primitive _MAKE_OBJECTS_MASK_. In the QUADRANT_JITTER recipe the central mask's diameter equals the length of the shorter side of a quadrant. The disadvantage is that the noise is higher within the occulted circle, and its variance is non-uniform across the central ninth of the mosaic. You can modify the the central mask size and shape; see primitives _FLAT_FIELD_QUADRANT_JITTER_ and _MAKE_OBJECTS_MASK_.

*[_MAKE_OBJECTS_MASK_, _MASK_OBJECTS_, _DEFINE_QUADRANT_MASKS_,
_MASK_QUADRANT_]*

After masking biases can be introduced as the objects or masks move to different locations on the detector each with a different response in the flat field. This is most pronounced for QUADRANT_JITTER where a quadrant of the detector is masked, and IRCAM2 which had a strongly sloping response. in which the mean flat is considerably different from the remaining quadrants. Merely taking a median at each pixel will preferentially select values from certain frames. Thus there has to be an allowance for these systematic differences before the data are combined to give representative relative intensities. The first frame becomes a reference frame against which the recipes scales the modal values of the other frames.
*[_NORMALISE_TO_MODE_, _NORMALISE_TO_MODE_EXTENDED_,
_CLIPPED_STATS_]*

The improved flat typically shows a uniformity at ∼0.02% of the sky. It is this flat which produces the flat-field frames for mosaicking. Systematic errors in the sky—a major uncertainty in infra-red point-source photometry—are also reduced significantly by this algorithm.

EXTRACTOR on occasions underestimate the sizes of objects, so there is an enlargement factor from 1.0–1.5, defaulting to 1.0, applied in primitive _MAKE_OBJECTS_MASK_. If at high contrast you find residual dark rings in your flat-fielded images, try adjusting the ENLARGE argument either in the recipe or the primitive default. See Section 4

and in particular Section 4.2.3

on how to tailor primitives.

3.4   Field-distortion Correction

The ISAAC instrument exhibits field distortion amounting to 2.5 pixels in the corners of the detector. If not corrected, mosaics with large dithers have multiple images in the overlap regions. Therefore recipes resample the image applying the published polynomial mapping. This correction also improves the registration. The sky co-ordinates are also corrected for the distortion.
*[ISAAC/_DEFINE_DISTORTION_TRANSFORMATION_,
ISAAC/_APPLY_DISTORTION_TRANSFORMATION_]*

3.5   Bias variation

The bias in the ISAAC short-wavelength camera depends upon the detected flux. Thus in target frames there is a residual bias not fully corrected by bias subtraction evident as two steepening ramps downwards to rows 1 and 513. An ISAAC-variant recipe corrects for this as follows. First within a flat-fielded frame they locate sources and mask them (as described in Section 3.3.2)

Then it forms a one-dimensional profile by masking objects collapsing along rows using the median, from which it subtracts a clipped mean of the profile to form a new profile of the bias variations. The bias-variation profile is then subtracted from each row of the original flat-fielded frame.
[ISAAC/_BIAS_CORRECT_GROUP_]

NACO has alternating positive and negative signals in its columns, most noticeable in longer exposures. The same filter as ISAAC, except it collapses along columns and does not mask

objects, is applied to the flat-fielded frames.
[NACO/_BIAS_CORRECT_GROUP_, _REMOVE_COLUMN_ROW_STRUCTURE_]

3.6 Sky Subtraction

In general the recipes do not sky-subtract in the literal sense of pixel-by-pixel subtraction of a sky frame, or better of some median of jittered sky or even target frames. Such recipes could readily be created if there is a demand. Instead the sky signature is usually accounted for in the flat-fielding. Therefore the mosaics generally have the sky signature removed, but not base level.

The sky varies rapidly in the thermal and mid-infra-red, so dithered pairs, or nodded and dithered pairs respectively are differenced to attempt to remove the sky signature. See Section 3.1.9.
[_DIFFERENCE_PAIR_]

Another recipe which performs a pixel-by-pixel sky subtraction is SKY_AND_JITTER. It's hardly used and not recommended because it demands a very stable sky, since only one sky frame is observed at the start of the observation; and a region of sky devoid of objects to avoid 'holes' appearing in the subtracted target frames.
[_SUBTRACT_SKY_SKY_AND_JITTER_]

Another form of sky subtraction is to remove a representative sky level. This benefits imaging of extended sources whose scale exceeds the dither pattern or even the detector's field of view. The normal procedure is to alternate between dithered integrations on target and a region of sky. The representative statistic is a multiply clipped mean at 2, 2.5, 3, then 3 standard deviations, which effectively gives the mode, and so is not biased by resolved sources. At present the CHOP_SKY_JITTER and EXTENDED_$n{\times}m$ recipes take the average of the modes of bracketted sky frames. For offline processing it would be possible to fit a spline, say, to the modes and provide a better subtraction. Under normal conditions, where the sky level is not varying rapidly or suddenly because of cirrus, the algorithm works well, especially over longer integrations.
[_FORM_SKY_LEVELS_, _NORMALISE_TO_MODE_EXTENDED_, _CLIPPED_STATS_,
_SUBTRACT_SKY_CHOP_SKY_, ESO/_SUBTRACT_SKY_CHOP_SKY_,
_SUBTRACT_SKY_EXTENDED_]

3.6.1 Polarimetry Extraction and Sky Subtraction

The polarimetry recipes are designed to work with a Wollaston prism. This divides the signal into four partitions. These are ordinary and extraordinary beams (usually abbreviated to e and o beams) for the target and a region of nearby sky. Thus the raw data comprise four strips with an aspect ratio of about 6.

These partitions are normally separated by a mask, but the recipes do not depend on having the mask to extract the various regions say by detecting the edges. For each instrument the pixel limits of each region are fixed. The current target limits are 30% to 70% of the width of the long axis of each region to allow for some reasonable dithering of point sources, since there are usually only three jitter positions, while making mosaics with few pixels not derived from all contributing jittered frames. For extended sources these limits change to 10% to 90% to include as much object as possible with smaller dithers and alternating to blank-sky regions. Thus the limits define a section 40% or 80% of the width of the frame, roughly centred on the source.

The limits on the sky regions are 1% to 99% of the frame width, mainly to avoid unreliable and pathological pixels at the detector's edge.
[<instrument>/_DEFINE_POL_REGIONS_]

The recipes extract the target regions into e- and o-beam frames. The modes (the means after clipping at 2, 3, 3 standard deviations) of the e- and o-beam sky regions are subtracted from their corresponding target beam, incorporating the uncertainty of each sky level in the corresponding target beam's variance array. For an extended-source observation, the sky levels are determined from the two corresponding regions for each beam in the following sky frame.
[_SUBTRACT_SKY_POL, _SUBTRACT_SKY_POL_EXTENDED_]

3.7 Automatic Registration

This makes an inventory of the sources above a threshold in each frame. It then performs a pattern recognition to identify common features in jittered frames. If the fraction of common objects is under 40% or the total is fewer than three, the registration fails, and so the script resorts to reading the telescope offsets stored in the FITS headers, or matching a central bright object in certain recipes. Using telescope offsets can lead to trailed sources, as occurred with the IRCAMDR package. The improved registration leads to the detection of fainter sources and more-accurate measurement thereof.
[_GENERATE_OFFSETS_ invoked within several wrappers tied to various families of recipes, _FIND_APPROX_OFFSETS_TELE_OFFSETS_, _GENERATE_TELE_OFFSETS_, _GET_CARTESIAN_TELESCOPE_OFFSETS_]

To make use of the best information, registration using more than one of the above methods is permitted.

There is also a new but not extensively tested option in which matching is performed only within overlapping regions as specified from the approximate world co-ordinate system, and a single match within 12 pixels is allowed to define the offsets between frames. This modification should allow more registrations using sources than from telescope offsets, which merely assist in the process. Since the robustness is unknown at present, this option is currently disabled by default. The easiest way to switch it on, is to change the the default value of the SKYREG argument primitive _GENERATE_OFFSETS_ to 1. See Section 4

for generic instructions to make a private _GENERATE_OFFSETS_.

NACO has a tiny plate scale; even with adaptive optics the tolerances (minimum number of pixels and the percentile threshold) searching for fiducial sources are increased, and by default the sky registration is enabled.
[NACO/_GENERATE_OFFSETS_]

3.7.1 Moving Targets

For a moving target, it is possible to permit the telescope to track at sidereal rate, and adjust the telescope offsets for the motion of a slowly moving target, *i.e.* one that stays within the field within an observation group. Registration uses the revised offsets and in the mosaic the stars trail while the object does not, and so can be seen and measured more easily.

At present the ephemeris is via a pre-prepared text file, given by environment variable

ORAC_EPHEMERIS, if defined, otherwise $ORAC_DATA_OUT/target_ephemeris.dat. See one of the moving-target recipes, say MOVING_JITTER_SELF_FLAT, for more details. Once the ephemeris is available as a web service, the file could become merely a backup; this would automate the process and so the observer would not have to prepare the file at altitude just before the epoch of observation.
[_ADJUST_OFFSETS_TO_MOVING_TARGET_]

3.7.2 Polarimetry Registration

Registration involves both the dithered observations as normal, but also the e- and o-beams at each waveplate angle must align. To aid the latter an approximate pixel displacement between the e- and o-beams from empirical data is set for each filter, as the displacement is wavelength dependent. Even an approximate offset almost invariably leads to successful registration using the central source.
[_GENERATE_OFFSETS_JITTER_POL_, _GENERATE_OFFSETS_POL_]

3.8 Mosaicking

This is fully automated. No longer do you have to measure painstakingly centroids and manually tile to form mosaics from the jittered frames. The jitter offsets are sufficiently small to permit shift of Cartesian co-ordinates to register. The offsets are derived from the automatic registration. No correction for the detector orientation is applied, since it degrades the quality of the data, despite the small rotation $(\sim 1°)$

normally involved. Allowance for this misalignment with the cardinal directions will be through the FITS world co-ordinate system. In the meantime should you need the rotation angle, pipe the output of KAPPA's **fitslist** function into grep.

```
% fitslist <your_frame> | grep CROTA2
```

The flat-fielded frame can either be resampled to give sub-pixel registration, or to the nearest pixel. The latter is much faster and is adequate for most uses of UFTI, UIST, thermal IRCAM, and Michelle. It also has the advantage of not smoothing the data and introducing covariance into the data errors. For older undersampled IRCAM3 data $(0.''286$

per pixel) and for INGRID $(0.''235)$

resampling has some merit.

The mosaicking uses the CCDPACK algorithm of its **makemos** command. Only zero-point shifts of intensity are applied to the resampled frames to create the mosaic. For most cases the comparison is of the sky levels as sky pixels dominate. This comparison is repeated for all pairs. **makemos** then finds the most mutually consistent set of additive corrections, weighting appropriately, to make the smoothest mosaic given the data. The first frame, which for the UFTI execs is the central (offset 0,0) frame, has no additive correction applied. The mosaic generation adjusts the zero-point of the jittered frames. Another way of looking at it is that mosaicking attempts to remove the sky variations. The additive corrections are normally quite small, like a few tenths of a count to a few counts. However, over longer integrations or in the thermal regime they can amount to a few score. A mosaic pixel value is the mean of all the adjusted contributing

pixel values at that location. It is possible to select other statistics for the contributing pixels, such as the median, through the METHOD argument of the _MAKE_MOSAIC_ family of primitives.

There is no normalisation to counts per second in the mosaic. The mosaic's signal corresponds to that of the first frame, thus the exposure time of the mosaic is that of one individual frame. The recipes assume that you have used their corresponding 'execs' or 'sequences', and hence have not changed the exposure time during a jitter. The exposure time (header EXP_TIME [UFTI/UIST/Michelle], EXPOSED [IRIS2], EXPTIME [Classic Cam/INGRID/ISAAC/NACO/-NIRI] or DEXPTIME [IRCAM]) is propagated from the first frame to the mosaic. Where multiple frames combined to create a mosaic pixel the signal-to-noise ratio corresponds to the combined integration time. The integration time (keyword INT_TIME [UFTI], TOTALEXP [IRIS2], or EXPOSED [IRCAM]) is the number of coadds times the exposure time per coadd.

Depending on the recipe, the mosaic may be trimmed to the dimensions of a raw frame. Mosaicking removes virtually all the bad pixels for standard stars where the jitter offsets are small.

A mosaic forms for each cycle of the recipe, *e.g.* all four frames in a QUADRANT_JITTER. For multiple cycles, an integrated 'grand' mosaic forms of improving signal-to-noise. To avoid the build up of bad pixels from cosmic rays, bad pixels are interpolated before the addition. This may result in some strange stripes in the top-left corner of UFTI frames where no interpolation can occur. Those pixels are bad in all frames and should be ignored. The exposure-time header for the integrating mosaic is the sum of the exposure times of the contributing mosaics. Again the signal is not divided by the exposure time.
[_MAKE_MOSAIC_ and invoked within several wrappers (all with the _MAKE_MOSAIC_ prefix) tied to various families of recipes]

3.8.1   Polarimetry Resampling

To permit the calculation of the Stokes parameters, polarimetry recipes resamples each frame using non-integer Cartesian offsets, or merely finds the offsets between frames to the nearest pixel and shifts the origin. The mosaic extends to include all pixels in contributing the frames, however, in practice there should be at most one pixel variation in dimensions.
[_RESAMPLE_MOSAICS_]

3.9   Polarimetry Parameters

These are outlined in the "Output Data" section of the polarimetry recipes such as POL_JITTER, and are calculated using standard formulae and the methods of POLPACK. A calibration correction (polrefang) measured in degrees clockwise is applied to the vector direction for the orientation of the analyser with respect to north based upon UKIRT's IRPOL instrument. Current values are -24 for Michelle and UIST, -9 for UFTI, and -96.3 for IRCAM. For other instruments, the offset can be determined using polarimetric-standard-star calibrations.
[_CALC_STOKES_, MICHELLE/_CALC_STOKES_NOD_CHOP_]

3.10   Near Infra-red Aperture Photometry

The recipes with an _APHOT suffix (except NOD_CHOP_APHOT) perform aperture photometry on the mosaic and the contributing flat-fielded frames. The method assumes that the target,

usually a standard star, is approximately centrally located after allowing for the jitter offsets. If you have data where the star lies outside the aperture, it is possible to apply an offset. See the RAOFF and DECOFF arguments of primitive _FIXED_APERTURE_PHOTOMETRY_ in _APHOT_MAG_ to adjust the aperture's position. The thermal chopped data have sources displaced from the centre, but the reductions allow for the symmetric offsets about the mean jitter position. For UIST thermal data, the search algorithm does not assume a central or centrally symmetric distribution of the positive and negative signals.

Residual bad pixels (usually in the individual flat-fielded frames are removed by median filtering. This does leave a bias in the wings of stars, but certainly the underestimate is far less than ignoring the bad pixels, and is typically far less than the other photometric errors.

The photometry is through a circular aperture located at the centroid of the source, with the sky measured from a concentric annulus outside the aperture. The default aperture size is 5 arcseconds (3 arcseconds for NACO). The annulus diameters are 6.5 to 10 arcseconds (all instruments but UFTI and NACO), 6.5 to 12.5 arcseconds for UFTI, and 3.9 to 6 arcseconds for NACO. The default estimator of the sky flux is the mode calculated from $3 * \text{median} - 2 * \text{mean}$ and Chauvenet's rejection criterion. The photometry accounts for fractional pixels at the aperture edge but without allowance for the local gradient.

The magnitudes are given by the expression $-2.5*\log10(\text{abs}(\text{counts})$ per second exposure time). Therefore negative sources can be measured too, as presented by the thermal photometry recipe NOD_SELF_FLAT_NO_MASK_APHOT. The photometry also yields an internal error determined from the sky variance.

A case- and space-insensitive comparison of the object name with the entries in a table provides a catalogue magnitude for a standard star in *I*, *Z*, *J*, *H*, or *K* for all instruments, and in *L* or *M* for IRCAM and ISAAC. Also a mean extinction is applied for the mean of the start and end airmasses. Thus the primitive calculates an approximate zero point. Note that ISAAC and NACO standard stars include additional objects not present in the UKIRT faint-standard list or Persson's HST list; for these the magnitude and derived zero point will not be determined automatically. For accurate photometry the actual extinction coefficients should be determined. As the output from the photometry is a small text list, you can use **catphotomfit** command of the CURSA package to achieve this. The units and meanings of the columns are documented within each small text list.

The seeing is estimated for each frame and the mosaic by fitting a two-dimensional Gaussian to the star, although in good seeing the UKIRT images are more centrally concentrated than a Gaussian. The full-width-half-maximum so derived is also tabulated in the small text list.
[_APHOT_MAG_, _APHOT_MAG_NCOLOUR_, _NOD_APHOT_MAG_ (plus NACO and UIST variants), _FIXED_APERTURE_PHOTOMETRY_, _MAKE_PHOTOMETRY_TABLE_, _GET_FILTER_PARAMETERS_ (and several instrument variants), _PERSSON_STANDARD_MAGNITUDE_, _UKIRT_STANDARD_MAGNITUDE_, _STANDARD_MAGNITUDE_ (and instrument-specific variants), _CLIPPED_STATS_, _FIND_SOURCE_CENTROID_, _GET_FRAME_CENTRE_]

3.11 Mid-infra-red Aperture Photometry

This is similar to near infra-red aperture photometry. The main difference is the source measured. Rather than measuring each positive and negative image, NOD_CHOP_APHOT integrates the average flux of the one or two pairs of positive and negative images in the chopped and

nodded mosaic. The number of pairs depends on the chop-and-nod orientations and the nod throw. The four (or two) images are first registered using their centroids; extracted with a symmetric neighbouring background, but without any duplication of pixels from the original mosaic; and finally combined, averaging to preserve the flux per unit time. The exposure-time header, therefore, remains that of the mosaic.

The aperture size is 5 arcseconds. The background is determined from an annulus with an inner diameter of 7.5 arcseconds and an outer diameter of 15 arcseconds enclosing the source. The Michelle chip characteristics and the UKIRT chop throw limit the area of this combined-image mosaic—normally a 15-arcsecond square—leaving comparatively little background. Recommended apertures and background regions may change in the light of experience with Michelle. Already in below-average seeing, it's evident that a 6-arcsecond circle will leave some signal in the background.

The object name is compared with a 28-member catalogue of 10- and 20-micron standards, a handful of which are known optical semi-regular variable stars. In addition to deriving magnitudes, the recipe calculates an approximate flux in Janskys. It is approximate because being an *absolute* measurement, it does require a magnitude zero point to be applied to the *relative* instrumental magnitude. At the time of writing this zero-point is only roughly known, so the default fluxes should be taken with a pinch of salt. The flux is not yet written to the results file. If you have determined the zero point from standards, you can rerun the pipeline for your target observations with that zero point assigned to argument ZP of primitive _NOD_CHOP_APHOT_MAG_ invoked within recipe NOD_CHOP_APHOT. See Section 4

for instructions to make and use a private version of a recipe.

There is an approximate extinction correction applied using a coeeficient of 0.18, because the coefficients for all but one of the various *N* and *Q* filters have yet to be determined.
[_NOD_CHOP_APHOT_MAG_, _FIXED_APERTURE_MIDIR_PHOTOMETRY_,
_MAKE_PHOTOMETRY_TABLE_, _GET_FILTER_PARAMETERS_,
_UKIRT_MIDIR_STANDARD_MAGNITUDE_, MICHELLE/_STANDARD_MAGNITUDE_, _FIND_SOURCE_CEN
_CLIPPED_STATS_]

3.12   Improving the signal-to-noise of Mid-infra-red Data

The Michelle electronics can leave an uneven mosaic with vertical banding from bias variations and horizontal ripple patterns from electronic pickup. Therefore, Michelle NOD_CHOP recipes subtract the median along each column of the mosaic, then subtracts the median along each row. This cleaning aids the visibility of faint sources.

In the mid-infra-red, the sky signal is vastly greater than the signal than even the brightest sources. While the nodding removes the bulk of the sky signal, the sky noise remains, still swamping the signal from a faint source. Integrating over many nod-chop cycles is needed, and the real-time display of ORAC-DR does permit interactive review of the signal-to-noise in GAIA so observers can curtail data collection when the required contrast is achieved. Averaging the positive and negative signals and neighbourhoods into a combined signal (*cf.* Section 3.11

only helps by a factor of two (or $\sqrt{2}$ for a single positive and negative pair). Recipe NOD_CHOP_FAINT smooths the combined quadrant with a 4-by-4-pixel neighbourhood running average filter to help reveal faint sources. Note that the source centroids are not used for registration as the signal is usually too weak.

While smoothing reveals the sources, it is not sufficient. There is also a confusion issue. The chopping can bring positive and negative sources actually located beyond the final quadrant to within it. There is an option of the _COMBINE_CHOPPED_SOURCE_ primitive, called by recipe NOD_CHOP recipes, which attempts to clarify which sources are actually present in the quadrant by forming a quality map. Each quality pixel is the sum of the four corresponding pixel values divided by their absolute values, after changing sign for the quadrants containing the negative images. In the map +4 indicates that a pixel had positive contributions from the positive quadrants and negative signals from the negative quadrants. A quality of +4 strongly implies that the signal is really at the sky location indicated. Thus it helps to discriminate from sources which have been chopped into view, for which there are no positive or negative counterparts (values $+/-2$) or noise (0, $+/-2$). Around the sky (which should be near zero) noise randomises the quality measurement, therefore smoothing should be used in conjunction with the quality map.
[_COMBINE_CHOPPED_SOURCE_, _REMOVE_COLUMN_ROW_STRUCTURE_, _REMOVE_COLUMN_ROW_STRUCTURE_SCAN_, FIND_SOURCE_CENTROID_, _GET_CHOP_OFFSETS_, _GET_FRAME_CENTRE_]

To create a quality map at each cycle, you should set argument QMAP=1 and SMOOTH> 1. For example in NOD_CHOP_FAINT, append `" QMAP=1"` to the line

```
_COMBINE_CHOPPED_SOURCE_ METHOD=median CENTROID=0 SMOOTH=4 CLEAN=1
```

3.13   Catalogue Generation

The recipes with a _CATALOGUE suffix create a source catalogue from the mosaic using EXTRACTOR. The catalogue includes objects having at least 12 connected pixels above 1.0 σ above sky. The catalogue is written in ARK Cluster format, with a field ID in column 0 (this value is typically zero); an object ID in column 1; right ascension in columns 2, 3, and 4; declination in columns 5, 6, and 7; x and y positions in columns 8 and 9; uncalibrated magnitude in column 10; the error in magnitude in column 11; and a quality flag in column 12 (this value is typically zero). The magnitudes are given by the expression $-2.5*\log10(\text{counts})$ per second exposure time.

As the final mosaic has varying noise characteristics, with higher noise regions at the edges, the detection limit varies across the mosaic. Fainter objects can be detected in the central region than near the edges and corners.
[_CREATE_SOURCE_CATALOGUE_, _GET_CATALOGUE_NAME_]

3.14   eSTAR Integration

ORAC-DR is also integrated with the eScience Telescopes for Astronomical Research (eSTAR) project, which is designed to automatically detect and follow up on transient astronomical objects using manned and robotic telescopes. Using the catalogue generated in Section 3.13

ORAC-DR will automatically send a trigger to the eSTAR network containing a FITS file of the final mosaic and the catalogue. This step is only done when an observation is taken as part of an eSTAR project, `i.e.` there is a RMTAGENT header and its value is ESTAR.
[general/_TRIGGER_ESTAR_, _SET_REMOTE_AGENT_HEADER_]

3.15   Tidying

Each recipe has a tidy procedure, which removes unnecessary intermediate frames when the recipe no longer requires them. Retained are the raw data, flat-fielded frames, differenced pairs (for chopped data), and the mosaics. Most of the intermediate small text files are removed in individual primitives, but some registration-related files do persist until the tidy script cleans up. If you need to retain the intermediate files, comment out (#) the final instruction of the recipe, which calls the tidy primitive, and follow the instructions in Section 4

to make and use a private version of a recipe, or set the $ORAC_KEEP environment variable to 1. *[<recipe_family>_TIDY_, such as _EXTENDED_TIDY_, _JITTER_SELF_FLAT_TIDY_, _NOD_CHOP_TIDY_, _REDUCE_DARK_TIDY_; general/_DELETE_A_FRAME_, general/_DELETE_INTERMEDIATE_GROUP_FILES_, general/_DELETE_TEMP_FILES_, general/_DELETE_TEMP_GROUP_FILES_, _IMAGING_GOODBYE_]*

4   Customising Recipes

If you wish to write your own data-reduction recipes, you should consult the ORAC-DR *Programmer's Guide* (SUN/233). However, for most purposes, observers wishing to modify existing scripts can get by without this document.

A easier-to-use tailoring system to control parameters and primitive arguments from the command line and 'personal' style files is under consideration.

4.1   Search paths

ORAC-DR allows you to create your own recipes and primitives, or modify those provided as part of the package. In either case you must tell ORAC-DR where your recipes and/or primitives are stored. This is achieved through two environment variables. $ORAC_RECIPE_DIR should equate to the directory containing your recipes. $ORAC_PRIMITIVE_DIR specifies the directory containing your primitives. Here's an example.

```
% setenv ORAC_RECIPE_DIR /home/user/drmoan/recipes
% setenv ORAC_PRIMITIVE_DIR /home/user/drmoan/primitives
```

Once these environment variables are defined, ORAC-DR first looks in $ORAC_RECIPE_DIR or $ORAC_PRIMITIVE_DIR to find a recipe or primitive respectively. If the script is absent, ORAC-DR looks in the standard $ORAC_DIR directories.

4.2   Anatomy of an imaging recipe

There are documentation modules—a Starlink style between #+ and #- delimiters at the head, and a Perl POD (Plain Old Documentation) at the foot. Between these is the code. This consists of calls to primitives, sometimes with arguments. Primitives have uppercase names preceded and terminated by underscores, such as _DIVIDE_BY_FLAT_.

4.2.1 Hello primitives

The first of these primitives is _IMAGING_HELLO_. It contains instrument-specific code and initialisation. It is best left alone. See

Section 3.1.2 for a description of what this primitive does for each instrument.

Second there is a recipe-specific primitive such as _JITTER_SELF_FLAT_HELLO_. This sets up CCDPACK, sets directives when to perform certain operations, optionally create variances, removes any bias, and edits the FITS headers.

Two things you might wish to change in the recipe `_HELLO` script are listed below.

- Change the extent of the images. If there is an instrumental defect in some peripheral rows you might not want to use the full bounds as given by headers `ORAC_X_LOWER_BOUND`, `ORAC_X_UPPER_BOUND`, `ORAC_Y_LOWER_BOUND`, `ORAC_Y_UPPER_BOUND`. Suppose you wanted to trim off the top three rows you could change the line.

  ```
  my $y2 = $Frm->uhdr( "ORAC_Y_UPPER_BOUND" );
  ```

  to

  ```
  my $y2 = $Frm->uhdr( "ORAC_Y_UPPER_BOUND" ) -3;
  ```

- Switch on error propagation. To save time at the telescope, the pipeline does not keep track of the errors per pixel (except for the polarimetry recipes with names starting "POL" and the NOD_CHOP series). If you wish to know realistic errors for your data, in the recipe switch on the USEVAR argument for the recipe's `_HELLO` primitive. Here is an example for the BRIGHT_POINT_SOURCE recipe.

  ```
  _BRIGHT_POINT_SOURCE_HELLO_ USEVAR=1
  ```

4.2.2 Steering primitive

Within a recipe's `_HELLO` primitive, a 'steering' primitive is invoked. These are best left well alone. They control when the various operations are performed. See

Appendix F.2 for more details. What you can safely adjust are the configurable steering parameters listed in the recipe documentation. In the main these parameters set the number of frames processed in a cycle through the recipe. The parameters are passed by argument through the recipe's `_HELLO` script to the steering primitive.

Suppose that for some reason an observation of a nine-point jitter self flat was aborted after seven positions. If you try the recipe stored in the headers some processing will occur, but it will not include mosaic creation. The final steps including mosaic creation occur once all nine frames are dark subtracted. Now there are no seven-point recipes to substitute on the command line. You could make your own seven-point recipe to reduce those data. First make a new recipe by copying the standard one.

```
% cd $ORAC_RECIPE_DIR
% cp $ORAC_DIR/recipes/imaging/JITTER_SELF_FLAT ./JITTER7_SELF_FLAT
```

Next edit JITTER7_SELF_FLAT and alter the line

```
_JITTER_SELF_FLAT_HELLO_
```

to become

```
_JITTER_SELF_FLAT_HELLO_ NUMBER=7
```

The recipe will then generate the self flat, flat field and make the mosaic once the seventh frame is dark-subtracted.

Recipes are stored in `$ORAC_DIR/recipes/imaging`; and for a few instrument-specific recipes in `$ORAC_DIR/recipes/<instrument>`, where <instrument> is IRCAM, MICHELLE, UFTI. IRIS2, ISAAC, or INGRID. See

Appendix E for some details.

4.2.3   Recipe primitives

After the steering primitive we come to the recipe-specific scripts that actually perform the recipe. The most likely and easiest things you would change are to add arguments or modify argument values of the primitives in the recipes. For instance, you might wish to change the aperture diameter for the aperture photometry. To alter to a 4-arcsecond aperture, change the APERTURE argument's value of the _APHOT_MAG_ primitive like below.

```
_APHOT_MAG_ APERTURE=4
```

To obtain details of a primitive's arguments, use the **oracman** or **perldoc** command. Thus

```
% oracman _MAKE_MOSAIC_
```

will display the documentation for primitive _MAKE_MOSAIC_. Space does not permit inclusion of the documentation of the many primitives in this manual. Most of primitives' source code is stored in `$ORAC_DIR/primitives/imaging`; the instrument-specific ones are situated within `$ORAC_DIR/primitives/<instrument>`, where <instrument> is UFTI, UIST, MICHELLE, IRCAM, IRIS2, ISAAC, or INGRID; and there are a few general scripts in

`$ORAC_DIR/primitives/general`.

While the simplest primitives just invoke a Starlink task, and updates just that and are amenable to customisation, some are quite complex especially for the registration. They may invoke other primitives, manipulate parameters and small data files so that the various tasks connect to cope with a variety of circumstances. The most likely change you will want to make is to change the parameter values of a Starlink task. Armed with the reference documentation for the application, say with a `findme <application>`, it is easy to change values or append further parameters.

Here is an example. Let us suppose you wanted to combine frames to make a self flat, not with the median, since you have heard that a mean trimmed of the most-extreme tenth of the values gives better results. First copy _MAKE_FLAT_FROM_GROUP_ to your primitives directory.

```
% cp $ORAC_DIR/primitives/imaging/_MAKE_FLAT_FROM_GROUP_ $ORAC_PRIMITIVE_DIR
```

Using an editor, find the first line in your copy of _MAKE_FLAT_FROM_GROUP_ commencing
`$hidden`. It should be as follows.

```
$hidden = "method=median sigmas=2.0 reset accept";
```

Change this to

```
$hidden = "method=trimmed alpha=0.1 sigmas=2.0 reset accept";
```

to effect the change of statistic. There is in fact a second line assigning variable `$hidden`
depending on argument CLEAN, and you should make the same alteration there too.

If there is demand, additional arguments could be provided for primitives, to simplify control.
Please contact the author if you have suggestions for arguments and new recipes, or need help
customising your ORAC-DR scripts.

4.3   Index files

Once the pipeline has run for a bit you will find text files in `$ORAC_DATA_OUT` called `index.flat`,
`index.dark` amongst others. These list the calibration frames. ORAC-DR uses these to find the
most-recent, appropriate calibration. For example, a flat requires that the filter of the flat matches
that of the frame being flat fielded, and a dark must have the same exposure time as the target
frame; and both must have been taken in the same instrument mode.

Here is an example of a flat index.

```
#FILTER MODE ORACTIME RDOUT_X1 RDOUT_X2 RDOUT_Y1 RDOUT_Y2 WPLANGLE
flat_Lp98_23 Lp98 STARE 13.3484 1 256 1 256 0
flat_K98_pol0_62 K98+pol NDSTARE 7.971 1 1024 1 1024 0.000
flat_K98_pol22_62 K98+pol NDSTARE 7.971 1 1024 1 1024 22.5
flat_K98_pol45_62 K98+pol NDSTARE 7.971 1 1024 1 1024 45
flat_K98_pol67_62 K98+pol NDSTARE 7.971 1 1024 1 1024 67.5
flat_J98_88 J98 flush_read 7.46111E+00 1 1024 1 1024 0
flat_H98_93 H98 flush_read 7.58330E+00 1 1024 1 1024 0
flat_K98_98 K98 flush_read 7.70559E+00 1 1024 1 1024 0
flat_H98_133 H98 flush_read 8.99034E+00 1 1024 1 1024 0
flat_H98_138 H98 flush_read 9.07139E+00 1 1024 1 1024 0
flat_H98_138_c1 H98 flush_read 9.12074E+00 1 1024 1 1024 0
flat_K98_290_row0 K98 flush_read 1.13094E+01 1 1024 1 1024 0
flat_K98_290_row1 K98 flush_read 1.15080E+01 1 1024 1 1024 0
```

The first line contains the column headings. ORACTIME is the UT in decimal hours, and
WPLANGLE is the polarisation waveplate angle.

In general you should not manipulate these files. Mis-editing can lead to the calibration system
breaking down. If you must edit this file, say to exclude a poor dark or an uneven flat, restrict
yourself deleting the line corresponding to that calibration file. It's safer to remove the calibration
file and recreate a new one with the calibration frames you want by running the pipeline.

If you want to nominate specific calibration frames, overriding those selected from the calibration
indices, there is a `-calib` option for the **oracdr** command to do this. See the section on
calibration options in SUN/230 for examples.

5   Correcting headers

There are reasons why you may need to edit some of the FITS headers used by ORAC-DR.

- At the summit of Mauna Kea, it's easy to make mistakes. One of the common ones is to make an error in the 'exec' or 'sequence'. This can cause, for example, frames to be in the wrong observation groups or be assigned the wrong data-reduction recipe. While the ORAC Observation Tool has reduced the frequency of such errors, they will not be eliminated.

- You may have made some trial observations before taking making a longer integration through several cycles of a recipe. Now you wish to combine all the observations of a target to obtain the best signal-to-noise.

The main headers to change are

RECIPE —the data-reduction recipe;

NOFFSETS —the number of offsets;

OBSNUM —the number of the frame, starting from 1 on each night;

GRPNUM —the group number, and should be given by the frame number (OBSNUM) of its first member; and

GRPMEM —whether or not the frame participates in group processing.

For IRCAM, Michelle, UIST, UFTI, and converted ISAC files, it is possible to edit the NDF's FITS extension using KAPPA's **fitsmod** command. The command is a bit long and the author regrets not defining a **fitsupdate** synonym. The following changes the GRPNUM keyword to have value 36 in the raw NDF frame f19991108_00042.

```
% fitsmod f19991108_00042 grpnum u 36 \$C !
```

It's possible to edit many files using a C-shell or Perl script to edit a series of files very quickly. If you do, it's better to specify the values by keyword instead of position, like this

```
% fitsmod ndf=f19991108_00042 edit=update keyword=grpnum value=36 \
          position=! comment=\$C
```

because it is better insulated against change to **fitsmod**.

If the file being edited is a multi-NDF container file, you can avoid disconcerting, but harmless error messages if you change the **fitsmod** command to specify the HEADER NDF. Here is an example for Michelle data, which changes the number of offsets.

```
% fitsmod m20011107_00079_raw.header noffsets u 5 \$C !
```

For Classic Cam, IRIS2, INGRID, ISAAC, NACO, NIRI, or old UFTI data, you can edit the raw FITS files. $ORAC_DIR/bin/fitsmod.pl is a documented example Perl script to edit FITS headers. The intention is for you to make a copy and edit to suit your particular header-editing requirements. For some of these instruments, it may prove easier to edit the FITS headers in the NDF form of the raw data, especially the ESO cameras, which have hierarchical headers.

6 Acknowledgments

ORAC-DR was developed at the Joint Astronomy Centre by Frossie Economou and Tim Jenness in collaboration with the UK Astronomy Technology Centre as part of the ORAC project. I should like to thanks to members of the ORAC team, UKIRT staff and observers who made suggestions for new or improved recipes. Special thanks go to Gillian Wright and Sandy Leggett for defining the initial specifications of the UFTI scripts, and for subsequent discussions. Chris Davis kindly supplied the specifications of the polarimetry and Fabry-Perot recipes. Paul Hirst wrote the original versions of the primitives which create the data variance. Stuart Ryder was instrumental in getting ORAC-DR commissioned and installed for use at the AAT. Thanks also to Frossie Economou and Tim Jenness for answering my ORAC-DR and Perl questions, and for incorporating my requested enhancements into ORAC-DRinfrastructure.

The application engines used in ORAC-DR were supplied by the Starlink Project, which is run by CCLRC on behalf of PPARC. I should like to thank the Starlink programmers for their excellent support, especially for quickly providing enhancements to tasks.

7 Copyright and License

ORAC-DR is copyright ©1998–2003 PPARC (the UK Particle Physics and Astronomy Research Council). It is distributed by Starlink under the GNU General Public License as published by the Free Software Foundation.

Whenever you have used ORAC-DR as part of a publication, please give an acknowledgment to ORAC-DR in the paper. This will help us assess the usage of ORAC-DR.

A   Processing UKIRT data obtained before 2000 August

Before the introduction of ORAC on 2000 August 1, UFTI raw data were in FITS format, IRCAM data had a different naming convention, and there were different default paths for `$ORAC_DATA_IN` and `$ORAC_DATA_OUT`. IRCAM's NDFs were copied to `$ORAC_DATA_OUT`, and UFTI's raw FITS files were converted to NDFs in that directory.

To process data from this era, follow the instructions

of Section 2.1 except you should invoke

```
% oracdr_<instrument>_old <date>
```

instead of

```
% oracdr_<instrument> <date>
```

to set up the necessary environment variables. The rest is the same.

You can use modern jitter-generic recipes too, provided they know how many frames to process. The easiest way to do that is make your own copy of the recipe and set the number frames as an argument to the steering primitive. See

Section 4.2.2 for details.

The standard raw and reduced directories prior to 2000 August were <instrument>_data/YYYYMMDD/raw/, and <instrument>_data/YYYYMMDD/reduced/ respectively, where <instrument> was either `ufti` or `ircam`.

Details of the former naming convention for IRCAM frames is given under the "Output Data" headings in the reference section.

B   File prefixes

The UKIRT-style naming convention comprises a prefix followed by the eight-digit UT date; underscore; the group or observation number, depending on whether it is a file associated with the group like a mosaic, or a single frame; and finally if it has undergone processing, a suffix.

Section C discusses and lists the last of these.

As a rule of thumb, the group prefix is the frame prefix preceded by g. Here is a table of the prefixes.

Instrument	Frame Prefix	Group Prefix	Notes
Classic Cam	cc	gcc	Applies to raw frames after processing with **cc2oracdr**.
INGRID	r	gingrid	Format is <prefix><obs_number>, for individual frames. Group files use the UKIRT convention.
IRCAM	i	gi	
	ro	rg	Before 2000 August
IRIS2		gi	Format is <date><obs_number>, for individual frames, where <date> is in the form ddmmm. Group files use the UKIRT convention.
ISAAC	isaac	gisaac	Applies to raw frames after processing with **isaac2oracdr**.
Michelle	m	gm	
NACO	naco	gnaco	Applies to raw frames after processing with **naco2oracdr**.
NIRI	N	gN	
UFTI	f	gf	
UIST	u	gu	

ORAC-DR converts the FITS data for instruments like INGRID and NIRI, note that the naming convention may be different for the raw FITS data compared with their raw NDF counterparts. For example, NIRI files are named `N` followed by the eight-digit UT date, then `S` and the four-digit observation number; and have file extension `.fits`.

C   File suffices

Files generated during ORAC-DR imaging data reduction have suffices denoting the processing step that created them. This appendix contains a list with short descriptions of what they mean. Most will be removed once a recipe has finished using them. So you will probably only see these files if you list the contents of directory `$ORAC_DATA_OUT` while the pipeline is running, or you interrupt the pipeline with `CTRL/C`, or something has gone wrong with the the recipe and pipeline has aborted, or you have commented out the '_TIDY' primitive from the recipe, or you have set the `ORAC_KEEP` environment variable to 1.

Frame suffices

Suffix	Stands for	Description
_adu	ADU	Multiplied by the exposure time to convert the units to ADU (Michelle)
_bc	Bias Corrected	Residual bias variations removed (ISAAC)
_bgl	BackGround Limited	Whether or not each pixel is background limited, `i.e.` Poisson noise exceeds the read noise
_bp	Bad Pixel	Co-added with the bad-pixel mask
_bpc	Bad Pixel Cumulative	Cumulative bad-pixel mask (UIST)
_bpd	Bad Pixel Data	Thresholded bias or dark frame for bad-pixel mask creation (UIST)
_cl	CLone	Modifiable copy of IRCAM raw data
_db	De-Biassed	The bias is actually zero, but it sets up various CCDPACK ancillary data for later processing
_dcb	Differenced Chop Beams	The difference of the A- and B-beam signals of nodded data, as used by NOD_CHOP recipes
_dk	DarK	Dark subtracted
_dp	Differenced Pair	The difference of successive frames in a NOD recipe
_dta	Distortion Transfomation Applied	Resampled for field distortion
_ess	E-beam Sky Subtracted	Polarimetry target e-beam after sky subtraction
_ff	Flat Field	Divided by the flat field
_fm	Flat Masked	This has the flagged deviant pixels detected by the initial flat-field creation restored after object masking
_fpm	Fabry-Perot Masked	After a mask is applied to exclude regions beyond the circle transmitted by the Fabry-Perot etalon.
_md	Masked Deviants	Deviant pixels from the neighbourhood (usually $3\,\sigma$ in $15{\times}15$-pixel region) flagged as bad
_nl	Non-Linearity	The standard non-linearity correction has been applied (IRCAM only)
_nm	Normalised to Mode	Normalised masked frames combined to make the flat field
_om	Objects Masked	This has sources masked with bad values so that they do not bias the self flat field
_oss	O-beam Sky Subtracted	Polarimetry target o-beam after sky subtraction

Suffix	Stands for	Description
`_qm`	Quadrant Masked	One of the quadrants is masked with bad pixels, created in QUADRANT_JITTER
`_pov`	POisson Variance	Poisson variance added
`_raw`	Raw copy	Copy of the raw data, but in output directory and has history recording enabled
`_rnv`	Read Noise Variance	Variance created containing the readnoise
`_ss`	Sky Subtracted	Global or local sky subtraction applied
`_th`	THresholded	Non-physical values set to bad
`_trn`	TRaNsform	The transformed or resampled data immediately prior to making a mosaic
`_xpr`	X PRofile	Median of each row (ISAAC)
`_ypr`	Y PRofile	Median of each column (ISAAC)

Group suffices

Suffix	Stands for	Description
_An	A beam Negative	Extracted negative A-beam source and region from chopped and nodded mosaic
_Ap	A beam Positive	Extracted positive A-beam source and region from chopped and nodded mosaic
_Bn	B beam Negative	Extracted negative B-beam source and region from chopped and nodded mosaic
_Bp	B beam Positive	Extracted Positive B-beam source and region from chopped and nodded mosaic
_cab	Combined A & B beams	Combined positive and negative images extracted from chopped and nodded mosaic
_cpc	Column-Profile Corrected	Removed column pattern, likely arising from pickup
_fb	Filled Bad pixels	Bad pixels in the mosaic are filled using smooth function of the neighbouring good pixels
_I	Intensity	Polarisation intensity
_mos	Mosaic	Final mosaic
_P	Percentage	Percentage polarisation
_PI	Polarisation Intensity	
_Q	Stokes Q	Stokes Q parameter
_qcab	Quality Combining A & B beams	Quality map from combining positive and negative images extracted from chopped and nodded mosaic
_rpc	Row-Profile Corrected	Removed row pattern, say due to bias variations
_scab	Smoothed Combined A & B beams	Block-smoothed combined positive and negative images extracted from chopped and nodded mosaic
_sp	Stokes Parameters	Data cube of Stokes parameters
_TH	THeta	Polarisation angle
_U	Stokes U	Stokes U parameter
_w	Wavelength	Fabry-Perot mosaic from different wavelengths
_xpr	X PRofile	Median of each row of the mosaic
_ypr	Y PRofile	Median of each column of the mosaic

Deprecated suffices

The following frame suffices were present prior to version 3.0 of ORAC-DR.

Suffix	Stands for	Description
_dg	De-Glitched	Bad pixels replaced by median of neighbours
_sbp	Substitute Bad Pixels	Bad pixels replaced (needed for PISA)
_sc	SCaled	Data scaled to lie within the range of values allowed by PISA for the object masking

The following mosaic suffices were present prior to version 3.0 of ORAC-DR.

Suffix	Stands for	Description
_mu	Mosaic (Unfiltered)	Intermediate mosaic (could contain bad/hot pixels)

D   Recipes

The original set of recipes and names were prescribed in G.S. Wright & S.K. Leggett, 1997, *Scripts for UFTI*, orac009-ufts, v01.

D.1   Classified Recipes

In hindsight you may decide that there was a better recipe for your data than stored in the RECIPE header. Also you may have used a faster variant of a recipe at the telescope, but now want the full reduction. Here is a classified list so that you can select an alternative. Magnitudes and dimensions apply to UFTI, except for the NOD_SELF_FLAT_NO_MASK recipes, whose magnitude ranges are for IRCAM; and the *NOD_CHOP* recipes which are applicable to Michelle. For IRCAM, dimensions are 10% smaller; for IRIS2 dimensions are 6.8× larger; for UIST they are either 33% smaller or 22% larger depending on the camera. The magnitude ranges are courtesy of Sandy Leggett and apply to UKIRT.

Calibration Recipes

Recipe Name	Type of Data	Function and Comments
ARRAY_TESTS	Array check	Calculates read noise and dark current.
DARK_AND_BPM	Dark and Mask	Measures dark current and creates a new bad-pixel mask for UIST.
DIFFERENCE_STATS	Array Check and Mask	Calculates statisticd for Michelle darks in a pairwise manner.
LAMP_FLAT	Mask	Creates and files imaging flat fields derived from a calibration lamp for ESO instruments.
MAKE_BPM	Mask	Creates a bad-pixel mask by standard-deviation thresholding.
MEASURE_READNOISE	Array check	Measures and files the readnoise for UIST from a set of dark frames.
REDUCE_DARK	Dark	Averages and files observations as the current dark.
REDUCE_FLAT	Flat	Reduces an imaging flat field.
SKY_FLAT	Flat	Creates and files a flat field derived from five jittered frames. Mostly for use with BRIGHT_POINT_SOURCE recipes. Requires a dark.
SKY_FLAT_FP	Flat	Creates a Fabry-Perot sky flat (from jittered blank-sky exposures) FP at on- and off-line wavelengths. Requires a dark.
SKY_FLAT_MASKED	Flat	As SKY_FLAT but masks objects to give a better flat field.
SKY_FLAT_POL	Flat	Obtain a 'master' polarimetry flat field from the median average of eight jittered frames; the waveplate is cycled after every second frame. Makes a copy of the flat for each waveplate angle. Requires a dark.
SKY_FLAT_POL_ANGLE	Flat	Obtain four polarimetry flat fields, one for each waveplate angle, from the median average of jittered frames. Requires a dark.

Miscellaneous recipes

Recipe Name	Type of Data	Function and Comments
ADDWCS		Creates the valid WCS in the FITS headers of raw data.
NIGHT_LOG		Generates a text-file log of a series of observations.

Very-bright-point-source recipes

Recipe Name	Type of Data	Function and Comments
BRIGHT_POINT_SOURCE	$IZ < 13$, $JHK < 9$, and bright $13 < IZ < 17$, $9 < JHK < 15$	Normally a 5-point jitter but would be usable as 3-point. Requires a separate flat as the background is too low to self flat, and a dark.
BRIGHT_POINT_SOURCE_APHOT		As BRIGHT_POINT_SOURCE, but also performs aperture photometry of the source.
BRIGHT_POINT_SOURCE_-NCOLOUR		As BRIGHT_POINT_SOURCE, but produces filenames that include filters for easier identification for multi-colour observations.
BRIGHT_POINT_SOURCE_-NCOLOUR_APHOT		As BRIGHT_POINT_SOURCE_-APHOT, but produces filenames that include filters for easier identification for multi-colour observations.
BRIGHT_POINT_SOURCE_TELE		As BRIGHT_POINT_SOURCE, but uses telescope offsets for registration.
BRIGHT_POINT_SOURCE_TELE_-APHOT		As BRIGHT_POINT_SOURCE_-APHOT, but uses telescope offsets for registration.

Point-source recipes—thermal

Recipe Name	Type of Data	Function and Comments
NOD_SELF_FLAT_NO_MASK	Bright $L < 10$; or all M, faint $L > 10$	Nod jitter, self flats of differenced pairs of frames. Has superior and fast sky subtraction. No object masking. Requires a dark. Use 4-point jitter for $L < 10$, and 8-point for fainter L and all M.
NOD_SELF_FLAT_NO_MASK_APHOT		As the previous recipe, but also performs aperture photometry of the positive and negative sources.
NOD_SKY_FLAT_THERMAL		Nod jitter, interspersed sky frames. Sky subtraction, flat field created from sky frames only. Requires a dark. 4-point jitter (8-point sequence).

Bright-point-source recipes

Recipe Name	Type of Data	Function and Comments
JITTER_SELF_FLAT	$13 < I,Z < 17$, $9 < JHK < 15$	Standard jitter, self flats. Normally 5-point jitter. Requires a dark.
JITTER_SELF_FLAT_APHOT		As JITTER_SELF_FLAT, but also performs aperture photometry of the source.
JITTER_SELF_FLAT_NO_MASK		As JITTER_SELF_FLAT but faster as it lacks object masking. It only suitable for uncrowded fields.
JITTER_SELF_FLAT_NCOLOUR_-APHOT		As JITTER_SELF_FLAT_APHOT, but produces filenames that include filters for easier identification for multi-colour observation sequences.
SKY_AND_JITTER		A sky frame and jitter on target. The sky is subtracted from the target frame before flat fielding. Requires a separate flat, as the background is too low to self flat, and a dark. No longer recommended as sky varies too quickly.
SKY_AND_JITTER_APHOT		As SKY_AND_JITTER, but also performs aperture photometry of the source.

Faint-point-source recipes

Recipe Name	Type of Data	Function and Comments
JITTER_SELF_FLAT	Faint $I,Z > 17$, $JHK > 15$	Standard jitter, self flats. Normally 9-point jitter. Requires a dark.
JITTER_SELF_FLAT_CATALOGUE		As JITTER_SELF_FLAT but produces an inventory of the locations and brightnesses of sources within the mosaic.
BRIGHT_POINT_SOURCE_-CATALOGUE		As BRIGHT_POINT_SOURCE but produces an inventory of the locations and brightnesses of sources within the mosaic. Preferred when a self-flat is not appropriate.
JITTER_SELF_FLAT_BASIC		Fastest JITTER_SELF_FLAT recipe as it lacks object masking, automatic registration and resampling.
JITTER_SELF_FLAT_NO_MASK		As JITTER_SELF_FLAT, but faster as it lacks object masking. It only suitable for uncrowded fields.

Extended-source recipes

Recipe Name	Type of Data	Function and Comments
QUADRANT_JITTER	Galaxies, quasars and nebulae of small (<45 arcsec) angular extent	4-point jitter; masks the quadrant containing the target to make the flat, then masks the objects. Requires a dark.
QUADRANT_JITTER_NO_MASK		As QUADRANT_JITTER but without object masking.
QUADRANT_JITTER_BASIC		Fastest QUADRANT_JITTER variant as it lacks object masking, automatic registration and resampling.
MOVING_QUADRANT_JITTER	Compact comets (<45 arcsec)	As QUADRANT_JITTER, but uses ephemeris data to track the non-sidereal source.
QUADRANT_JITTER_TELE		As QUADRANT_JITTER but uses telescope offsets for registration. Telescope tracks object.
EXTENDED_3x3	Galaxies and nebulae with angular extent <2 arcminutes	Sky-subtracted 3×3 grid mosaic on target. Frames alternate between sky and target. Requires a dark.
EXTENDED_3x3_BASIC		As EXTENDED_3x3 but lacks resampling and registers using telescope offsets.
EXTENDED_5x5	Galaxies and nebulae with angular extent <3 arcminutes	Sky-subtracted 5×5 grid mosaic of the target. Frames alternate between sky and target. Requires a dark.
EXTENDED_5x5_BASIC		As EXTENDED_5x5 but lacks resampling and registers using telescope offsets.

Point-source recipes—mid-infra-red

Recipe Name	Type of Data	Function and Comments
NOD_CHOP	Bright N and Q (limits to be determined)	Chopped and nodded observations to remove sky background and telescope contributions; differences both chopped beams and nodded pairs of frames giving a mosaic with two positive and two negative images. There is no masking or flat-fielding. Requires a bias in CHOP mode, *e.g.* from recipe ARRAY_TESTS.
NOD_CHOP_APHOT		As the previous recipe, but also performs aperture photometry of the combined four images (after extraction and centroid registration).
NOD_CHOP_FAINT		As NOD_CHOP, but removes column and row artifacts from the mosaic, then it combines each chopped and nodded image using a median filter to form an image of the source with four times the signal. This image is then smoothed to enhance the visibility of faint sources.
NOD_CHOP_SCAN		As NOD_CHOP, but for chopped and nodded observations that are taken in a scan pattern. A mosaic is formed at each scan position.

Moving (non-sidereal) source recipes

Recipe Name	Type of Data	Function and Comments
MOVING_JITTER_SELF_FLAT	Minor planets, comets	As JITTER_SELF_FLAT, but uses ephemeris data to track the non-sidereal source.
MOVING_JITTER_SELF_FLAT_BASIC		As JITTER_SELF_FLAT_BASIC, but uses ephemeris data to track the non-sidereal source.
JITTER_SELF_FLAT_TELE		Standard jitter, using telescope offsets. This is needed when the telescope has tracked on the non-sidereal target. Requires a dark.
MOVING_NOD_CHOP		As NOD_CHOP, but uses ephemeris data to track the non-sidereal source in the mid-infra-red.
MOVING_QUADRANT_JITTER	Compact comets	As QUADRANT_JITTER, but uses ephemeris data to track the non-sidereal source.
QUADRANT_JITTER_TELE		As QUADRANT_JITTER but uses telescope offsets for registration. This is needed when the telescope has tracked on the non-sidereal target.

Polarimetry recipes

Recipe Name	Type of Data	Function and Comments
POL_ANGLE_JITTER	Polarimetry of point or small ($<\sim$ 35 arcsec) extended sources	Makes a polarisation map from frames at the four waveplate angles at each of at least three jittered positions; the waveplate is moved before the telescope. An appropriate dark and separate flat fields at each waveplate angle (using SKY_FLAT_POL or SKY_FLAT_POL_ANGLE) must be obtained.
POL_ANGLE_NOD_CHOP	Polarimetry of point or small ($<\sim$ 10 arcsec) extended sources in mid-infra-red	Makes a polarisation map from chopped and nodded frames at the four waveplate angles at two nod positions; the waveplate is moved before the telescope is nodded. There is no object masking or flat fielding. Requires a bias in CHOP mode, *e.g.* from recipe ARRAY_TESTS.
POL_EXTENDED	Polarimetry of extended sources	Makes a polarisation map of an extended source from frames nodded between object and blank sky. The object-sky pairs must be taken at each of the four waveplate angles. Requires an appropriate dark and separate flat fields at each waveplate angle.
POL_JITTER	Polarimetry of point or small ($<\sim$ 35 arcsec) extended sources	Makes a polarimetry map from frames at the four waveplate angles at each of at least three jittered positions; the telescope is moved before the waveplate. An appropriate dark and flat fields at each waveplate angle must be obtained.
POL_NOD_CHOP	Polarimetry of point or small ($<\sim$ 10 arcsec) extended sources in mid-infra-red	As POL_ANGLE_NOD_CHOP but the telescope is nodded before the waveplate is moved.
POL_QU_FIRST_NOD_-CHOP	Polarimetry of point or small ($<\sim$ 10 arcsec) extended sources in mid-infra-red	It is a hybrid of POL_ANGLE_NOD_CHOP and POL_NOD_CHOP as the waveplate angle iterates in pairs at each jitter position.
SKY_FLAT_POL	Flat	Obtain a 'master' polarimetry flat field from the median average of eight jittered frames; the waveplate is cycled after every second frame. Makes a copy of the flat for each waveplate angle. Requires a dark.

Fabry-Perot recipes

Recipe Name	Type of Data	Function and Comments
FP	Fabry-Perot	Uses a sequence of eight frames; object-sky pairs at on-line, off-line (blue), on-line and off-line (red) FP settings to make a mosaic. Requires a separate flat field (made by SKY_FLAT_FP) and a dark.
FP_JITTER		On/Off-line images with nodding to blank sky (as FP), and spatial jittering on-source. Requires a separate flat field and a dark.
FP_JITTER_NO_SKY		On/Off-line images without nodding to blank sky (i.e. sequence of four frames), and spatial jittering on-source. Requires a separate flat field and a dark.
SKY_FLAT_FP	Flat	Creates a Fabry-Perot sky flat (from jittered blank-sky exposures) FP at on- and off-line wavelengths. Requires a dark.

D.2   Reference documentation

The following recipes apply to both UFTI and IRCAM unless otherwise noted. Where there are processing differences for the two instruments, they are noted in the reference specification. Also there is an ARRAY_TESTS recipe for each instrument, of which only UFTI's is presented below for technical reasons.

In the **Configurable Steering Parameters** sections the defaults appear at the end of the parameter's description between [ ].

The non-generic recipes of the original release are not documented here, but are still available for reducing pre-ORAC (2000 August) data. They are listed in the **Deprecated variants** section of their generic counterpart. Each behaves as the generic counterpart, except the number of jitter points is fixed. Thus the JITTER9_SELF_FLAT reduces a nine-point jitter.

ADDWCS
Creates the valid WCS in the FITS headers of raw data

Description:

This recipe uses the existing hotpotch of UKIRT and AAT imaging headers of raw data to make a new set of headers which define a valid world co-ordinate system, using the AIPS convention.

Notes:

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

Output Data :

- The corrected frame in $<$i$><$date$>$_$<$obs_number$>$_raw, where $<$i$>$ is the frame prefix. For IRIS2, the corrected frame is in $<$date$><$obs_number$>$, where $<$date$>$ is in the form ddmmm. If the file already exists, it's only updated.

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.

ARRAY_TESTS
Calculates the readout noises and dark current for UFTI

Description:

This script calculates for UFTI the NDSTARE readout noise, and the dark current from a series of four engineering frames taken with the sequence called array_tests. The results are compared with the nominal values, and you are notified whether or not the values obtained are within limits. At UKIRT, the results are also logged to an engineering file for archival purposes.

Notes:

- Intermediate frames are deleted.
- The engineering log contains the UT date and time, the NDSTARE readout noise and the dark current. The results are normally appended to the log. If for some reason it does not exist, a new log is created containing the column headings.
- Multiple array tests are permitted. A new set of results is reported and logged for each cycle.
- The NDSTARE readout noise is filed in the calibration system.

Output Data :

- The engineering log `$ORAC_DATA_OUT/ufti_array_tests.log`.

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.

BRIGHT_POINT_SOURCE
Reduces a bright-point-source photometry observation

Description:

This recipe reduces a "bright standard" photometry observation. It takes an imaging observation comprising a series of jittered object frames and a dark frame, and a predetermined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

As the name implies, it is intended for bright point sources, such as standard stars, but also any observation where using its own frames to make the flat is not appropriate.

Notes:

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.
- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- Where automatic registration is not possible, the recipe matches the centroid of central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers areupdated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.
- Sub-arrays are supported.

Output Data :

- The resultant mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 5 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_APHOT, JITTER_SELF_FLAT, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

BRIGHT_POINT_SOURCE_APHOT
Reduces a bright-point-source photometry observation and performs aperture photometry

---

Description:

This recipe reduces a "bright standard" photometry observation It takes an imaging observation comprising a series of jittered object frames and a dark frame with a predetermined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs a null debiassing, bad-pixel masking, dark subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in `$ORAC_DATA_OUT/aphot_results.txt` in the form of a Starlink small text list. The analysis of each star is appended to this file.

As the name implies, it is intended for bright point sources, such as standard stars, but also any observation of a point source where using its own frames to make the flat is not appropriate.

Notes:

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- Where automatic registration is not possible, the recipe matches the centroid of central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.

-
  The photometry uses the mode calculated from $3*$ median $-2*$ mean and Chauvenet's rejection criterion to estimate the sky level in an annulus about the source. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI, and twice the aperture (10 arcsec) for IRCAM, Michelle, and IRIS2.
  The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

- Sub-arrays are supported.

Output Data :

- The resultant mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- Results tabulation to log $ORAC_DATA_OUT/aphot_results.txt.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 5 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
BRIGHT_POINT_SOURCE, JITTER_SELF_FLAT_APHOT, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and PHOTOM.

- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

BRIGHT_POINT_SOURCE_CATALOGUE
Reduces a bright-point-source photometry observation, producing a catalogue of all sources in the field

---

Description:

This recipe reduces a "bright standard" photometry observation. It takes an imaging observation comprising a series of jittered object frames and a dark frame, and a predetermined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

Source extraction is performed only on the reduced mosaic, and uses EXTRACTOR. The results appear in $ORAC_DATA_OUT/catalogue_<group_number>.txt. No zero-point or airmass corrections are applied to the instrumental magnitudes.

As the name implies, it is intended for bright point sources, such as standard stars, but also any observation where using its own frames to make the flat is not appropriate.

Notes:

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is $ORAC_DATA_CAL/bpm.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- Where automatic registration is not possible, the recipe matches the centroid of central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers areupdated to match that of the last-observed frame contributing to the mosaic.

- The catalogue includes the right ascension and declination, instrumental apparent magnitude (calculated as $-2.5 * \log(\text{counts})$), and the error in the magnitude.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

- Sub-arrays are supported.

Output Data :

- The resultant mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The catalogue in `catalogue_<group_number>.txt`.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 5 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_APHOT, BRIGHT_POINT_SOURCE,
JITTER_SELF_FLAT_CATALOGUE, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

BRIGHT_POINT_SOURCE_NCOLOUR_APHOT
Reduces a multi-colour bright-point-source photometry observation and performs aperture photometry.

Description:

This recipe reduces a "bright standard" photometry observation observed through one or more filters. For each filter it takes an imaging observation comprising a series of jittered object frames and a dark frame with a predetermined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs a null debiasing, bad-pixel masking, dark subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in `$ORAC_DATA_OUT/aphot_results.txt` in the form of a Starlink small text list. The analysis of each star is appended to this file.

As the name implies, it is intended for bright point sources, such as standard stars, but also any observation of a point source where using its own frames to make the flat is not appropriate.

Notes:

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.
- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- Where automatic registration is not possible, the recipe matches the centroid of the central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers areupdated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.

- The photometry uses the mode calculated from 3*median-2*mean and Chauvenet's rejection criterion to estimate the sky level in an annulus about the source. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI, and twice the aperture (10 arcsec) for IRCAM and Michelle.
  The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

- Sub-arrays are supported.

Output Data :

- The resultant mosaic in <m><date>_<group_number>_<filter>_mos, where <m> is the instrument's group prefix.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- Results tabulation to `$ORAC_DATA_OUT/aphot_results.txt`.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 5 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE, BRIGHT_POINT_SOURCE_APHOT,
JITTER_SELF_FLAT, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

***

BRIGHT_POINT_SOURCE_TELE
Reduces a bright-point-source photometry observation using telescope offsets for registration.

***

Description:

This script reduces a "bright standard" photometry observation with UKIRT imaging data. It takes an imaging observation comprising a series of jittered object frames and a dark frame, and a predetermined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, registration using telescope offsets, and resampling. See the "Notes" for details.

As the name implies, it is intended for bright point sources, such as standard stars, or any observation where using its own frames to make the flat is not appropriate, and where automatic registration fails.

Notes:

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.
- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- Registration is performed using the telescope offsets transformed to pixels.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers areupdated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.
- Sub-arrays are supported.

Output Data :

- The resultant mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 5 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE, BRIGHT_POINT_SOURCE_APHOT,
JITTER_SELF_FLAT, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

CHOP_SKY_JITTER
Reduction of alternating sky-target jitters using interpolated sky subtraction

---

Description:

This recipe reduces a moderately extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. Both the sky and target frames are jittered. The recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, sky subtraction, registration, resampling, and mosaicking. The "Notes" give more details.

It is suitable for extended objects where the object fills or nearly fills the frame, so sky estimation within the frame is impossible or unreliable, but the extended mapping of the target is not required.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- For INGRID, the pre- and post-exposure images are subtracted. A non-linearity correction is then applied.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created from all the jittered sky frames, and applied to all the target frames.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile

reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- The sky subtraction comes from linear interpolation of the sky modal values of the two flat-fielded sky frames which immediately bracket the target frame.

- The field distortion of ISAAC is corrected in the target frames using the mappings documented on the ISAAC problems web page.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it then tries the crosshead offsets. If these are null, the script resorts to the telescope offsets.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- At the end of each cycle of sky and object frames the full mosaic of target frames is created and displayed. The mosaic has its bad pixels filled by interpolation. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of jittered target frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- For ISAAC, the individual bias-corrected frames in isaac<date>_<obs_number>_bc.

- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER

The number of target frames in the jitter pattern. If this is not set, a value is derived from the number of offsets, as given by header NOFFSETS. The formula is NOFFSETS / 2 − 1. An error results should NOFFSETS be odd. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
    Whether or not to create and propagate variance arrays. [0]

Related Recipes :
    CHOP_SKY_JITTER_BASIC, EXTENDED_3x3, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

CHOP_SKY_JITTER_BASIC
Basic reduction of alternating sky-target jitters using interpolated sky subtraction

Description:

This recipe reduces a moderately extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. Both the sky and target frames are jittered. The recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiassing, dark subtraction, flat-field division, sky subtraction, registration using telescope offsets, and mosaicking. The "Notes" give more details.

It is suitable for extended objects where the object fills or nearly fills the frame, so sky estimation within the frame is impossible or unreliable, but the extended mapping of the target is not required.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created from all the jittered sky frames, and applied to all the target frames.

- The sky subtraction comes from linear interpolation of the sky modal values of the two sky frames which immediately bracket the target frame.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- At the end of each cycle of sky and object frames the full mosaic of target frames is created and displayed. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of target frames in the jitter pattern. If this is not set, a value is derived from the number of offsets, as given by header NOFFSETS. The formula is NOFFSETS / 2 − 1. An error results should NOFFSETS be odd. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
CHOP_SKY_JITTER_BASIC, EXTENDED_3x3_BASIC, QUADRANT_JITTER_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

DARK_AND_BPM
Measures dark current and creates a new bad-pixel mask for UIST

Description:

This recipe is used to measure the dark current for UIST, using a long-exposure DARK frame. It first finds and bad pixels in the DARK, then measures and reports the dark current. The recipe appends to a tabulation of the dark current in an engineering log file, `$ORAC_DATA_OUT/uist_array_tests.log`, which it creates with headings if the log does not exist.

Notes:

- The recipe applies thresholds to the dark frame and flags pixels outside these limits as bad. The thresholds are derived from 3-standard-deviation clipped statistics; pixels more than 5 standard deviations above the mean are flagged.
- The bad pixels detected are added into the current bad-pixel mask and then this is filed with the calibration system as a new and current bad-pixel mask.
- The new bad-pixel mask is applied to the original dark frame, whose unclipped mean scaled by the gain and inverse exposure time is the dark current in electrons per second.

Output Data :

- The engineering log `$ORAC_DATA_OUT/uist_array_tests.log`.

Related Recipes :

ARRAY_TESTS, MEASURE_READNOISE.

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- Error propagation is not used.

DARK_SUBTRACT
Subtracts a dark frame

Description:

This recipe subtracts a corresponding dark calibration frame, then displays the result.

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.

DIFFERENCE_STATS
Calculates statistics for Michelle darks in a pairwise manner

Description:

>This recipe is meant to be used in an array tests suite. It takes a group of observations, subtracts successive pairs, then calculates and reports the standard deviation for each resulting differenced frame in a central 200-pixel-square region, and in the four channels of the detector. It finally provides the average of these statistics for the group as a whole.

Notes:

- The frames must be in the same group.

Parameters:

NUMBER = INTEGER

>The number of frames in the group. The default is used if the number of frames is fewer than 2 or is not even. [20]

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.

EXTENDED_3x3
Extended-source standard reduction using interpolated sky subtraction

Description:

This recipe reduces an extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. The target frames are arranged in an overlapping (30–50%) grid of 3×3 frames from which the recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiassing, dark subtraction, flat-field division, sky subtraction, registration, resampling, and mosaicking. The "Notes" give more details.

It is suitable for extended objects up to 2 arcminutes across with UFTI, 28 arcseconds with IRCAM, and 14 arcminutes across with IRIS2.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created for each row of the grid of target frames, and applied only to that row of target frames.

- The sky subtraction comes from linear interpolation of the sky modal values of the two flat-fielded sky frames which immediately bracket the target frame.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it then tries the crosshead offsets. If these are null, the script resorts to the telescope offsets.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- Mosaics are made and displayed for each row, except the last. At the end of each cycle of 19 frames the full mosaic of nine target frames is created and displayed instead. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The full mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each row in `<m><date>_<group_number>_mos<row_number>`, where `<row_number>` is 0 or 1.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NROW = INTEGER
The number of target frames in a row of the mosaic. Its minimum is 3 because this number of blank skies are needed to form a flat field. [3]

NCOL = INTEGER
The number of target frames in a column of the mosaic. Its minimum is 2. [3]

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
EXTENDED_3x3_BASIC, EXTENDED_5x5, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

EXTENDED_3x3_BASIC
Basic extended-source standard reduction using interpolated sky subtraction

Description:

This recipe reduces an extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. The target frames are arranged in an overlapping (30–50%) grid of 3×3 frames from which the recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiassing, dark subtraction, flat-field division, sky subtraction, registration using telescope offsets, and mosaicking. The "Notes" give more details.

It is suitable for extended objects up to 2 arcminutes across with UFTI, 28 arcseconds with IRCAM, and 14 arcminutes with IRIS2.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created for each row of the grid of target frames, and applied only to that row of target frames.

- The sky subtraction comes from linear interpolation of the sky modal values of the two flat-fielded sky frames which immediately bracket the target frame.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the

mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- Mosaics are made and displayed for each row, except the last. At the end of each cycle of 19 frames the full mosaic of nine target frames is created and displayed instead. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The full mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each row in `<m><date>_<group_number>_mos<row_number>`, where `<row_number>` is 0 or 1.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NROW = INTEGER
The number of target frames in a row of the mosaic. Its minimum is 3 because this number of blank skies are needed to form a flat field. [3]

NCOL = INTEGER
The number of target frames in a column of the mosaic. Its minimum is 2. [3]

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
EXTENDED_3x3, EXTENDED_5x5_BASIC, QUADRANT_JITTER_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

EXTENDED_5x5
Extended-source standard reduction using interpolated sky subtraction

Description:

This recipe reduces an extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. The target frames are arranged in an overlapping (30–50%) grid of 5×5 frames from which the recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, sky subtraction, registration, resampling, and mosaicking. The "Notes" give more details.

It is suitable for extended objects up to 3 arcminutes across with UFTI, 42 arcseconds with IRCAM, and 20 arcminutes with IRIS2.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created for each row of the grid of target frames, and applied only to that row of target frames.
- The sky subtraction comes from linear interpolation of the sky modal values of the two flat-fielded sky frames which immediately bracket the target frame.
- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it then tries the crosshead offsets. If these are null, the script resorts to the telescope offsets.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- Mosaics are made and displayed for each row, except the last. At the end of each cycle of 51 frames the full mosaic of 25 target frames is created and displayed instead. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The full mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each row in <m><date>_<group_number>_mos<row_number>, where <row_number> is 0 to 3.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NROW = INTEGER
 The number of target frames in a row of the mosaic. Its minimum is 3 because this number of blank skies are needed to form a flat field. [5]

NCOL = INTEGER
 The number of target frames in a column of the mosaic. Its minimum is 2. [5]

USEVAR = LOGICAL
 Whether or not to create and propagate variance arrays. [0]

Related Recipes :
 EXTENDED_3x3, EXTENDED_5x5_BASIC, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

EXTENDED_5x5_BASIC
Basic extended-source standard reduction using interpolated sky subtraction

Description:

This recipe reduces an extended source using near-infrared imaging data. The data comprise alternating blank-sky and target frames commencing and ending with a blank sky. The target frames are arranged in an overlapping (30–50%) grid of 5×5 frames from which the recipe makes a sky-subtracted untrimmed mosaic automatically.

The script performs bad-pixel masking, null debiasing, dark subtraction, flat-field division, sky subtraction, registration using telescope offsets, and mosaicking. The "Notes" give more details.

It is suitable for extended objects up to 3 arcminutes across with UFTI, 42 arcseconds with IRCAM, and 20 arcminutes with IRIS2.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is derived from the sky frames as follows. The mode (sigma-clipped mean) is used to offset each sky frame's mode to that of the first sky frame. The corrected sky frames are combined pixel by pixel using a median of the values in each frame. The resultant frame is normalised by its median to form the flat field. This frame median is subtracted from the source frames after they have been flat-fielded. A flat field is created for each row of the grid of target frames, and applied only to that row of target frames.

- The sky subtraction comes from linear interpolation of the sky modal values of the two flat-fielded sky frames which immediately bracket the target frame.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The noise will be greater in the

mosaic's peripheral areas, having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- Mosaics are made and displayed for each row, except the last. At the end of each cycle of 51 frames the full mosaic of 25 target frames is created and displayed instead. On the second and subsequent cycles the full mosaic is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The full mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each row in `<m><date>_<group_number>_mos<row_number>`, where `<row_number>` is 0 to 3.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NROW = INTEGER
The number of target frames in a row of the mosaic. Its minimum is 3 because this number of blank skies are needed to form a flat field. [5]

NCOL = INTEGER
The number of target frames in a column of the mosaic. Its minimum is 2. [5]

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
EXTENDED_3x3_BASIC, EXTENDED_5x5, QUADRANT_JITTER_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

FP
Reduces an 8-frame Fabry-Perot observation

Description:

This script reduces a Fabry-Perot observation with UFTI data. It takes an imaging observation comprising eight object frames and a dark frame to make a continuum-subtracted and sky-subtracted, untrimmed mosaic automatically.

The sequence of frames expected in the observations are tabulated below.

Frame	Position	Wavelength
1	On source	On line
2	Off source	On line
3	Off source	Off line, positive offset
4	On source	Off line, positive offset
5	On source	On line
6	Off source	On line
7	Off source	Off line, negative offset
8	On source	Off line, negative offset

It performs a null debiassing, bad-pixel masking, dark subtraction, pairwise frame differencing, flat-field division, integer shifts of origin to register, and mosaicking. The desired result is given by

$$\frac{[(F1 - F2) - (F4 - F3)] + [(F5 - F6) - (F8 - F7)]}{\text{Flatfield}}$$

where Fn is the bad-pixel masked and dark subtracted frame n. In practice, the flat field is applied to each differenced pair, such as $(F4 - F3)$, when the pair becomes available, rather than waiting until all eight frames have been observed.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- The bad-pixel mask applied is $ORAC_DATA_CAL/bpm.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- You should use SKY_FLAT_FP to make the flat field.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of eight, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in gf<date>_<group_number>_mos, where <date> is the UT date in *yyyymmdd* format.

- A mosaic for each cycle of eight in
  gf<date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in f<date>_<obs_number>_ff.

Parameters:

    NPAIRS = INTEGER
        The number of frame pairs to be differenced. It must be a multiple of 2 otherwise 4 is assumed. A value of four or more is assumed to indicate sky subtraction. [4]

    NUMBER = INTEGER
        The number of spatial jitter positions. For each spatial position there are NPAIRS pairs of frames. A value of 1 also dictates that no jittering has occurred. To make a master mosaic combining spatial positions NUMBER should be at least 3.

        If NUMBER is absent, the number of offsets, as given by internal header NOFFSETS, minus one is used. An error state arises if the resulting number of jittered frames is fewer than 3, and a default of 3 is assumed.

        If neither NUMBER nor NOFFSETS is defined, 1 is used. [1]

    USEVAR = LOGICAL
        Whether or not to create and propagate variance arrays. [0]

Related Recipes :

    SKY_FLAT_FP, FP_JITTER, FP_JITTER_NO_SKY.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

FP_JITTER
Reduces spatially jittered sets of 8-frame Fabry-Perot observations

Description:

This script reduces a Fabry-Perot observation with UFTI data. It takes an imaging observation comprising at least three sets of eight object frames, each set being for a different telescope position. The recipe combines these with a dark frame and a separate flat, to make a continuum-subtracted and sky-subtracted, untrimmed mosaic automatically.

Each sequence of eight frames expected in each spatial position are tabulated below.

Frame	Position	Wavelength
1	On source	On line
2	Off source	On line
3	Off source	Off line, positive offset
4	On source	Off line, positive offset
5	On source	On line
6	Off source	On line
7	Off source	Off line, negative offset
8	On source	Off line, negative offset

For each spatial set, the recipe performs a null debiassing, bad-pixel masking, dark subtraction, pairwise frame differencing, flat-field division, integer shifts of origin to register, and mosaicking. The wavelength-shifted mosaic is given by

$$\frac{[(F1 - F2) - (F4 - F3)] + [(F5 - F6) - (F8 - F7)]}{\text{Flatfield}}$$

where Fn is the bad-pixel masked and dark subtracted frame n. In practice, the flat field is applied to each differenced pair, such as $(F4 - F3)$, when the pair becomes available, rather than waiting until all eight frames have been observed.

Finally the recipe registers all the wavelength mosaics spatially, and forms a untrimmed mosaic, combined using the median to reduce stellar artifacts.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- You should use SKY_FLAT_FP to make the flat field.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- For each set of eight, the recipe creates a wavelength mosaic. For each cycle of spatial positions the wavelength mosaics are registered to form a spatial mosaic. For repeat cycles the spatial mosaic is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the master mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. No mosaic is trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas of the spatial having received less exposure time. Each mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `gf<date>_<group_number>_mos`, where `<date>` is the UT date in *yyyymmdd* format.

- A mosaic for each cycle of eight in
  `gf<date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `f<date>_<obs_number>_ff`.

Parameters:

NPAIRS = INTEGER
  The number of frame pairs to be differenced. It must be a multiple of 2 otherwise 4 is assumed. A value of four or more is assumed to indicate sky subtraction. [4]

NUMBER = INTEGER
  The number of spatial jitter positions. For each spatial position there are NPAIRS pairs of frames. A value of 1 also dictates that no jittering has occurred. To make a master mosaic combining spatial positions NUMBER should be at least 3.

  If NUMBER is absent, the number of offsets, as given by internal header NOFFSETS, minus one is used. An error state arises if the resulting number of jittered frames is fewer than 3, and a default of 3 is assumed.

  If neither NUMBER nor NOFFSETS are defined, 1 is used. []

USEVAR = LOGICAL
  Whether or not to create and propagate variance arrays. [0]

Related Recipes :

FP, FP_JITTER_NO_SKY, SKY_FLAT_FP.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

FP_JITTER_NO_SKY
Reduces a spatially jittered 4-frame Fabry-Perot observation

Description:

This script reduces a Fabry-Perot observation with UFTI data. It takes an imaging observation comprising at least three sets of four object frames, each set being for a different telescope position. The recipe combines these with a dark frame and a separate flat, to make a continuum-subtracted, untrimmed mosaic automatically.

Each sequence of four frames expected in each spatial position are tabulated below.

Frame	Position	Wavelength
1	On source	On line
2	On source	Off line, positive offset
3	On source	On line
4	On source	Off line, negative offset

For each spatial set, the recipe performs a null debiasing, bad-pixel masking, dark subtraction, pairwise frame differencing, flat-field division, integer shifts of origin to register, and mosaicking. The wavelength-shifted mosaic is given by

$$\frac{[(F1 - F2) - (F4 - F3)]}{\text{Flatfield}}$$

where Fn is the bad-pixel masked and dark subtracted frame n. In practice, the flat field is applied to each differenced pair, such as $(F1 - F2)$, when the pair becomes available, rather than waiting until all four frames have been observed.

Finally the recipe registers all the wavelength mosaics spatially, and forms a untrimmed mosaic, combined using the median to reduce stellar artifacts.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- You should use SKY_FLAT_FP to make the flat field.
- Registration is performed using the telescope offsets transformed to pixels.
- There is no resampling, merely integer shifts of origin.
- For each set of four, the recipe creates a wavelength mosaic. For each cycle of spatial positions the wavelength mosaics are registered to form a spatial mosaic. For repeat cycles the spatial mosaic is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the master mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. No mosaic is trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas of the spatial having received less exposure time. Each mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `gf`<date>`_`<group_number>`_mos`, where <date> is the UT date in *yyyymmdd* format.
- A mosaic for each cycle of eight in `gf`<date>`_`<group_number>`_mos`<cycle_number>, where <cycle_number> counts from 0.
- The individual flat-fielded frames in `f`<date>`_`<obs_number>`_ff`.

Parameters:

NPAIRS = INTEGER
> The number of frame pairs to be differenced. It must be a multiple of 2 otherwise 4 is assumed. A value of four or more is assumed to indicate sky subtraction. [2]

NUMBER = INTEGER
> The number of spatial jitter positions. For each spatial position there are NPAIRS pairs of frames. A value of 1 also dictates that no jittering has occurred. To make a master mosaic combining spatial positions NUMBER should be at least 3.
>
> If NUMBER is absent, the number of offsets, as given by internal header NOFFSETS, minus one is used. An error state arises if the resulting number of jittered frames is fewer than 3, and a default of 3 is assumed.
>
> If neither NUMBER nor NOFFSETS are defined, 1 is used. []

USEVAR = LOGICAL
> Whether or not to create and propagate variance arrays. [0]

Related Recipes :
> SKY_FLAT_FP, FP, FP_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

JITTER_SELF_FLAT
Reduces a "standard jitter" photometry observation using object masking

---

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiasing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe works well for faint sources and for moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- For INGRID, the pre- and post-exposure images are subtracted. A non-linearity correction is then applied.

- The dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- The field distortion of ISAAC is corrected in the target frames using the mappings documented on the ISAAC problems web page.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- For ISAAC, the individual bias-corrected frames in `isaac<date>_<obs_number>_bc`.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
   The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
   Whether or not to create and propagate variance arrays. [0]

Related Recipes :

   JITTER_SELF_FLAT_APHOT, JITTER_SELF_FLAT_BASIC,
   JITTER_SELF_FLAT_NO_MASK, JITTER_SELF_FLAT_TELE,
   MOVING_JITTER_SELF_FLAT, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

JITTER3_SELF_FLAT, JITTER5_SELF_FLAT, JITTER9_SELF_FLAT.

JITTER_SELF_FLAT_APHOT
Reduces a "standard jitter" photometry observation using object masking, and performs aperture photometry

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiasing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in `$ORAC_DATA_OUT/aphot_results.txt` in the form of a Starlink small text list. The analysis of each star is appended to this file.

This recipe works well for faint sources in moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- For INGRID, the pre- and post-exposure images are subtracted. A non-linearity correction is then applied.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- The field distortion of ISAAC is corrected in the target frames using the mappings documented on the ISAAC problems web page.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.

- The photometry uses a multiply clipped (2,2,2.5,3 standard deviations) mean to estimate the sky mode in an annulus about the source. This is not unduly biased by the presence of the self-flat artifact in the pixel histogram. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI and twice the aperture (10 arcsec) for IRCAM and IRIS2. The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- For ISAAC, the individual bias-corrected frames in `isaac<date>_<obs_number>_bc`.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_APHOT, JITTER_SELF_FLAT, JITTER_SELF_FLAT_BASIC, JITTER_SELF_FLAT_NO_MASK, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, EXTRACTOR, and PHOTOM.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

JITTER5_SELF_FLAT_APHOT.

JITTER_SELF_FLAT_BASIC
Reduces a "standard jitter" photometry observation using just the basic operations for speed

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation, flat-field division, integer shifts of origin to register, and mosaicking. See the "Notes" for further information.

This recipe aims to keep pace with the pipeline's incoming data, and is intended for faint sources and for moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised object frames using the clipped median at each pixel.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- A mosaic for each cycle of jittered frames in <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.
- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [0]

Related Recipes :

JITTER_SELF_FLAT, JITTER_SELF_FLAT_APHOT,
JITTER_SELF_FLAT_NO_MASK, JITTER_SELF_FLAT_TELE,
MOVING_JITTER_SELF_FLAT_BASIC, QUADRANT_JITTER_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

JITTER5_SELF_FLAT_BASIC, JITTER9_SELF_FLAT_BASIC.

JITTER_SELF_FLAT_CATALOGUE
Reduces a "standard jitter" photometry observation using object masking, and produces a catalogue of all sources in the field

---

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiasing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Source extraction is performed only on the reduced mosaic, and uses EXTRACTOR. The results appear in `$ORAC_DATA_OUT/catalogue_`<group_number>`.txt`. No zero-point or airmass corrections are applied to the instrumental magnitudes.

This recipe works well for faint sources and for moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- For INGRID, the pre- and post-exposure images are subtracted. A non-linearity correction is then applied.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- The field distortion of ISAAC is corrected in the target frames using the mappings documented on the ISAAC problems web page.
- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- The catalogue includes the right ascension and declination, instrumental apparent magnitude (calculated as $-2.5 * \log(\text{counts})$), and the error in the magnitude.
- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of jittered frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- For ISAAC, the individual bias-corrected frames in `isaac<date>_<obs_number>_bc`.
- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.
- The catalogue in `catalogue_<group_number>.txt`.

Parameters:

NUMBER = INTEGER
   The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
   Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_CATALOGUE, JITTER_SELF_FLAT,
JITTER_SELF_FLAT_APHOT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

JITTER_SELF_FLAT_NCOLOUR
Reduces a multi-colour "standard jitter" photometry observation using object masking

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data observed through one or more filters. For each filter it takes an observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe works well for faint sources and for moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time.

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of jittered frames for each filter in `<m><date>_<group_number>_<filter>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 5 is used. An error state arises if the number of jittered frames is fewer than 3. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
JITTER_SELF_FLAT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :
JITTER5_SELF_FLAT_NCOLOUR.

JITTER_SELF_FLAT_NCOLOUR_APHOT
Reduces a multi-colour "standard jitter" photometry observation using object masking, and performs aperture photometry

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data observed through one or more filters. For each filter it takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiasing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in $ORAC_DATA_OUT/aphot_results.txt in the form of a Starlink small text list. The analysis of each star is appended to this file.

This recipe works well for faint sources in moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is $ORAC_DATA_CAL/bpm.

- For INGRID, the pre- and post-exposure images are subtracted. A non-linearity correction is then applied.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- The field distortion of ISAAC is corrected in the target frames using the mappings documented on the ISAAC problems web page.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.

- The photometry uses a multiply clipped (2,2,2.5,3 standard deviations) mean to estimate the sky mode in an annulus about the source. This is not unduly biased by the presence of the self-flat artifact in the pixel histogram. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI and twice the aperture (10 arcsec) for IRCAM and IRIS2. The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in
  `<m><date>_<group_number>_<filter>_mos<cycle_number>`, where
  `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- For ISAAC, the individual bias-corrected frames in `isaac<date>_<obs_number>_bc`.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
> The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
> Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_APHOT, JITTER_SELF_FLAT,
JITTER_SELF_FLAT_BASIC, JITTER_SELF_FLAT_NO_MASK,
QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, EXTRACTOR, and PHOTOM.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

JITTER_SELF_FLAT_NO_MASK
Reduces a "standard jitter" photometry observation without object masking

---

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe works well for faint sources and sparse fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised object frames using the clipped median at each pixel.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header.

Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- A mosaic for each cycle of jittered frames in <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.
- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
JITTER_SELF_FLAT, JITTER_SELF_FLAT_APHOT,
JITTER_SELF_FLAT_BASIC, JITTER_SELF_FLAT_TELE,
QUADRANT_JITTER_NO_MASK.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :
JITTER5_SELF_FLAT_NO_MASK, JITTER9_SELF_FLAT_NO_MASK.

JITTER_SELF_FLAT_TELE
Reduces a "standard jitter" photometry observation using object masking, and telescope offsets for registration

Description:

This script reduces a "standard jitter" photometry observation with near-infrared imaging data. It takes an observation comprising jittered object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation and division, registration using telescope offsets, and resampling. See the "Notes" for further information.

This recipe works well for faint sources and for moderately crowded fields. It is also used for observations that track a moving object.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- Intermediate frames are deleted except for the flat-fielded (_ff suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- A mosaic for each cycle of jittered frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.
- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [0]

Related Recipes :

JITTER_SELF_FLAT, JITTER_SELF_FLAT_APHOT,
JITTER_SELF_FLAT_BASIC, JITTER_SELF_FLAT_NO_MASK,
MOVING_JITTER_SELF_FLAT, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

JITTER9_SELF_FLAT_TELE.

LAMP_FLAT
Creates and files imaging flat fields derived from a calibration lamp

---

Description:

This recipe makes one or more flats for ESO infrared imaging from a series of internal flat frames with the calibration lamp alternating on then off. A new flat is made for each combination of filter.

It performs a null debiassing, bad-pixel masking, then differences each pair of frames. Once all pairs have been so processed, these are then treated like sky flats; normalised frames are combined pixel by pixel using the median. Details of each flat are filed in the index of flats for future selection and use of the flat. See the "Notes" for further details.

Notes:

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- Intermediate frames are deleted.
- Sub-arrays are supported.

Output Data :

- The created flat field in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent recipe cycles. Token `<filter>` is the filter name, `<group_number>` is the frame number of the group, and `<cycle_number>` is the number of the cycle, counting from one.
- The flats are filed in `$ORAC_DATA_OUT/index.flat`.

Parameters:

NUMBER = INTEGER

The number of frames in the group. If absent, the number of offsets, as given by header `HIERARCH.ESO.TPL.NEXP`. If neither is available, 6 is used. An error state arises if the number of jittered frames is fewer than 6 or is odd numbered. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [1]

References :

ISAAC Data Reduction Guide 1.5, P. Amico et al., 2002.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK and KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

MAKE_BPM
Creates and files a bad-pixel mask from a long-exposure dark

Description:

This recipe reduces a long-exposure dark-frame observation of infrared imaging data to create a bad-pixel mask. It files the mask in the mask index file. Reduction comprises only thresholding the pixel values about a clipped mean using a multiple of the clipped standard deviation.

Notes:

- The dark must have a minimum exposure of 20 seconds.
- Clipping is at 2,3,3 standard deviations.
- Bad values are deemed to be those beyond the range of the clipped mean $+/- 5$ standard deviations.
- Intermediate frames are deleted.
- Sub-arrays are supported.

Output Data :

- The bad-pixel mask is called `bpm_<frame_number>`.
- The bad-pixel mask is filed in `$ORAC_DATA_OUT/index.mask`.

Related Recipes :

DARK_AND_BPM.

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is `"bpm_<frame_number>"`.
- Error propagation is not used.

MEASURE_READNOISE
Measures and files the readnoise for UIST from a set of dark frames

Description:

This recipe measures the readnoise for a group of five short-exposure UIST DARK frames and files that measurement with the calibration system. It also determines the readnoise variance. The readnoise result is compared with the nominal value; you are notified whether or not the value is within acceptable limits. The recipe appends a tabulation of the readnoise and its variance in a log file, `$ORAC_DATA_OUT/uist_array_tests.log`, which it creates with headings if the log does not exist.

Notes:

- The first dark frame is used to 'clean up' the array before statistics are done on the remaining frames.
- The recipe calculates the readnoise as follows. It first derives the population variance estimate (PVE) of the dark frames, calculated on a per-pixel basis. It then finds the square root of the mean of this PVE image. If the images were taken before UT 2002 December 2, the readnoise is calculated as the product of the square root of the mean of the PVE, the number of reads minus one, the read interval, and the the gain, divided by the number of multiple reads. Otherwise, the readnoise is formed by multiplying the gain by the square root of the mean.
- The readnoise is nominal if it falls between 38 and 45 e-/second.

Output Data :

- The engineering log `$ORAC_DATA_OUT/uist_array_tests.log`.

Parameters:

DARK_FRAMES = INTEGER

The number of dark frames to combine. The maximum allowed is 9. [5]

Related Recipes :

ARRAY_TESTS, DARK_AND_BPM.

Implementation Status:

- The processing engines are from the Starlink packages KAPPA and CCDPACK.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- Error propagation is not used.

MOVING_JITTER_SELF_FLAT
Reduces a "standard jitter" photometry observation of a moving target using object masking

---

Description:

This script reduces a "standard jitter" photometry observation with UKIRT imaging data. It takes an observation comprising jittered object frames of a moving target and a dark frame to make automatically a calibrated, untrimmed mosaic in the reference frame of the target.

It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Registration is adjusted to track the motion of the moving target using ephemeris data stored in file `target_ephem.dat`. See "Ephemeris-file Format" for details of this file's format.

This recipe works well for faint moving sources and in moderately crowded fields. It should not be used for frames where the telescope guided on the moving object. In that case reduction should be performed by JITTER_SELF_FLAT_TELE which registers using the telescope offsets alone.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat field is created by combining normalised object frames using the median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels. Once the offsets are determined, they are

adjusted for the motion of the target, so that the final mosaic registers the target, not the background stars.

- The ephemeris file is specified by environment variable `ORAC_EPHEMERIS`, defaulting to `$ORAC_DATA_OUT/target_ephem.dat`.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Ephemeris-file Format :

The current format of the ephemeris file is one line per object comprising three space-separated fields in the following order:

- the objectname, which may contain embedded spaces;

- the motion in the plane of the sky in arcsec/second for right ascension then declination.

Note that the right ascension motion is the change in right ascension multiplied by the cosine of the declination. The format will change to include UT and possibly date.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in
<m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in `flat_`<filter>`_`<group_number> for the first or only cycle, and `flat_`<filter>`_`<group_number>`_c`<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER
>   The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
>   Whether or not to create and propagate variance arrays. [0]

Related Recipes :

>   JITTER_SELF_FLAT, JITTER_SELF_FLAT_TELE, MOVING_JITTER_SELF_FLAT_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

>   MOVING_JITTER9_SELF_FLAT.

MOVING_JITTER_SELF_FLAT_BASIC
Reduces a "standard jitter" photometry observation of a moving target using just the basic operations for speed

---

Description:

> This script reduces a "standard jitter" photometry observation with UKIRT imaging data. It takes an observation comprising jittered object frames of a moving target and a dark frame to make automatically a calibrated, untrimmed mosaic in the reference frame of the target.

> It performs a null debiassing, bad-pixel masking, dark subtraction, flat-field creation and division, amd integer shifts of pixel origin to register to fixed sky co-ordinates. See the "Notes" for further information.

> The registration is adjusted to track the motion of the moving target using ephemeris data stored in file `target_ephem.dat`. See "Ephemeris-file Format" for details of this file's format.

> This recipe aims to keep pace with the pipeline's incoming data. It works well for faint moving sources and in moderately crowded fields. It should not be used for frames where the telescope guided on the moving object. In that case reduction should be performed by JITTER_SELF_FLAT_TELE which registers using the telescope offsets alone.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised object frames using the median at each pixel.

- Registration is performed using the telescope offsets transformed to pixels. Once the offsets are determined, they are adjusted for the motion of the target, so that the final mosaic registers the target, not the background stars.

- There is no resampling, merely integer shifts of origin.

- The ephemeris file is specified by environment variable `ORAC_EPHEMERIS`, defaulting to `$ORAC_DATA_OUT/target_ephem.dat`.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Ephemeris-file Format :

The current format of the ephemeris file is one line per object comprising three space-separated fields in the following order:

- the objectname, which may contain embedded spaces; and

- the motion in the plane of the sky in arcsec/second for right ascension then declination.

Note that the right-ascension motion is the change in right ascension multiplied by the cosine of the declination. The format may change to include UT and possibly date.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of jittered frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER

The number of frames in the jitter pattern. If not supplied the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 9 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

Related Recipes :

JITTER_SELF_FLAT_BASIC, JITTER_SELF_FLAT_TELE,
MOVING_JITTER_SELF_FLAT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

MOVING_JITTER9_SELF_FLAT_BASIC.

MOVING_NOD_CHOP
Reduces a chopped and nodded observation of a moving target

Description:

    This script reduces a chopped and nodded observation, currently just for Michelle data. It takes an imaging observation comprising a multiple-of-four object frames to make automatically a calibrated, untrimmed mosaic in the reference frame of a moving target.

    It performs a null debiassing, creation and propagation of data variance, difference the integrations for each AB chop beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

    Registration is adjusted to track the motion of the moving target using ephemeris data stored in file `target_ephem.dat`. See "Ephemeris-file Format" for details of this file's format.

Notes:

- A variance array is created for each beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixel, adjusted for the motion of the target, so that the final mosaic registers the target, not the background stars.

- The ephemeris file is specified by environment variable `ORAC_EPHEMERIS`, defaulting to `$ORAC_DATA_OUT/target_ephem.dat`.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- Intermediate frames are deleted except for the differenced pairs (_dp suffix) frames.

Ephemeris-file Format :

The current format of the ephemeris file is one line per object comprising three space-separated fields in the following order:

- the objectname, which may contain embedded spaces;
- the motion in the plane of the sky in arcsec/second for right ascension then declination.

Note that the right ascension motion is the change in right ascension multiplied by the cosine of the declination. The format will change to include UT and possibly date.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.
- A mosaic for each cycle of object frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number>
  counts from 0.
- The differenced pairs in <i><date>_<obs_number>_dp, where <i> is the frame prefix.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

NOD_CHOP, MOVING_JITTER_SELF_FLAT, MOVING_QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

MOVING_QUADRANT_JITTER
Reduces a "Quadrant Jitter" observation, of a moving target including object masking

---

Description:

This script reduces a "quadrant jitter" photometry observation with UKIRT imaging data. It takes an imaging observation comprising one or more series of four object frames where the target is approximately centred in each quadrant; and a dark frame to make automatically a calibrated, untrimmed mosaic in the reference frame of the moving target.

It performs bad-pixel masking, null debiassing, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Registration is adjusted to track the motion of the moving target using ephemeris data stored in file `target_ephem.dat`. See "Ephemeris-file Format" for details of this file's format.

This recipe works well for extended moving sources (comets), whose extent does not exceed 45 arcseconds for UFTI or 10 arcseconds for IRCAM, in moderately crowded fields. Sources may include those with a comparatively bright core embedded in faint extended emission. The object need not be isolated, as the recipe masks objects within the other quadrants, and hence does not introduce significant artifacts into the flat field. This recipe should not be used for frames where the telescope guided on the moving object. In that case reduction should be performed by QUADRANT_JITTER_TELE, which registers using the telescope offsets alone.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First the quadrant containing the object is masked in each object frame. Second an approximate flat field is created by combining the normalised and masked object frames using the clipped median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are

detected, and masked in the dark-subtracted frames. The second stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it matches the centroid of the central source. If this fails, the script resorts to using the telescope offsets transformed to pixels. Once the offsets are determined, they are adjusted for the motion of the target, so that the final mosaic registers the target, not the background stars.
- The ephemeris file is specified by environment variable `ORAC_EPHEMERIS`, defaulting to `$ORAC_DATA_OUT/target_ephem.dat`.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame. Thus the noise will be greater in the peripheral areas having received less exposure time. The full signal will be in the central ninth containing the main object. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- For each cycle of four, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Ephemeris-file Format :

The current format of the ephemeris file is one line per object comprising three space-separated fields in the following order:

- the objectname, which may contain embedded spaces;
- the motion in the plane of the sky in arcsec/second for right ascension then declination.

Note that the right ascension motion is the change in right ascension multiplied by the cosine of the declination. The format will change to include UT and possibly date.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of four in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is f for UFTI and i for IRCAM, and u for UIST.
- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

>   **USEVAR = LOGICAL**
>
>>   Whether or not to create and propagate variance arrays. [0]

Related Recipes :

>   MOVING_JITTER_SELF_FLAT, QUADRANT_JITTER, QUADRANT_JITTER_TELE.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

NIGHT_LOG
Produces a text file log of a night's observations

Description:

This recipe takes a night's observations, and creates a text file containing a headed tabulation of parameters for each frame.

The parameters are: observation number, object name, observation type, UT start time, exposure time, number of coadds, read mode and speed, filter, start airmass, frame dimensions in pixels, base equatorial co-ordinates, and data-reduction recipe name.

Notes:

- Run with `oracdr -noeng -nodisplay -from 1 -skip` for efficiency.
- The <date> comes from the header keyword DATE.
- Specification provided by Sandy Leggett.

Output Data :

- The text log file `$ORAC_DATA_IN/`<date>`.nightlog`, where <date> is the UT date.

NOD_CHOP
Reduces a chopped and nodded observation

Description:

This script reduces a chopped and nodded observation, currently just for Michelle data. It takes an imaging observation comprising a multiple-of-four object frames to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, creation and propagation of data variance, difference the integrations for each AB chop beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

Notes:

- A variance array is created for each beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of object frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The differenced pairs in `<i><date>_<obs_number>_dp`, where `<i>` is the frame prefix.

Parameters:

NUMBER = INTEGER

The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [1]

Related Recipes :

NOD_CHOP_APHOT, NOD_SELF_FLAT_NO_MASK.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format and multi-NDF HDS container files.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

NOD_CHOP_APHOT
Reduces a chopped and nodded observation, and performs aperture photometry

---

Description:

This script reduces a chopped and nodded observation, currently just for Michelle data. It takes an imaging observation comprising a multiple-of-four object frames to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, creation and propagation of data variance, difference the integrations for each AB chop beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

The script combines and registers the various chopped and nodded images of the point source and neighbouring background to form to form a single image with four times the signal. Photometry of the point source using a fixed 5-arcsecond aperture is then calculated. The results appear in `$ORAC_DATA_OUT/aphot_results.txt` in the form of a Starlink small text list. The analysis of each star is appended to this file.

Notes:

- A variance array is created for each beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The

exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The combined source image is made by taking symmetrical areas about each source, such that no pixels are duplicated. Thus the divisions occur at midpoints of the chop throw and the nod separations. These are registered using the source centroids.
- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.
- The photometry uses a multiply clipped (2,2,2.5,3 standard deviations) mean to estimate the sky mode in an annulus about the source. The inner annulus diameter is 1.5 times that of the aperture (7.5 arcsec); the outer annulus is 3.0 times (15 arcsec) for Michelle. The errors are internal, based on the sky noise.
- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of object frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The combined source image and neighbourhoods in `<m><date>_<group_number>_cab`.
- The differenced pairs in `<i><date>_<obs_number>_dp`, where `<i>` is the frame prefix.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :
NOD_CHOP, NOD_SELF_FLAT_NO_MASK_APHOT,
BRIGHT_POINT_SOURCE_APHOT, JITTER_SELF_FLAT_APHOT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and PHOTOM.
- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

NOD_CHOP_FAINT
Reduces a chopped and nodded observation of a faint source, combining images and smoothing

Description:

This script reduces a chopped and nodded observation of a faint point or compact source, currently just for Michelle data. It takes an imaging observation comprising a multiple-of-four object frames to make a calibrated, smoothed combined image of the source automatically.

It performs a null debiassing, creation and propagation of data variance, difference the integrations for each AB chop beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic containing positive and negative images of the source. Column and row patterns are filtered.

The script extracts the various chopped and nodded images of the source and neighbouring background from the mosaic. It combines them using a median filter at each pixel to form to form a single image of the source with four times the signal. This combined frame is smoothed to enhance the visibility of faint sources.

See the "Notes" for further information.

Notes:

- A variance array is created for each beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Pickup and bias variation patterns, evident as ripples in the rows or bands in the columns respectively, are removed by subtracting the median along each column or row from the pixels in that column or row.

- The combined source image is made by taking symmetrical areas about the expected position of each source (derived from the chop throw and the nod separations), corrected for a shift of the base location from its nominal position. The shift comes from centroiding on bright sources with recipe NOD_CHOP_APHOT. The areas extend such that no pixels are duplicated. Thus the divisions occur at midpoints between the four images.

- The combined source image is smoothed using a 4-by-4 pixel block-average filter.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix), and the bias- and pickup-corrected frames (`_cpc` and `_rpc` suffices).

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of object frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The combined source image and neighbourhoods in
  <m><date>_<group_number>_cab. The smoothed combined image in
  <m><date>_<group_number>_scab.

- The differenced pairs in <i><date>_<obs_number>_dp, where <i> is the frame prefix.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

NOD_CHOP, NOD_CHOP_APHOT, NOD_SELF_FLAT_NO_MASK.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

NOD_CHOP_SCAN
Reduces a chopped and nodded observation in a scan pattern

Description:

This script reduces chopped and nodded observation, currently just for Michelle data. It takes an imaging observation comprising a multiple-of-four object frames at a series of scan offset positions to make calibrated, untrimmed mosaics at each scan position automatically.

It performs a null debiasing, creation and propagation of data variance, difference the integrations for each AB chop beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic at each scan position. See the "Notes" for further information.

Notes:

- A variance array is created for each beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- The telescope offsets of the first frame in each multiple-of-four frames define scan position. The recipe creates a mosaic at each distinct pair of offsets.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames at each distinct scan position, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise for that scan position. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end

UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

-
    The integrated mosaics in <m><date>_<group_number>_o<scan>_mos, where <m> is the instrument's group prefix, <group_number> is the number of group, and <scan> is the index number of the distinct scan position counting from 0.

- A mosaic for each cycle of object frames in <m><date>_<group_number>_o<scan>_mos<cycle_number>, where <cycle_number> counts from 0.

- The differenced pairs in <i><date>_<obs_number>_dp, where <i> is the frame prefix.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

NOD_CHOP, NOD_SELF_FLAT_NO_MASK_APHOT,

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format and multi-NDF HDS container files.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

NOD_SELF_FLAT_NO_MASK
Reduces a "nod jitter" observation

Description:

This script reduces a "nod jitter" observation with UKIRT imaging data. It takes an imaging observation comprising a multiple-of-four object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, difference adjacent pairs, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe works well for faint sources in moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised object frames using the median at each pixel. There is no cleaning of extreme outliers.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.
- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of object frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- For ISAAC, the individual bias-corrected frames in `isaac<date>_<obs_number>_bc`.
- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :
BRIGHT_POINT_SOURCE, NOD_SELF_FLAT_NO_MASK_APHOT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :
NOD4_SELF_FLAT_NO_MASK, NOD8_SELF_FLAT_NO_MASK.

NOD_SELF_FLAT_NO_MASK_APHOT
Reduces a "nod jitter" photometry observation, and performs aperture photometry

---

Description:

This script reduces a "nod jitter" photometry observation with UKIRT imaging data. It takes an imaging observation comprising a multiple-of-four object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, difference adjacent pairs, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in $ORAC_DATA_OUT/aphot_results.txt in the form of a Starlink small text list. The analysis of each star is appended to this file.

This recipe works well for faint sources in moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is $ORAC_DATA_CAL/bpm.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised object frames using the median at each pixel. There is no cleaning of extreme outliers.

- For ISAAC, residual bias variations along the columns are largely removed from each flat-fielded frame. The recipe first masks the sources, then collapses the frame along its rows to form a profile, whose clipped mean is subtracted. The resultant profile reflects the bias variations. The recipe subtracts this profile from each column of the flat-fielded frame.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. To discriminate between the various results, the positive images have suffix _pos after the frame name and the negative images have a _neg suffix. There are headings at the top of each column.

- The photometry uses a multiply clipped (2,2,2.5,3 standard deviations) mean to estimate the sky mode in an annulus about the source. This is not unduly biased by the presence of the self-flat artifact in the pixel histogram. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI and twice the aperture (10 arcsec) for IRCAM. The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (_ff suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of object frames in
  <m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- For ISAAC, the individual bias-corrected frames in isaac<date>_<obs_number>_bc.

- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE_APHOT, NOD_SELF_FLAT_NO_MASK.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

NOD4_SELF_FLAT_NO_MASK_APHOT, NOD8_SELF_FLAT_NO_MASK_APHOT.

NOD_SKY_FLAT_THERMAL
Reduces a "nod jitter" observation creating a flat from sky frames

---

Description:

This script reduces a "nod jitter" observation with UKIRT imaging data. It takes an imaging observation comprising a multiple-of-eight object frames and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs a null debiassing, bad-pixel masking, dark subtraction, difference adjacent pairs, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe works well for faint sources in moderately crowded fields.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created by combining normalised sky frames using the median at each pixel. There is no cleaning of extreme outliers.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of object frames, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- A mosaic for each cycle of object frames in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.
- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 8 is used. An error state arises if the number of jittered frames is fewer than 8 and not a multiple of 8. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

BRIGHT_POINT_SOURCE, NOD_SELF_FLAT_NO_MASK.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

POL_ANGLE_JITTER
Reduces an imaging polarimetry observation, where waveplate angle iterates at each jitter position

Description:

This script reduces a polarimetry observation with UKIRT imaging data. It takes an imaging observation comprising object frames at the four waveplate angles 0, 45, 22.5, 67.5 degrees for each of of a series of jitter positions offset in Right Ascension; and a dark frame to make calibrated polarisation images and vectors automatically. See "Output Data" for a list of these images.

It performs a null debiasing, bad-pixel masking, dark subtraction and flat-field division on all frames. Next the sections of the frame representing the e- and o-beam target and sky regions are extracted, and the target frames sky-subtracted. The resultant frames undergo registration and resampling to form a mosaic for each waveplate angle and beam. Once all eight mosaics are formed they are registered and resampled, and then combined to form the various polarisation images. The polarisation data are binned and noisy data excluded from a final catalogue of vectors. See the "Notes" for details.

This recipe works well for point sources, and for extended sources whose sizes in Right Ascension and Declination are less than about 35 and 15 arcseconds respectively for UFTI, or 9 and 4 arcseconds for IRCAM. Objects which would appear in both the target and sky regions, *i.e.* Declination extents south of the centre larger than 35 arcseconds (UFTI) or 8 arcseconds (IRCAM), should use recipe POL_EXTENDED for best results.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- Data errors are propagated through all processing steps. The initial values are found by applying the nominal ADU conversion and read noise.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- You should use SKY_FLAT_POL or SKY_FLAT_POL_ANGLE to make the flat fields.

- The target regions are 30% to 70% of the frame width about the Right-ascension centre, *i.e.* roughly centred on the source. The current sky limits are 1% to 99% of

the frame width along the Right-ascension axis. The Declination pixel limits are instrument dependent, and are as follows. For UFTI, o sky: 69–264; e sky: 320–484; o target: 601–764; e target: 824–988. For IRCAM, o sky: 12–52; e sky: 67–107; o target: 152–192; e target: 207–247.

- The sky subtraction for a beam uses a constant modal sky level from the corresponding sky region.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects for automatic registration, the recipe matches the centroid of central source within an 8-arcsecond box. Should that fail for the jittered e- and o-beam sections, the recipe resorts to using the telescope offsets transformed to pixels. However, the final option for registering the e and o-beam mosaics at different waveplate angles, uses the beam offsets in arcseconds for the current filter converted to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the few pixels in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of twelve frames, the recipe creates mosaics for each beam and waveplate angle. Each mosaic has its bad pixels filled and after the first cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 6.3 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north. A non-null value will be applied once it is determined.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames and the mosaics (`_mos` or `_mos_c`<cycle_number> suffix).

Output Data :

- The integrated mosaics in `<m><date>_<group_number>_<beam><angle>_mos`, where `<m>` the instrument's group prefix. Token `<beam>` is e or o; and `<angle>` is 0, 22, 45, or 67.

- A mosaic for each cycle of jittered frames per beam and angle in `<m><date>_<group_number>_<beam><angle>_mos_c<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- Polarisation frames `<m><date>_<group_number>_<suffix>`, each with a different suffix for the each parameter. The suffices are:

I	intensity
P	percentage polarisation
PI	polarisation intensity
Q	Stokes Q
TH	polarisation angle
U	Stokes U

- A FITS binary-table catalogue of the binned and culled polarisation data, called `<m><date>_<group_number>_I.FIT`. For each point it tabulates the x-y co-ordinates, the total intensity, the Stokes parameters, the percentage polarisation, the polarisation angle and intensity. There are additional columns giving the standard deviation on each of the tabulated values (excluding the co-ordinates). Likewise `<m><date>_<group_number>_all.FIT` and `<m><date>_<group_number>_bin.FIT` store the full and binned catalogues respectively.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern, per waveplate angle. If this is not set, the number of offsets, as given by FITS header NOFFSETS, minus one is used. If neither is available, 3 is the default. An error state arises if the number of jittered frames is fewer than 3. For observations prior to the availability of full ORAC, header NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

POL_EXTENDED, POL_JITTER, SKY_FLAT_POL, SKY_FLAT_POL_ANGLE.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, POLPACK, FIGARO, and CURSA.

- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.
- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.
- The units are set for the frames with suffices (see "Output Data") `P` to `%`, and `TH` to `degrees`.

POL_ANGLE_NOD_CHOP
Reduces a chopped and nodded polarimetry observation, where waveplate angle iterates at each jitter position

---

Description:

This script reduces a chopped and nodded single-beam polarimetry observation, currently just for Michelle data. The imaging observation should comprise chopped object frames at the four waveplate angles 0, 45, 22.5, 67.5 degrees for each of a multiple-of-two pairs of nod positions. For each waveplate angle the recipe makes automatically a calibrated, untrimmed mosaic. The recipe combines the multiple images of the source within each of these mosaics into new frames, and uses those four combined frames to calculate automatically calibrated polarisation images and vectors of the source. See "Output Data" for a list of these images.

It performs a null debiasing, creation and propagation of data variance, difference the integrations for each AB chop-beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

Notes:

- A variance array is created for each chop beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two chop beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of frames, the recipe creates mosaics for each chop beam and waveplate angle (modulo 180 degrees). Each mosaic has its bad pixels filled and after the first

cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- For each waveplate angle, the combined source image is made by taking symmetrical areas about each source, such that no pixels are duplicated. Thus the divisions occur at midpoints of the chop throw and the nod separations. These are registered using the source centroids.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 0.0 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north. A non-null value will be applied once it is determined.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_NOD_CHOP_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

- The integrated mosaics in `<m><date>_<group_number>_p<angle>_mos`, where `<m>` is the instrument's group prefix; and `<angle>` is 0, 22, 45, or 67.

- A mosaic for each cycle of chopped and nodded frames per waveplate angle in `<m><date>_<group_number>_p<angle>_mos_c<cycle_number>`, where `<cycle_number>` counts from 0.

- The combined source image and neighbourhoods at each waveplate angle in `<m><date>_<group_number>_p<angle>_cab`.

- The differenced pairs in `<i><date>_<obs_number>_dp`, where `<i>` is the frame prefix.

- Polarisation frames `<m><date>_<group_number>_<suffix>`, each with a different suffix for the each parameter. The suffices are:

> I intensity
>
> P percentage polarisation
>
> PI polarisation intensity
>
> Q Stokes *Q*
>
> TH polarisation angle
>
> U Stokes *U*

- A FITS binary-table catalogue of the binned and culled polarisation data, called `<m><date>_<group_number>_I.FIT`. For each point it tabulates the *x-y* co-ordinates, the total intensity, the Stokes parameters, the percentage polarisation, the polarisation angle and intensity. There are additional columns giving the standard deviation on each of the tabulated values (excluding the co-ordinates). Likewise `<m><date>_<group_number>_all.FIT` and `<m><date>_<group_number>_bin.FIT` store the full and binned catalogues respectively.

Parameters:

NUMBER = INTEGER

The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [1]

Related Recipes :

POL_NOD_CHOP, POL_QU_FIRST_NOD_CHOP, NOD_CHOP_APHOT, POL_ANGLE_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, POLPACK, FIGARO, and CURSA.
- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.
- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.
- The units are set for the frames with suffices (see "Output Data") P to `%`, and TH to `degrees`.
- Error propagation is controlled by the USEVAR parameter.

POL_EXTENDED
Reduces an imaging polarimetry observation of an extended source

Description:

This script reduces a polarimetry observation with UKIRT imaging data. It takes an imaging observation comprising alternating object and sky frames at the four waveplate angles 0, 45, 22.5, 67.5 degrees in turn, then jittered to at least three positions; and a dark frame to make calibrated polarisation images and vectors automatically. See "Output Data" for a list of these images.

It performs a null debiasing, bad-pixel masking, dark subtraction and flat-field division on all frames. Next the sections of the target frame representing the e- and o-beam target regions are extracted and sky-subtracted. The sky levels are determined from the two corresponding regions for each beam in the following sky frame. The resultant frames undergo registration and resampling to form a mosaic for each waveplate angle and beam. Once all eight mosaics are formed they are registered and resampled, and then combined to form the various polarisation images. The polarisation data are binned and noisy data excluded from a final catalogue of vectors. See the "Notes" for details.

This recipe is intended for extended sources whose sizes are more than about 35 arcseconds respectively for UFTI, or 8 arcseconds for IRCAM.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- Data errors are propagated through all processing steps. The initial values are found by applying the nominal ADU conversion and read noise.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- You should use SKY_FLAT_POL or SKY_FLAT_POL_ANGLE to make the flat fields.

- The target regions are 10% to 90% of the frame width about the Right-ascension centre, *i.e.* roughly centred on the source. The current sky limits are 1% to 99% of the frame width along the Right-ascension axis. The Declination pixel limits are instrument dependent, and are as follows. For UFTI, o sky: 69–264; e sky: 320–484; o target: 601–764; e target: 824–988. For IRCAM, o sky: 12–52; e sky: 67–107; o target: 152–192; e target: 207–247.

- The sky subtraction for a beam uses a constant modal sky level from the corresponding sky regions.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects for automatic registration, the recipe matches the centroid of central source within an 8-arcsecond box. Should that fail for the jittered e- and o-beam sections, the recipe resorts to using the telescope offsets transformed to pixels. However, the final option for registering the e and o-beam mosaics at different waveplate angles, uses the beam offsets in arcseconds for the current filter converted to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the few pixels in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of twelve frames, the recipe creates mosaics for each beam and waveplate angle. Each mosaic has its bad pixels filled and after the first cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 6.3 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames and the mosaics (`_mos` or `_mos_c<cycle_number>` suffix).

Output Data :

- The integrated mosaics in `<m><date>_<group_number>_<beam><angle>_mos`, where `<m>` the instrument's group prefix. Token `<beam>` is e or o; and `<angle>` is 0, 22, 45, or 67.

- A mosaic for each cycle of jittered frames per beam and angle in
  <m><date>_<group_number>_<beam><angle>_mos_c<cycle_number>, where
  <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i>
  is the frame prefix. The naming format is slightly different for some non-UKIRT
  instruments.

- Polarisation frames <m><date>_<group_number>_<suffix>, each with a differ-
  ent suffix for the each parameter. The suffices are:

I	intensity
P	percentage polarisation
PI	polarisation intensity
Q	Stokes Q
TH	polarisation angle
U	Stokes U

- A FITS binary-table catalogue of the binned and culled polarisation data, called
  <m><date>_<group_number>_I.FIT. For each point it tabulates the x-y co-ordinates,
  the total intensity, the Stokes parameters, the percentage polarisation, the polarisation
  angle and intensity. There are additional columns giving the standard deviation on
  each of the tabulated values (excluding the co-ordinates). Likewise
  <m><date>_<group_number>_all.FIT and
  <m><date>_<group_number>_bin.FIT store the full and binned catalogues re-
  spectively.

Parameters:

    NUMBER = INTEGER

    The number of frames in the jitter pattern, per waveplate angle. If this is not set,
the number of offsets, as given by FITS header NOFFSETS, minus one is used. If
neither is available, 3 is the default. An error state arises if the number of jittered
frames is fewer than 3. For observations prior to the availability of full ORAC, header
NOFFSETS will be absent. []

    USEVAR = LOGICAL

    Whether or not to create and propagate variance arrays. [1]

Related Recipes :

    POL_ANGLE_JITTER, SKY_FLAT_POL, SKY_FLAT_POL_ANGLE.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, POLPACK,
  FIGARO, and CURSA.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.

- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.

- The units are set for the frames with suffices (see "Output Data") P to `%`, and TH to `degrees`.

POL_JITTER
Reduces an imaging polarimetry observation jittered at each angle

Description:

This script reduces a polarimetry observation with UKIRT imaging data. It takes an imaging observation comprising object frames jittered in Right Ascension at the four waveplate angles 0, 45, 22.5, 67.5 degrees in turn; and a dark frame to make calibrated polarisation images and vectors automatically. See "Output Data" for a list of these images.

It performs a null debiasing, bad-pixel masking, dark subtraction and flat-field division on all frames. Next the sections of the frame representing the e- and o-beam target and sky regions are extracted, and the target frames sky-subtracted. The resultant frames undergo registration and resampling to form a mosaic for each waveplate angle and beam. Once all eight mosaics are formed they are registered and resampled, and then combined to form the various polarisation images. The polarisation data are binned and noisy data excluded from a final catalogue of vectors. See the "Notes" for details.

This recipe works well for point sources, and for extended sources whose sizes in Right Ascension and Declination are less than about 35 and 15 arcseconds respectively for UFTI, or 9 and 4 arcseconds for IRCAM. Objects which would appear in both the target and sky regions, *i.e.* Declination extents south of the centre larger than 35 arcseconds (UFTI) or 8 arcseconds (IRCAM), should use recipe POL_EXTENDED for best results.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- Data errors are propagated through all processing steps. The initial values are found by applying the nominal ADU conversion and read noise.
- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- You should use SKY_FLAT_POL or SKY_FLAT_POL_ANGLE to make the flat fields.
- The target regions are 30% to 70% of the frame width about the Right-ascension centre, *i.e.* roughly centred on the source. The current sky limits are 1% to 99% of the frame width along the Right-ascension axis. The Declination pixel limits are instrument dependent, and are as follows. For UFTI, o sky: 69–264; e sky: 320–484; o target: 601–764; e target: 824–988. For IRCAM, o sky: 12–52; e sky: 67–107; o target: 152–192; e target: 207–247.

- The sky subtraction for a beam uses a constant modal sky level from the corresponding sky region.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects for automatic registration, the recipe matches the centroid of central source within an 8-arcsecond box. Should that fail for the jittered e- and o-beam sections, the recipe resorts to using the telescope offsets transformed to pixels. However, the final option for registering the e and o-beam mosaics at different waveplate angles, uses the beam offsets in arcseconds for the current filter converted to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the few pixels in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of twelve frames, the recipe creates mosaics for each beam and waveplate angle. Each mosaic has its bad pixels filled and after the first cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 6.3 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames and the mosaics (`_mos` or `_mos_c<cycle_number>` suffix).

Output Data :

- The integrated mosaics in `<m><date>_<group_number>_<beam><angle>_mos`, where `<m>` the instrument's group prefix. Token `<beam>` is e or o; and `<angle>` is 0, 22, 45, or 67.

- A mosaic for each cycle of jittered frames per beam and angle in
  `<m><date>_<group_number>_<beam><angle>_mos_c<cycle_number>`, where
  `<cycle_number>` counts from 0.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>`
  is the frame prefix. The naming format is slightly different for some non-UKIRT
  instruments.
- Polarisation frames `<m><date>_<group_number>_<suffix>`, each with a differ-
  ent suffix for the each parameter. The suffices are:

  I   intensity

  P   percentage polarisation

  PI   polarisation intensity

  Q   Stokes *Q*

  TH   polarisation angle

  U   Stokes *U*

- A FITS binary-table catalogue of the binned and culled polarisation data, called
  `<m><date>_<group_number>_I.FIT`. For each point it tabulates the *x-y* co-ordinates,
  the total intensity, the Stokes parameters, the percentage polarisation, the polarisation
  angle and intensity. There are additional columns giving the standard deviation on
  each of the tabulated values (excluding the co-ordinates). Likewise
  `<m><date>_<group_number>_all.FIT` and
  `<m><date>_<group_number>_bin.FIT` store the full and binned catalogues re-
  spectively.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter pattern, per waveplate angle. If this is not set,
the number of offsets, as given by FITS header NOFFSETS, minus one is used. If
neither is available, 3 is the default. An error state arises if the number of jittered
frames is fewer than 3. For observations prior to the availability of full ORAC, header
NOFFSETS will be absent. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

POL_ANGLE_JITTER, POL_EXTENDED, SKY_FLAT_POL,
SKY_FLAT_POL_ANGLE.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO,
  POLPACK, and CURSA.
- Uses the Starlink NDF format.
- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.

- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.

- The units are set for the frames with suffices (see "Output Data") P to `%`, and TH to `degrees`.

Deprecated Variants :

POL_JITTER3.

POL_NOD_CHOP
Reduces a chopped and nodded polarimetry observation nodded at each angle

Description:

This script reduces a chopped and nodded single-beam polarimetry observation, currently just for Michelle data. The imaging observation should comprise a multiple-of-four object frames nodded and chopped, and integrated at the four waveplate angles 0, 45, 22.5, 67.5 degrees in turn. For each waveplate angle the recipe makes automatically a calibrated, untrimmed mosaic. The recipe combines the multiple images of the source within each of these mosaics into new frames, and uses those four combined frames to calculate automatically calibrated polarisation images and vectors of the source. See "Output Data" for a list of these images.

It performs a null debiasing, creation and propagation of data variance, difference the integrations for each AB chop-beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

Notes:

- A variance array is created for each chop beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two chop beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of frames, the recipe creates mosaics for each chop beam and waveplate angle (modulo 180 degrees). Each mosaic has its bad pixels filled and after the first

cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- For each waveplate angle, the combined source image is made by taking symmetrical areas about each source, such that no pixels are duplicated. Thus the divisions occur at midpoints of the chop throw and the nod separations. These are registered using the source centroids.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 0.0 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north. A non-null value will be applied once it is determined.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_NOD_CHOP_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

- The integrated mosaics in <m><date>_<group_number>_p<angle>_mos, where <m> is the group prefix; and <angle> is 0, 22, 45, or 67.

- A mosaic for each cycle of chopped and nodded frames per waveplate angle in <m><date>_<group_number>_p<angle>_mos_c<cycle_number>, where <cycle_number> counts from 0.

- The combined source image and neighbourhoods at each waveplate angle in <m><date>_<group_number>_p<angle>_cab.

- The differenced pairs in <i><date>_<obs_number>_dp, where <i> is the frame prefix.

- Polarisation frames <m><date>_<group_number>_<suffix>, each with a different suffix for the each parameter. The suffices are:

I	intensity
P	percentage polarisation
PI	polarisation intensity
Q	Stokes *Q*
TH	polarisation angle
U	Stokes *U*

- A FITS binary-table catalogue of the binned and culled polarisation data, called `<m><date>_<group_number>_I.FIT`. For each point it tabulates the *x-y* co-ordinates, the total intensity, the Stokes parameters, the percentage polarisation, the polarisation angle and intensity. There are additional columns giving the standard deviation on each of the tabulated values (excluding the co-ordinates). Likewise `<m><date>_<group_number>_all.FIT` and `<m><date>_<group_number>_bin.FIT` store the full and binned catalogues respectively.

Parameters:

NUMBER = INTEGER
The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

POL_ANGLE_NOD_CHOP, POL_QU_FIRST_NOD_CHOP, NOD_CHOP_APHOT, POL_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, POLPACK, FIGARO, and CURSA.
- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.
- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.
- The units are set for the frames with suffices (see "Output Data") P to `%`, and TH to `degrees`.
- Error propagation is controlled by the USEVAR parameter.

POL_QU_FIRST_NOD_CHOP
Reduces a chopped and nodded polarimetry observation, where waveplate angle iterates in pairs at each jitter position

Description:

This script reduces a chopped and nodded single-beam polarimetry observation, currently just for Michelle data. The imaging observation should comprise chopped object frames at the angles 0, 45 degrees for each of a multiple-of-two pairs of nod positions, followed by frames at waveplate angles 22.5, 67.5 degrees also for each of a multiple-of-two pairs of nod positions.

For each waveplate angle the recipe makes automatically a calibrated, untrimmed mosaic. The recipe combines the multiple images of the source within each of these mosaics into new frames, and uses those four combined frames to calculate automatically calibrated polarisation images and vectors of the source. See "Output Data" for a list of these images.

It performs a null debiassing, creation and propagation of data variance, difference the integrations for each AB chop-beam pair, bad-pixel masking, difference adjacent nodded pairs, registers the frames, and forms a mosaic. See the "Notes" for further information.

Notes:

- A variance array is created for each chop beam, first using the read noise, and once the bias is removed, Poisson noise is added.

- A bias frame selected from the calibration system is removed from each beam in CHOP read mode. If no bias frame is available in the CHOP mode, the recipe subtracts a null bias, so the errors will be overestimated in the CHOP read mode; the data array will be unaffected once the beams are differenced. The ARRAY_TESTS recipe files a suitable short-exposure dark as a bias in the calibration system.

- The integrations of the two chop beams are differenced, the first subtracted from the second in each pair.

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies integer shifts of origin. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame, thus the noise will be greater in the peripheral areas having received less exposure time. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of frames, the recipe creates mosaics for each chop beam and waveplate angle (modulo 180 degrees). Each mosaic has its bad pixels filled and after the first cycle is then added into its own master mosaic of improving signal to noise. The exposure time is also summed and stored in each master mosaic's corresponding header. Likewise the end airmass and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- For each waveplate angle, the combined source image is made by taking symmetrical areas about each source, such that no pixels are duplicated. Thus the divisions occur at midpoints of the chop throw and the nod separations. These are registered using the nominal chop throws and telescope offsets.

- The polarised intensity is corrected for the statistical bias of the noise by subtracting the variance of Q or U.

- An offset of 0.0 degrees clockwise is applied to the rotation angle for the orientation of the analyser with respect to north. A non-null value will be applied once it is determined.

- The polarisation data for each pixel are also stored in catalogues. See "Output Data".

- The intensity image may be displayed with vectors overlaid. Steps are taken to reduce the number of noisy or insignificant pixels, as well as clutter. First, the polarisation catalogue data are averaged in 3-by-3-pixel bins. Second, a binned pixel is rejected if its polarisation is greater than 50% or is not positive, or its polarisation signal to noise less than 3, or its polarisation error is greater 5%. The bin size and thresholds can readily be changed by supplying arguments to the _CALC_STOKES_NOD_CHOP_ primitive.

- At the end of each cycle, the grand mosaics are registered, and new polarisation maps and catalogues constructed.

- Intermediate frames are deleted except for the differenced pairs (`_dp` suffix) frames.

Output Data :

- The integrated mosaics in `<m><date>_<group_number>_p<angle>_mos`, where `<m>` is the instrument's group prefix; and `<angle>` is 0, 22, 45, or 67.

- A mosaic for each cycle of chopped and nodded frames per waveplate angle in `<m><date>_<group_number>_p<angle>_mos_c<cycle_number>`, where `<cycle_number>` counts from 0.

- The combined source image and neighbourhoods at each waveplate angle in `<m><date>_<group_number>_p<angle>_cab`.

- The differenced pairs in `<i><date>_<obs_number>_dp`, where `<i>` is the frame prefix.

- Polarisation frames `<m><date>_<group_number>_<suffix>`, each with a different suffix for the each parameter. The suffices are:

> I   intensity
>
> P   percentage polarisation
>
> PI   polarisation intensity
>
> Q   Stokes *Q*
>
> TH   polarisation angle
>
> U   Stokes *U*

- A FITS binary-table catalogue of the binned and culled polarisation data, called `<m><date>_<group_number>_I.FIT`. For each point it tabulates the *x-y* co-ordinates, the total intensity, the Stokes parameters, the percentage polarisation, the polarisation angle and intensity. There are additional columns giving the standard deviation on each of the tabulated values (excluding the co-ordinates). Likewise `<m><date>_<group_number>_all.FIT` and `<m><date>_<group_number>_bin.FIT` store the full and binned catalogues respectively.

Parameters:

NUMBER = INTEGER

The number of frames in the nod pattern. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 4 is used. An error state arises if the number of jittered frames is fewer than 4 and not a multiple of 4. []

USEVAR = LOGICAL

Whether or not to create and propagate variance arrays. [1]

Related Recipes :

POL_NOD_CHOP, POL_ANGLE_NOD_CHOP, NOD_CHOP_APHOT.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, POLPACK, FIGARO, and CURSA.
- Uses the Starlink NDF format and multi-NDF HDS container files.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaics. The polarisation maps have new titles as follows using the suffices described in `Output Data`. I: `Intensity`; P: `Polarisation`; PI: `Polarised Intensity`; Q: `Stokes Q`; TH: `Polarisation Angle`; U: `Stokes U`.
- The origins of the generated polarisation maps are set to [1,1]. The WCS current frame is unchanged.
- The units are set for the frames with suffices (see "Output Data") P to `%`, and TH to `degrees`.
- Error propagation is controlled by the USEVAR parameter.

QUADRANT_JITTER
Reduces a "Quadrant Jitter" observation, including object masking

Description:

This script reduces a "quadrant jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising one or more series of four object frames where the target is approximately centred in each quadrant; and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs bad-pixel masking, null debiassing, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe is suitable for faint objects or objects within a comparatively bright core embedded in faint extended emission, *e.g.* a quasar; or extended objects less than 45 arcseconds across with UFTI, 10 arcseconds with IRCAM, and 2 arcminutes with IRIS2. The object need not be isolated, as the recipe masks objects within the other quadrants, and hence does not introduce significant artifacts into the flat field. For isolated objects use QUADRANT_JITTER_NO_MASK; or where speed is critical, use QUADRANT_JITTER_BASIC instead.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First the quadrant containing the object is masked in each object frame. Second an approximate flat field is created by combining the normalised and masked object frames using the clipped median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The second stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it matches the centroid of the central source. If this fails, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame. Thus the noise will be greater in the peripheral areas having received less exposure time. The full signal will be in the central ninth containing the main object. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of four, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of four in
  `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

   USEVAR = LOGICAL
   Whether or not to create and propagate variance arrays. [0]

Related Recipes :

   QUADRANT_JITTER_BASIC, QUADRANT_JITTER_NO_MASK, EXTENDED_3x3, EXTENDED_5x5.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

QUADRANT_JITTER_BASIC
Reduces a "Quadrant Jitter" observation, using just the basic operations for speed

---

Description:

This script reduces a "quadrant jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising one or more series of four object frames where the target is approximately centred in each quadrant; and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs bad-pixel masking, null debiasing, dark subtraction, flat-field creation and division, and registration using telescope offsets. See the "Notes" for further information.

This recipe aims to keep pace with the pipeline's incoming data and many options which improve the final mosaic are omitted. This recipe is suitable for faint objects or objects within a comparatively bright core embedded in faint extended emission, *e.g.* a quasar; or extended objects less than 45 arcseconds across with UFTI, 10 arcseconds with IRCAM, and 2 arcminutes with IRIS2. If the object is not isolated, there will be artifacts introduced into the flat field. These arise from the contribution of sources outside the quadrant containing the primary object. This variant of QUADRANT_JITTER is best for isolated objects or where speed is critical. Use QUADRANT_JITTER itself if object masking is required instead.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created in two steps. The quadrant containing the object is masked in each object frame. Then the recipe combines the normalised and quadrant-masked object frames using the median at each pixel.

- Registration is performed using the telescope offsets transformed to pixels.

- There is no resampling, merely integer shifts of origin.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to

the dimensions of a single frame. Thus the noise will be greater in the peripheral areas having received less exposure time. The full signal will be in the central ninth containing the main object. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of four, the recipe creates a mosaic, which is added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of four in
  `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

    USEVAR = LOGICAL
        Whether or not to create and propagate variance arrays. [0]

Related Recipes :

    QUADRANT_JITTER, QUADRANT_JITTER_NO_MASK, EXTENDED_3x3_BASIC, EXTENDED_5x5_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

QUADRANT_JITTER_NO_MASK
Reduces a "Quadrant Jitter" observation without object masking

---

Description:

This script reduces a "quadrant jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising one or more series of four object frames where the target is approximately centred in each quadrant; and a dark frame to make a calibrated, untrimmed mosaic automatically.

It performs bad-pixel masking, null debiassing, dark subtraction, flat-field creation and division, feature detection and matching between object frames, and resampling. See the "Notes" for further information.

This recipe is suitable for faint objects or objects within a comparatively bright core embedded in faint extended emission, *e.g.* a quasar; or extended objects less than 45 arcseconds across with UFTI, 10 arcseconds with IRCAM, and 2 arcminutes with IRIS2. If there are other objects of comparable brightness to the principal target in other quadrants, they will introduce artifacts into the flat field. Use QUADRANT_JITTER for those.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created in two steps. The quadrant containing the object is masked in each object frame. Then the recipe combines the normalised and quadrant-masked object frames using the median at each pixel.

- Registration is performed using common point sources in the overlap regions. If the recipe cannot identify sufficient common objects, it matches the centroid of the central source. If this fails, the script resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to

the dimensions of a single frame. Thus the noise will be greater in the peripheral areas having received less exposure time. The full signal will be in the central ninth containing the main object. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of four, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.

- A mosaic for each cycle of four in `<m><date>_<group_number>_mos<cycle_number>`, where `<cycle_number>` counts from 0.

- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent cycles.

Parameters:

   USEVAR = LOGICAL

      Whether or not to create and propagate variance arrays. [0]

Related Recipes :

   QUADRANT_JITTER, QUADRANT_JITTER_BASIC, EXTENDED_3x3_BASIC, EXTENDED_5x5_BASIC.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

QUADRANT_JITTER_TELE
Reduces a "Quadrant Jitter" observation, using object masking, and telescope offsets for registration

Description:

   This script reduces a "quadrant jitter" photometry observation with near-infrared imaging data. It takes an imaging observation comprising one or more series of four object frames where the target is approximately centred in each quadrant; and a dark frame to make a calibrated, untrimmed mosaic automatically.

   It performs bad-pixel masking, null debiasing, dark subtraction, flat-field creation and division, registration using telescope offsets, and resampling. See the "Notes" for further information.

   This recipe is intended for extended moving sources (comets) tracked by the telescope. The source extent should not exceed 45 arcseconds for UFTI or 10 arcseconds for IRCAM, in moderately crowded fields. Sources may include those with a comparatively bright core embedded in faint extended emission. The object need not be isolated, as the recipe masks objects within the other quadrants, and hence does not introduce significant artifacts into the flat field. This recipe should not be used for frames where the telescope has not guided on the moving object. In that case reduction should be performed by MOVING_QUADRANT_JITTER, which registers using the telescope offsets alone.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First the quadrant containing the object is masked in each object frame. Second an approximate flat field is created by combining the normalised and masked object frames using the clipped median at each pixel. This flat field is applied to the object frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The second stage is repeated but applied to the masked frames to create the final flat field.

- Registration is performed using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaics by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is not trimmed to the dimensions of a single frame. Thus the noise will be greater in the peripheral areas having received less exposure time. The full signal will be in the central ninth containing the main object. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of four, the recipe creates a mosaic, which has its bad pixels filled and is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

Output Data :

- The integrated mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- A mosaic for each cycle of four in
<m><date>_<group_number>_mos<cycle_number>, where <cycle_number> counts from 0.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

- The created flat fields in flat_<filter>_<group_number> for the first or only cycle, and flat_<filter>_<group_number>_c<cycle_number> for subsequent cycles.

Parameters:

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

JITTER_SELF_FLAT_TELE, MOVING_JITTER_SELF_FLAT,
MOVING_QUADRANT_JITTER, QUADRANT_JITTER.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, FIGARO, and EXTRACTOR.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is not used.

REDUCE_DARK
Averages and files observations as the current dark

Description:

This recipe reduces dark-frame observations with infrared imaging data. Multiple darks of the same exposure time are averaged. It files the single or averaged dark in the dark index file. Other reduction steps comprise bad-pixel masking, optional creation of data errors.

This recipe reduces a dark-frame observation with infrared imaging data. It files the dark in the dark index file. Reduction comprises bad-pixel masking, and optional creation of data errors.

Notes:

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- Intermediate frames are deleted.
- Sub-arrays are supported.

Output Data :

- The dark called `dark_<exposure_time>_<frame_number>`. For single frames the group number is the same as the frame number. The decimal point in the time is replaced by `p`.
- The dark is filed in `$ORAC_DATA_OUT/index.dark`.

Parameters:

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Implementation Status:

- The processing engines are from the Starlink package KAPPA.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the dark.

REDUCE_FLAT
Reduces an imaging flat field

Description:

This reduces a flat field in the conventional manner consisting of bad-pixel masking, dark subtraction, and normalisation by the mean of the image. It also files the normalised flat-field frame for use by subsequent flat-fielding operations.

Notes:

- This recipe will reduce any image passed to it. Care must be taken to ensure that a proper flat-field image will be reduced.
- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- The flat field is normalised using a clipped mean, where the clipping levels are 2, 3, and 3 standard deviationss. This value is effectively the mode of the image. The entire image is divided by this value to normalise it.
- Intermediate frames are deleted.

Output Data :

- The flat is called `flat_<frame_number>`.
- The flat is filed in `$ORAC_DATA_OUT/index.flat`.

Implementation Status:

- The processing engines are from the Starlink packages KAPPA and CCDPACK.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Variance information is not supported.

SKY_AND_JITTER
Reduces a "combined jitter" photometry observation

Description:

This script reduces a "combined jitter" photometry observation with UKIRT imaging data. It takes an imaging observation comprising one or more sets of frames, each set containing a sky frame, followed by jittered object frames; and a pre-determined flat-field frame to make a calibrated, trimmed mosaic automatically.

This recipe performs bad-pixel masking, null debiasing, sky subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

This recipe is suitable for moderately faint point sources.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.
- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.
- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.
- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.
- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.
- The most-recent sky frame is used for the sky subtraction.
- Where automatic registration is not possible, the recipe matches the centroid of central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.
- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.
- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).
- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.
- Sub-arrays are supported.

Output Data :

- The resultant mosaic in `<m><date>_<group_number>_mos`, where `<m>` is the instrument's group prefix.
- The individual flat-fielded frames in `<i><date>_<obs_number>_ff`, where `<i>` is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter, excluding the sky frame. If absent, the number of offsets, as given by header NOFFSETS, minus two is used. If neither is available, 5 is used. An error state arises if the number of jittered frames is fewer than 3. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

JITTER_SELF_FLAT, NOD_SELF_FLAT_NO_MASK, SKY_AND_JITTER_APHOT,
SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through intermediate files to the mosaic.
- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

SKY_AND_JITTER5.

SKY_AND_JITTER_APHOT
Reduces a "combined jitter" photometry observation, and performs aperture photometry

Description:

> This script reduces a "combined jitter" photometry observation with UKIRT imaging data. It takes an imaging observation comprising one or more sets of frames, each set containing a sky frame, followed by jittered object frames; and a pre-determined flat-field frame to make a calibrated, trimmed mosaic automatically.

> This recipe performs bad-pixel masking, null debiasing, sky subtraction, flat-field division, feature detection and matching between object frames, and resampling. See the "Notes" for details.

> Photometry of the point source using a fixed 5-arcsecond aperture is calculated for each jitter frame and the mosaic. The results appear in `$ORAC_DATA_OUT/aphot_results.txt` in the form of a Starlink small text list. The analysis of each star is appended to this file.

> This recipe is suitable for moderately faint point sources.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- You may use SKY_FLAT or SKY_FLAT_MASKED to make the flat field.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The most-recent sky frame is used for the sky subtraction.

- Where automatic registration is not possible, the recipe matches the centroid of central source, and should that fail, it resorts to using the telescope offsets transformed to pixels.

- The resampling applies non-integer shifts of origin using bilinear interpolation. There is no rotation to align the Cartesian axes with the cardinal directions.

- The recipe makes the mosaic by applying offsets in intensity to give the most consistent result amongst the overlapping regions. The mosaic is trimmed to the dimensions of an input frame. The mosaic is not normalised by its exposure time (that being the exposure time of a single frame).

- For each cycle of jittered frames, the recipe creates a mosaic, which is then added into a master mosaic of improving signal to noise. The exposure time is also summed and stored in the mosaic's corresponding header. Likewise the end airmass header and end UT headers are updated to match that of the last-observed frame contributing to the mosaic.

- The photometry tabulation includes the file name, source name, time, filter, airmass, the catalogue magnitude and estimates of the zero-point with and without the application of a mean extinction. There are headings at the top of each column.

- The photometry uses the mode calculated from $3 * \text{median} - 2 * \text{mean}$ and Chauvenet's rejection criterion to estimate the sky level in an annulus about the source. The inner annulus diameter is 1.3 times that of the aperture (6.5 arcsec); the outer annulus is 2.5 times (12.5 arcsec) for UFTI and twice the aperture (10 arcsec) for IRCAM. The errors are internal, based on the sky noise.

- Intermediate frames are deleted except for the flat-fielded (`_ff` suffix) frames.

- Sub-arrays are supported.

Output Data :

- The resultant mosaic in <m><date>_<group_number>_mos, where <m> is the instrument's group prefix.

- The individual flat-fielded frames in <i><date>_<obs_number>_ff, where <i> is the frame prefix. The naming format is slightly different for some non-UKIRT instruments.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter, excluding the sky frame. If absent, the number of offsets, as given by header NOFFSETS, minus two is used. If neither is available, 5 is used. An error state arises if the number of jittered frames is fewer than 3. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

JITTER_SELF_FLAT_APHOT, NOD_SELF_FLAT_NO_MASK_APHOT,
SKY_AND_JITTER, SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages: CCDPACK, KAPPA, AND FIGARO.

- Uses the Starlink NDF format.

- History is recorded within the data files.

- The title of the data is propagated through intermediate files to the mosaic.

- Error propagation is controlled by the USEVAR parameter.

Deprecated Variants :

   SKY_AND_JITTER5_APHOT.

SKY_FLAT
Creates and files a flat field derived from jittered frames

Description:

This recipe makes a sky flat for UKIRT imaging from a series of sky or object frames which are combined using one of a selection of statistics. It expects one dark frame followed by jittered sky frames.

It performs a null debiassing, bad-pixel masking, and dark subtraction before combining normalised frames pixel by pixel using the median. Details of the flat are filed in the index of flats for future selection and use of the flat. See the "Notes" for further details.

For best results the field observed should contain few stars and no bright ones. In contaminated sky regions, recipe SKY_FLAT_MASKED will greatly reduce artifacts appearing in the resultant flat.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- Intermediate frames are deleted.

- Sub-arrays are supported.

Output Data :

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent recipe cycles. Token $<$`filter`$>$ is the filter name, $<$`group_number`$>$ is the frame number of the group, and $<$`cycle_number`$>$ is the number of the cycle, counting from one.

- The flats are filed in `$ORAC_DATA_OUT/index.flat`.

Parameters:

NUMBER = INTEGER
> The number of frames in the jitter. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 5 is used. An error state arises if the number of jittered frames is fewer than 3. []

USEVAR = LOGICAL
> Whether or not to create and propagate variance arrays. [0]

Related Recipes :

SKY_FLAT_FP, SKY_FLAT_MASKED, SKY_FLAT_POL.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK, KAPPA, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

SKY_FLAT_FP
Creates and files a flat field derived from multiples of four frames using object masking to reduce artifacts

Description:

This recipe make a sky flat for UFTI from a series of four (or multiples of four) sky or object frames combined using one of a selection of statistics. It is intended to be used to make a flat for Fabry-Perot data.

It performs a null debiassing, bad-pixel and non-signal region masking, and dark subtraction before combining the sky frames pixel by pixel to make the flat. See the "Notes" for further details. The parameters of the flat are filed in the index of flats for future selection and use of the flat.

For best results the field observed should contain few stars and no bright ones.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- Applies a mask about the Fabry-Perot's transmitted circular region on the detector. If the centre is not known through the fpcentre calibration, it is determined using profiles of the surrounding ring.

- The flat field is created iteratively. First an approximate flat field is created by combining normalised sky frames using the median at each pixel. This flat field is applied to the sky frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Intermediate frames are deleted.

- Sub-arrays are supported.

Output Data :

- The created flat field in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent recipe cycles. Token $<$filter$>$ is the filter name, $<$group_number$>$ is the frame number of the group, and $<$cycle_number$>$ is the number of the cycle, counting from one.

- The flats are filed in `$ORAC_DATA_OUT/index.flat`.

Parameters:

> **USEVAR = LOGICAL**
>> Whether or not to create and propagate variance arrays. [0]

Related Recipes :

> SKY_FLAT, SKY_FLAT_MASKED.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

SKY_FLAT_MASKED
Creates and files a flat field derived from jittered frames using object masking to reduce artifacts

Description:

This recipe makes a sky flat for UKIRT imaging from a series of sky or object frames which are combined using one of a selection of statistics. It expects one dark frame followed by jittered sky frames.

It performs a null debiasing, bad-pixel masking, and dark subtraction before combining the sky frames pixel by pixel to make the flat. See the "Notes" for further details. The parameters of the flat are filed in the index of flats for future selection and use of the flat.

For best results the field observed should contain few stars and no bright ones. In sparse fields, recipe SKY_FLAT is a faster alternative.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised sky frames using the median at each pixel. This flat field is applied to the sky frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Intermediate frames are deleted.

- Sub-arrays are supported.

Output Data :

- The created flat fields in `flat_<filter>_<group_number>` for the first or only cycle, and `flat_<filter>_<group_number>_c<cycle_number>` for subsequent recipe cycles. Token `<filter>` is the filter name, `<group_number>` is the frame number of the group, and `<cycle_number>` is the number of the cycle, counting from one.

- The flats are filed in `$ORAC_DATA_OUT/index.flat`.

Parameters:

NUMBER = INTEGER
The number of frames in the jitter. If absent, the number of offsets, as given by header NOFFSETS, minus one is used. If neither is available, 5 is used. An error state arises if the number of jittered frames is fewer than 3. []

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [0]

Related Recipes :

SKY_FLAT.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

SKY_FLAT_POL
Creates and files a flat field derived from eight frames using object masking to reduce artifacts

Description:

This recipe make a sky flat for UKIRT imaging from a series of eight sky or object frames combined using one of a selection of statistics. It is intended to be used to make a flat for polarimetry data. The data should comprise two spatial positions at the waveplate angles 0, 45, 22.5, and 67.5 degrees.

It performs a null debiassing, bad-pixel masking, and dark subtraction before combining the sky frames pixel by pixel to to make the flat. See the "Notes" for further details. The parameters of the flat are filed in the index of flats for future selection and use of the flat.

For best results the field observed should contain few stars and no bright ones.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat-field is created by combining normalised sky frames using the median at each pixel. This flat field is applied to the sky frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Intermediate frames are deleted.

- Sub-arrays are supported.

Output Data :

- There are flats for each waveplate angle and also for 90 and 135 degrees, made by copying the original flat frame. This is to permit both flats made at each angle (SKY_FLAT_POL_ANGLE), or with the angles combined as here. Each flat is called `flat_<filter>_pol<waveplate_angle>_<group_number>`. The

<waveplate_angle> is the integer part of the angle, *e.g.* 22, 67; where <filter> is the filter name (excluding the +pol), and <group_number> is the frame number of the group. For each subsequent cycle of the recipe, the recipe makes new flats which have a _c<cycle_number> suffix, where <cycle_number> is the number of the cycle, counting from one.

- The flats are filed in $ORAC_DATA_OUT/index.flat.

Parameters:

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

SKY_FLAT_FP, SKY_FLAT_MASKED, SKY_FLAT_POL_ANGLE.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK, KAPPA, FIGARO, and EXTRACTOR.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

SKY_FLAT_POL_ANGLE
Creates and files flat fields derived from jittered frames at each waveplate angle, using object masking to reduce artifacts

---

Description:

This recipe make a sky flat for UKIRT imaging from a series of sky or object frames combined using one of a selection of statistics. It is intended to be used to make flats at each waveplate angle for polarimetry data. The data should comprise at least three spatial positions for each waveplate angle 0, 45, 22.5, and 67.5 degrees in turn.

It performs a null debiassing, bad-pixel masking, and dark subtraction before combining the sky frames pixel by pixel to to make the flat. See the "Notes" for further details. The parameters of the flat are filed in the index of flats for future selection and use of the flat.

For best results the field observed should contain few stars and no bright ones.

Notes:

- A World Co-ordinate System (WCS) using the AIPS convention is created in the headers should no WCS already exist.

- For IRCAM, old headers are reordered and structured with headings before groups of related keywords. The comments have units added or appear in a standard format. Four deprecated headers are removed. FITS-violating headers are corrected. Spurious instrument names are changed to IRCAM3.

- The bad-pixel mask applied is `$ORAC_DATA_CAL/bpm`.

- Each dark-subtracted frame has thresholds applied beyond which pixels are flagged as bad. The lower limit is 5 standard deviations below the mode, but constrained to the range -100 to 1. The upper limit is 1000 above the saturation limit for the detector in the mode used.

- The flat field is created iteratively. First an approximate flat field is created by combining normalised sky frames using the median at each pixel. This flat field is applied to the sky frames. Sources within the flat-fielded frames are detected, and masked in the dark-subtracted frames. The first stage is repeated but applied to the masked frames to create the final flat field.

- Intermediate frames are deleted.

- Sub-arrays are supported.

Output Data :

- The flats are called `flat_<filter>_pol<waveplate_angle>_<group_number>`, The $<$waveplate_angle$>$ is the integer part of the angle, *e.g.* 22, 67; $<$filter$>$ is the filter name (excluding any +pol); and $<$group_number$>$ is the frame number of the group. For each subsequent cycle of the recipe, the recipe makes new flats which have

a `_c<cycle_number>` suffix, where `<cycle_number>` is the number of the cycle, counting from one.

- The flats are filed in `$ORAC_DATA_OUT/index.flat`.

Parameters:

USEVAR = LOGICAL
Whether or not to create and propagate variance arrays. [1]

Related Recipes :

SKY_FLAT_FP, SKY_FLAT_MASKED, SKY_FLAT_POL.

Implementation Status:

- The processing engines are from the Starlink packages CCDPACK, KAPPA, FIGARO, and FIGARO.
- Uses the Starlink NDF format.
- History is recorded within the data files.
- The title of the data is propagated through the intermediate file to the flat.
- Error propagation is controlled by the USEVAR parameter.

E   Instrument Recipe Notes

This appendix summarises the modifications of instrument-specific variants of the recipes, and the restricted availability of certain recipes by instrument.

E.1   Classic Cam

There is no recipe name in the headers, therefore recipes must be supplied on the command line or the data's FITS headers edited.   See Section 5 for instructions.

Actual testing has been performed on the core recipes JITTER_SELF_FLAT(_APHOT), BRIGHT_POINT_SOURCE(_APHOT), REDUCE_DARK. Other near infra-red plain-imaging recipes should work provided the observation pattern matches that expected by the recipe.

Aperture photometry accesses the Persson HST lists.

E.2   INGRID

By default the infrastructure uses information in the headers to assign recipe names to RE-DUCE_DARK for a dark, and JITTER_SELF_FLAT for jittered target frames. Otherwise it defaults to QUICK_LOOK. Other recipes supplied on the **oracdr** command line should work provided the observation pattern matches that expected by the recipe.

E.3   IRCAM

ARRAY_TESTS uses eight frames and derives both the STARE and ND_STARE readout noises, which are both filed with the calibration system. The range of acceptable dark current is 0–10 electrons/sec compared with -1–$+1$ for UFTI, and the nominal ND_STARE readout noise is 20–50 electrons compared with 8–30 electrons for UFTI. The STARE mode readout noise should be 30–70 electrons to be nominal. The log file goes to `$ORAC_DATA_OUT/ircam3_array_tests.log`.

E.4   IRIS2

The four EXTENDED_$n \times m$ recipes cope with missing offset information and a lower complete-ness (0.25 from 0.4) for automatic registration, thus a smaller fraction of detected objects need match.

ARRAY_TESTS logs to a different location `/inst2_soft/iris2red/logs/array_tests.log` at the AAT compared with UFTI at UKIRT. References to NDSTARE are not applicable, and so are absent from the documentation. The observation mode is DRM. The upper limit on the readout noise is 20 electrons compared with 30 electrons for UFTI.

There are JITTER_SELF_FLAT variants with a higher registration completeness (0.5 from 0.4) giving purer registration but may rely on the telescope offsets more often.

For a full list of available recipes see the IRIS2 web page.

E.5   ISAAC

The observation template and sequence names are converted into matching recipes with a default of QUICK_LOOK. Available are: CHOP_SKY_JITTER, JITTER_SELF_FLAT, JITTER_SELF_FLAT_APHOT, NOD_SELF_FLAT_NO_MASK, NOD_SELF_FLAT_NO_MASK_-APHOT, POL_JITTER, REDUCE_DARK, and SKY_FLAT_MASKED. Please note that at the time of writing the thermal recipes have yet to be tested with ISAAC data. Polarimetry is not available yet. Other recipes supplied on the **oracdr** command line should work provided the observation pattern matches that expected by the recipe.

The CHOP_SKY_JITTER recipe has no leading sky frame, so the recipe subtracts the mode of the first sky frame from the first target frame, rather than interpolating between the bracketting sky frames as occurs in the standard version. The sequence of frames can also end on the source.

The spatial distortion correction in the core jitter recipes uses the ATOOLS package in addition to those engines listed in the generic documentation.

Aperture photometry accesses both the UKIRT Faint-standard and Persson HST lists.

E.6   Michelle

DIFFERENCE_STATS, POL_ANGLE_NOD_CHOP, POL_QU_FIRST_NOD_CHOP, and POL_NOD_CHOP only apply to Michelle.

ARRAY_TESTS calculates and files both an average bias frame and the read noise in the calibration system. There is no logging. Checks are made that appropriate waveforms are selected.

E.7   NACO

The scope of the recipes is similar to that for ISAAC. No attempt has been made to reduce any coronographic data with the standard recipes.

Registration is critical for such small point-spread functions to match well. Therefore the minimum number of contiguous pixels above the threshold to be counted as a fiducial source for automatic registration is increased from 9 to 15, and the percentile detection threshold raised from 98% to 99%. These may need further adjustment as a wider selection of observations are processed. The sky co-ordinates are used to aid registration by default.

Aperture photometry accesses both the UKIRT Faint-standard and Persson HST lists.

E.8   NIRI

Tests with 2004 data are limited to thermal data and REDUCE_DARK recipes.

The prelminaries fudge the WCS headers to give a consistent set

A 3-arcsecond aperture is used by default for the thermal imaging aperture photometry.

E.9   UFTI

The FP family of recipes only applies to UFTI.

The sky counts are validated as part of the initial processing.

E.10   UIST

There is no ARRAY_TESTS recipe. Instead there are DARK_AND_BPM and
MEASURE_READNOISE.

There are JITTER_SELF_FLAT variants with a higher registration completeness (0.5 from 0.4)
giving purer registration but may rely on the telescope offsets more often.

Measurement of the full-array statistics avoids the outer 5% border, in order to avoid aberrant
pixels which can still bias the clipped mean.

Raw polarimetric data are flipped along the second axis to counter a bug in POLPACK computing
vector orientations for left-handed co-ordinates.

F   Internal Headers

F.1   Translated Headers

The ORAC-DR infrastructure translates instrument metadata stored in FITS headers into internal headers. An internal header offers a common name, meaning, and units independent of the instrument, thereby makes most recipe code portable between instruments, and increases code reuse, a major theme of ORAC-DR. The internal headers are accessed through the `ORAC::Frame::uhdr()` and `ORAC::Group::uhdr()` methods (see Frame headers in SUN/233).

Below is a list of these translated internal headers and their meanings and units. Here are some notes pertaining to the table.

- In a primitive these header names have an `ORAC_` prefix, for example, ORAC_DEC_SCALE.

- The abbreviations for data type are B: boolean, F: floating point, I: integer, and S: string. Of the floating point only the RA_BASE and DEC_BASE warrant double precision, but the recipes impose no restriction.

- Units are in brackets.

- Some headers retain instrument-specific values, like ORAC_SPEED until a common set of instrument-independent names is compiled.

Name	Type	Meaning
AIRMASS_START	F	Airmass at the start of the observation
AIRMASS_END	F	Airmass at the end of the observation
DEC_BASE	F	Declination (J2000) at reference position (and offset 0,0) [deg]
DEC_SCALE	F	Pixel scale along declination axis [arcsec]
DEC_TELESCOPE_OFFSET	F	Telescope declination offset with respect to the DEC_BASE position [arcsec]
DETECTOR_BIAS	F	Detector bias (only used by IRCAM) [V]
DETECTOR_MODE	S	Such as STARE, NDSTARE, CHOP
EQUINOX	F	Equinox of co-ordinates (fix at 2000.0) [y]
EXPOSURE_TIME	F	Exposure time for each co-add [s]
FILTER	S	Filter name
GAIN	F	Conversion factor [electrons/data number]
INSTRUMENT	S	Name of instrument such as IRCAM, UFTI, UIST Michelle, ISAAC but its use is deprecated
NUMBER_OF_EXPOSURES	I	Number of exposures in the integration
NUMBER_OF_OFFSETS	I	Number of jitter offset positions
NUMBER_OF_READS	I	Number of reads per exposure
OBJECT	S	Name of the object, preferably adhering to IAU standard
OBSERVATION_MODE	S	Operating mode of multi-mode instrument, `imaging`, `spectroscopy`, or `ifu`
OBSERVATION_NUMBER	I	Observation number (starting at 1 each night)
OBSERVATION_TYPE	S	`BIAS`, `DARK`, `FLAT`, `LAMP`, `OBJECT`, `SKY`
RA_BASE	F	Right Ascension (J2000) at reference position (and offset 0,0) [h]

RA_SCALE	F	Pixel scale along right-ascension axis [arcsec]
RA_TELESCOPE_OFFSET	F	Telescope right-ascension offset with respect to the RA_BASE position [arcsec]
READMODE	S	Such as STARE, NDSTARE, CHOP
ROTATION	F	Angle of the Declination axis with respect to the frame's y axis measured counter clockwise.
SPEED_GAIN	S	Speed and type of readout, e.g. Normal, Fast, Higain for UFTI; Standard, Fast, Deepwell for IRCAM.
TELESCOPE	S	The PAL palObs telescope name
STANDARD	B	Whether or not the observation is of a standard
UTDATE	S	UT date of the observation in *yyyymmdd* format
UTEND	F	UT of the end of the observation [h]
UTSTART	F	UT of the start of the observation [h]
WAVEPLATE_ANGLE	F	Polarimetry waveplate position angle [deg]
X_LOWER_BOUND	I	Start column of array readout
X_REFERENCE_PIXEL	F	The pixel index of the x reference position for RA_BASE and DEC_BASE
X_UPPER_BOUND	I	End column of array readout
Y_LOWER_BOUND	I	Start row of array readout
Y_REFERENCE_PIXEL	F	The pixel index of the y reference position for RA_BASE and DEC_BASE
Y_UPPER_BOUND	I	End row of array readout

F.2   Steering Headers

Headers which direct a recipe are called steering headers. These can include timing information, such as when a flat should be made; qualifiers about the frame such as whether it is of a target or sky. They use header information and membership index within a group. The names of steering headers should be longer than eight characters to avoid confusion with FITS keywords.

Most steering headers are per frame and are accessed by `ORAC::Frame::uhdr()` (see Frame headers in SUN/233). Steering headers which must persist between frames, such as those which define groups of related raw or processed frames like differenced chopped pairs, must be stored in the group internal headers, accessed by `ORAC::Group::uhdr()`.

While programmers can create their own steering headers as needed, it is silly to invent new names for ones which already exist and are accessed by many primitives, which the programmers would then not be able to use directly.

Here is a list of the common steering headers used in the imaging recipes. Not all headers

are pertinent to all recipes. The headers used exclusively by specialist recipes, such as for Fabry-Perot or nod-chop in a scan mode, are not listed below. The abbreviations for data type are B: boolean (1 for true, 0 for false), I: integer, F: floating point, and S: string. Note that some of the timing headers may in fact switch on for the same frame.

Name	Type	Meaning
CREATE_CATALOGUE	B	Whether or not to create an object catalogue.
CYCLE_NUMBER	I	Number of the cycle, a cycle being a set of frames to complete a pass through the recipe. The first cycle is 0.
DEC_OFFSET_DIFFERENCE	F	The declination displacement between a nodded pair of frames.
DIFFERENCE_PAIR	B	Whether or not to subtract pairs. It is normally true every second frame.
DO_APHOT	B	Whether or not perform aperture photometry. Photometry is performed once the mosaic is made.
EXTENDED_ROW	I	The row number of the frame for EXTENDED_$n \times m$ recipes.
FLAT_DIVIDE	B	Whether or not to apply a flat field.
JITTER_FIRST	B	This selects the ordering of polarimetry frames. If true the jittering occurs at all positions before the waveplate is turned. If false, all waveplate angles are observed at a given offset.
JITTER_NUMBER	I	The number of frames in the jitter.
MAKE_FLAT	B	Whether or not to make a flat field.
MAKE_GRAND_MOSAIC	B	Whether or not register the frames and make the full mosaic for EXTENDED_$n \times m$ recipes.
MAKE_MOSAIC	B	Whether or not register the frames and make the mosaic. For EXTENDED_$n \times m$ recipes it is the time to make a row mosaic.
MASK_OBJECTS	B	Whether or not to mask the objects. Masking occurs when all the jittered frames in a cycle are available.
PAIR_ORDER	B	Pair subtraction order, true means take second from the first, and false means take the first from the second.
POL_CYCLE_NUMBER	B	Number of the polarimetry cycle, a cycle being a set of frames to complete a pass through the recipe for all waveplate angles. The first cycle is 0.
RA_OFFSET_DIFFERENCE	F	The right-ascension displacement between a nodded pair of frames.
REFERENCE_FRAME	B	A true value specifies the reference frame for normalisation of the masked frames. It is true for the first frame and false for all subsequent frames in the observation.
REFERENCE_LEVEL	F	The reference modal level, used when combining masked frames to form a flat.
REFERENCE_SKY	F	The reference sky level, used for sky subtraction.
REGISTER_IMAGES	B	Whether or not to register and resample the polarimetric e- and o-beam mosaics.

Name	Type	Meaning
TARGET_NUMBER	I	When TARGET_OR_SKY is `"target"`, this counts the target frames, starting from zero. It is used for interpolation between sky measurements.
USE_VARIANCE	B	Whether or not variance processing is to occur.
WAVEPLATE_FLAT	B	Whether or not to make a flat for each polarimeter waveplate angle. For non-polarimetric data, the value is immaterial. For polarimetric data, false means combine all waveplate angles to make the flat (there should be an equal number of each angle).

G   Release Notes—V4.1

The main changes are improvements to polarimetry, aperture photometry, and Fabry-Perot reductions; several new recipes; and support for Magellan Classic Cam, VLT NACO, and Gemini NIRI instruments.

G.1   New recipes

BRIGHT_POINT_SOURCE_NCOLOUR Reduces a multi-colour bright-point-source photometry observation.

BRIGHT_POINT_SOURCE_NCOLOUR_APHOT Reduces a multi-colour bright-point-source photometry observation and performs aperture photometry.

BRIGHT_POINT_SOURCE_CATALOGUE Reduces a bright-point-source photometry observation, producing a catalogue of all sources in the field.

BRIGHT_POINT_SOURCE_TELE Reduces a bright-point-source photometry observation, using telescope offsets for registration.

BRIGHT_POINT_SOURCE_TELE_APHOT Reduces a bright-point-source photometry observation, using telescope offsets for registration, and performs aperture photometry.

DIFFERENCE_STATS Calculates statistics for Michelle darks in a pairwise manner.

JITTER_SELF_FLAT_CATALOGUE Reduces a "standard jitter" photometry observation using object masking, and produces a catalogue of all sources in the field.

JITTER_SELF_FLAT_NCOLOUR_APHOT Reduces a multi-colour "standard jitter" photometry observation using object masking, and performs aperture photometry.

LAMP_FLAT Creates and files imaging flat fields derived from a calibration lamp for ISAAC and NACO.

POL_QU_FIRST_NOD_CHOP Reduces a Michelle chopped and nodded polarimetry observation, where waveplate angle iterates in pairs at each jitter position. Its purpose is to form the *Q,U* Stokes parameters as quickly as possible, hence offer more-accurate polarimetry than integrating at four waveplate angles before nodding.

G.2   Modified recipes

- The UIST ARRAY_TESTS calculates the dark current and readnoise in four regions equally divided along the *X*-axis. The recipe reports the quality of each statistic compared with normal operating parameters.

- FP, FP_JITTER, and FP_JITTER_NO_SKY May allow scaling of the various components in the formula to create the reduced data, if other external calibrations are available, defaulting to the previous behaviour.

- MAKE_BPM No longer applies an existing bad pixel mask and creates unwanrted variance.

- REDUCE_DARK Permits averaging of darks of the same exposure time. The dark frame name now includes the exposure time. This does not apply to instruments whose darks only have a group identifier of 0, namely IRCAM, UFTI, and UIST.

- SKY_FLAT and SKY_FLAT_MASKED These can now process multi-coloured observations.

- SKY_FLAT_FP Applies a mask beyond the etalon's transmitted circular region on the detector. The circle's centre is either determined through the new fpcentre calibration, or it is determined using profiles of the surrounding ring.

G.3   Global changes

The main changes from a user perspective were as follows.

- Updated and expanded documentation, featuring the new recipes and instrument-data supported, more on the "Features of the Primitives", and a new appendix listing the file prefixes.

- Added support for Magellan Classic Cam imager, and the ESO NACO instrument in imaging mode. These have only processed a few nights data exercising a few recipes. The NACO pipeline would certainly benefit from more data.

- Upgraded support for the Gemini NIRI instrument from pre-alpha to alpha, through more-thorough header translations, adjusting the nominal WCS to an actual one, setting the saturation levels, tested aperture photometry within a 3-arcsec aperture of thermal data. It's still a work in progress, but it should be possible to process non-thermal data too given a sequence of frames matching a standard sequence.

- Incorporated eStar (intelligent-agent) calls in in five recipes, including the two new recipes which make an inventory of the sources such as BRIGHT_POINT_SOURCE_CATALOGUE.

- The IRIS2 ARRAY_TESTS has been improved. The mode, speed, and readnoise are written to the calibration log and reported.

- Several recipes in the BRIGHT_POINT_SOURCE JITTER_SELF_FLAT families and CHOP_SKY_JITTER have two additional steps: one to correct for residual bias variations along columns, and the other to correct for field distortion. For most instruments these are null, but not for ISAAC, where both are applied; and NACO, where the bias striping is removed.

- The ISAAC sky co-ordinate system is tied to the undistorted grid co-ordinates rather than than the distorted grid. This means that measured sky positions for sources will not be misplaced by the distortion. IRIS2 now corrects its world co-ordinate system for spatial distortion, but the pixel grid is not resampled.

- Improved central-source registration and corrected the world co-ordinate system associated with polarimetry products. The latter includes allowance for the bifurcation in the Wollaston prism, and the left-handed sky co-ordinates of UIST (by flipping the raw data and WCS headers). Enlarged the extracted target and sky regions for UIST.

- Combining sources in NOD_CHOP recipes revamped, using the chop and nod offsets to decide upon the visibility of an image. Thus it can now cope with just one positive-negative image pair.

- Centroid hunting allows for the source to be just outside the initial search box, and should lead to better registration of single sources and photometry, when the source is not placed at the nominal location (such as shifted instrument apertures at UKIRT).

- The option to clean small blemishes using a 15×15 box filter during flat creation is now accessible from recipes, say when invoking _FLAT_FIELD_MASKED_GROUP_.

- Object masking reports the detection threshold read from the EXTRACTOR parameter file rather than a hardwired value.

- For ESO data CHOP_SKY_JITTER allows the last frame to be of the target. Fixed a bug where the normalised sky frames were flat-fielded prior to object masking instead of flattening the frames with masked deviant pixels.

- Fixed a bug in determining a consistent set of sky levels for CHOP_SKY_JITTER in multiple recipe iterations.

- Fixed a bug where exposure time was not integrated in the grand mosaic of EXTENDED recipes. The the start and end times and airmasses are also updated in the grand mosaic's headers. The mosaic's WCS is set to the SKY by default.

- Allow polarimetry recipes to work with angles of 0, 45, 90, and 135 degrees. This permits the circular polarisation to be determined from the Stokes images. Appropriately named flats for 90 and 135 degrees are created where necessary.

- There is a new polrefang calibration to correct the measured polarisation angles to position angle, allowing for instrumental alignment. The index file may be created manually to override the default calibrations.

- The default near-infrared polarisation selection criteria have changed. The maximum polarisation threshold is increased from 50% to 75%. The standard deviation is relaxed to 10%. There is a new criterion where the minimum intensity must be at least three standard deviations above zero. The binning size is increased from 3 to 5 and its value is reported.

- The polarimetry-parameter images now have their world co-ordinates set to SKY, so plots will be annotated with equatorial co-ordinates.

- More intermediate frames are tidied at the end of recipes, notably in CHOP_SKY_JITTER.

- Improved header translations for ISAAC, such as offering fallback translations if the primary keyword is absent, setting cumulative offsets to zero for the first jitter position, recognising different observation-type values, and improved the test for the source to be a standard for ESO data. More headers are tested for a waveplate angle leading to greater robustness when the polarimetry recipes are released. These improvements were based upon experience of more data from a longer range of epochs.

- ORACUT now set for selection of calibrations of INGRID and ESO data. Michelle's ORACTIME is now in decimal UT days.

- The changed Michelle headers are supported.

- Michelle reductions adapted to handle the nod iterator used in observation preparation.

- UIST uses separate flat calibrations for imaging and spectroscopy.

- The KAPVIEW display recognises a boolean `key` keyword. If set to true (1), it places a key, such as the colour table, besides the graphic.

The main changes from a programmer perspective were as follows.

- Much of the ISAAC code applies generally to ESO instruments. There are new ESO/ directories for recipes and primitives. These directories are ahead of the instrument-specific directories in the search path for the recipe and primitive source code. `ORAC::Frame::ESO` and `ORAC::Group::ESO` modules from which instrument-specific frame and group methods are sub-classed.

- Sigma-clipping is allowed for certain combination methods during mosaic formation via a new SIGMA argument. There is also a new ZERO argument to permit optional zero-point shifts between contributing frames. While the default continues to apply intensity offsets, Fabry-Perot wavelength mosaics set ZERO to false.

- There is a new POLARIMETRY argument for some of the registration primitives.

- In aperture photometry primitives the centroid search boxsize is now decoupled from the aperture size via a BOXSIZE argument.

- LAMP is a new value for internal header ORAC_OBSERVATION_TYPE to support internal lamps used to make flats.

- The handling of the filters and standards' catalogues has been redesigned in the photometry primitives to be be more transparent and to make it easier to use multiple catalogues of standards. A waveband is interchanged rather than a column index.

- _CALC_STOKES_ has a boolean DEBIAS argument to control the application of the statistical-bias correction.

- _CALC_STOKES_NOD_CHOP_ has a new boolean CENTROID argument, whose valued is passed to _COMBINE_CHOPPED_SOURCE_.

H   Release Notes—V4.0

The main changes are the addition of support for UIST, ISAAC, and INGRID instruments.

H.1   New recipes

DARK_AND_BPM The design of the data-handling system means a traditional ARRAY_TESTS is not possible for UIST. Thus there are two recipes to perform the equivalent steps for darks not in the same observation group. This recipe uses a long-exposure DARK frame to locate bad pixels, then creates and files a bad-pixel mask. It then measures and reports the dark current from a long-exposure DARK frame. If this recipe is run at UKIRT, the dark current is stored in an engineering log file.

MAKE_BPM Creates and files a bad-pixel mask from a long-exposure dark by thresholding $+/- 5$ clipped standard deviations about the clipped mean.

MEASURE_READNOISE This is the second UIST-specific recipe replacing ARRAY_TESTS. This recipe measures the readnoise for a group of UIST DARK frames and files that measurement with the calibration system. If this recipe is run in a UKIRT environment it will create or append to a log file located at `/ukirt_sw/logs/uist_array_tests.log`.

NOD_SKY_FLAT_THERMAL This is like NOD_SELF_FLAT_NO_MASK, but it expects that the nod throws alternate to sky. The sky frames alone make the flat.

H.2   Modified recipes

- ARRAY_TESTS   The log file is now called <instrument>`_array_tests.log` where <instrument> is the lowercase instrument name.

- NOD_CHOP_FAINT   There is now an option to create and display a quality map of the combined source images.

- NOD_SELF_FLAT_NO_MASK   This recipe and NOD_SELF_FLAT_NO_MASK_APHOT now do not clean deviant pixels from the individual frames contributing to the flat. Bad pixels in the mosaic are not filled. The difference-pair frames (suffix `_dp`) are no longer removed upon recipe completion.

- REDUCE_DARK   This now supports variance creation and propagation by default. There is a new steering parameter USEVAR to disable this.

H.3   Global changes

The main changes from a user perspective were as follows.

- Updated and expanded documentation, particularly the description of the primitives and lists the main primitives used at each stage, mention of **xoracdr**, and many more hyperlinks.

- Introduction of environment variable `ORAC_KEEP` to retain intermediate frames.

- Environment variables `ORAC_DATA_IN` and `ORAC_DATA_OUT` can define paths relative to the current working directory.

- **xoracdr** now knows the default calibrations for all instruments. A bug which prevented **xoracdr** operating on non-networked computers is fixed.

- Calibration reference offset added for off-centre nod patterns. This and the baseshift offset used array references. Arrays of values in a `-calib` can be comma separated.

- A bug affecting ORACTIME in calibrations for Michelle fixed; it was previously set to 0 in all cases.

- Cater for early Michelle data which had missing metadata affecting six headers. Allow for missing, undefined, or malformed RA_BASE and DEC_BASE headers in UFTI data.

- Allow for old UFTI data with CTYPE1 header set erroneously to 'Detector Rows' to create WCS.

- For IRIS2, modify the header translations for filter name and observation mode; correct the airmass calculation; and allow alternate frames to have alternate files, but still be members of the same group by using different headers for group membership checks.

- There is no longer the assumption of a right-handed world co-ordinate system in co-ordinate transformations benefitting UIST and future instruments with an extra reflection in the optical train.

- There are additional checks for significantly negative frame means after dark subtraction. Depending on the severity this may stop the pipeline, or merely issue a warning. This is to ensure that valid darks are observed.

- The broadened mean method has been reinstated for combining frames into a flat. This avoids the artefact in self-flat fielding where in an n-point jitter pattern, $1/n$ pixels divide by themselves giving rise to a false peak at the normalisation factor in the histogram of values in a flat-fielded frame.

- A possible cause of cause of a negative mode in is given in the warning announnced by _NORMALISE_TO_MODE_.

- _GENERATE_OFFSETS_EXTENDED_ now uses the 98 percentile, the same as the other _GENERATE_OFFSETS_ primitives.

- Centroiding now has a default smoothing option to make object location more robust for fainter sources.

- A double-pass median filtering of bad pixels now occurs just prior to the near-infra-red photometry to reduce bias from unmeasured flux. It particulalry helps the flat-fielded images, as the mosaics almost invariably lose their bad pixels in the region of interest during their formation.

- If variance information is available, the aperture photometry uses it to estimate photometric errors.

- Tidying correctly erases NDF components within HDS container files. Allow for `_bc` files in relevant _TIDY_ primitives.

- Support added to remove electronic ghosting, correct residual bias, and resample for field distortion, all initially for ISAAC.

The main changes from a programmer perspective were as follows.

- Calibration index files permit user header (uhdr) to be in expressions for bias, dark, flat, and sky.

- Infrastructure uses perl modules `Astro::FITS::Header` for access to FITS headers, and `Starlink::HDSPACK` for manipulation of HDS components.

- Add the internal headers ORAC_TELESCOPE, ORAC_X_REFERENCE_PIXEL, and ORAC_Y_REFERENCE

- There is a `$Grp->memberindex( $Frm )` method which returns the position the supplied frame has in the group, starting from 0. This is useful in batch-mode processing where the groups are pre-populated.

- _CLIPPED_STATS_ has new NUMBER argument returning the number of good pixels.

- _DIVIDE_BY_FLAT_CHOP_SKY_ and _DIVIDE_BY_FLAT_EXTENDED_ permit flat-fielding of sky frames via a new argument FLATSKY.

- There are some new primitives including _GET_FILTER_PARAMETERS_ to encapsulate filter attributes, and _FORM_SKY_LEVELS_ to determine the modal sky level for the sky frames after flat-fielding.

- Script `nongeneric_imaging.csh` is moved from the `etc/` to the new `admin/` directory.

- Lexical handles rather than global variables are used for accessing files.

I   Release Notes—V3.1-1

The main change is the addition of support for IRIS2.

I.1   New recipes

ADDWCS Adds a valid WCS in the FITS headers of raw data. This recipe is for all instruments.

ARRAY_TESTS This is for IRIS2. It calculates and reports the readnoise and dark current using a group of four frames taken in the array tests sequence. The readnoise and dark current are logged to a text file, and the readnoise is filed with the calibration system.

DARK_SUBTRACT Subtracts a dark frame from an observation. This recipe is meant for a quick look of data that are being taken at a high rate.

NOD_CHOP_FAINT Reduction of nodded and chopped data, specifically for Michelle. It is similar to NOD_CHOP, but first removes horizontal and vertical artifacts from the mosaic, then forms a single image of the source using a median filter, and finally smooths the combined image to enhance the visibility of faint sources.

NOD_CHOP_SCAN Reduction of nodded and chopped data taken in a scan pattern, specifically for Michelle. It is similar to NOD_CHOP, but will create a mosaic from nodded and chopped groups taken in a scan pattern.

REDUCE_FLAT Reduces a flat field by masking bad pixels, subtracting a dark, and normalising the result. This recipe is for all instruments.

I.2 Modified recipes

- NOD_CHOP_APHOT The photometry aperture has been increased to 5 arcseconds from 3, and inner and outer diameters of the sky annulus have been modified correspondingly. The reported photometry uses filter zero points and widths, the latter to find and report the broadband-*N* equivalent magnitude and flux for narrowband-*N* filters. Extinctions are now included.

  The recipe now determines the baseshift calibration, which measures the displacement of the source images from the nominal locations.

- ARRAY_TESTS For Michelle the estimated readout noise is no longer reported.

I.3 Global changes

The main changes from a user perspective were as follows.

- Undefined or nonsense headers used by the recipes are replaced internally to pipeline, but not in the physical files, This usually permits the pipeline to complete. Recipes report the names of modified headers, for manual checking and correction of the raw data.

- Further sanity checks are included for Michelle. The waveform header is validated, and incorrect combinations reported. The mean data value must lie between 25000 and 48000 counts to prevent a warning message from appearing.

- Recipes make an on-the-fly night log in `$ORAC_DATA_OUT`. Night logs contain the group number. The file name has an `_im` suffix for Michelle.

- The default UFTI gain reflects the gain's changing values arising from the use of a different controller.

- NDF blank titles are removed.

- The UFTI bad-pixel mask has changed, now having 0.3% more bad pixels.

- Centroiding is protected against data comprising all bad values.

- The hybrid registration has a further improvement for a special case.

- The near-infrared photometry uses the latest standards list, which also includes standards for the *IZJHKLM* wavebands.

- In photometry recipes the sky units are now reported as counts/s/pixel.

- Scripts have improved error detection and reporting after accessing text files.

- The default display suffices have been augmented. These are for Michelle `_cab`, `_scab`, `_cpc`, and `_rpc`; and `_dk` for IRCAM.

- The calibration called mask, should be used to specify other than the default bad-pixel mask, rather than the bpm calibration.

The main changes from a programmer perspective were as follows.

- There is better structuring of the preliminary steps to permit instrument-specific variants, such as correcting the world co-ordinate system headers.

- The waveplate angle header values are standardised to a single format.

- The preliminary operation includes the removal of axes after a rearrangement in ORAC::Convert.

- The registration was modified to allow for changes to CCDPACK.

- The _COMBINE_CHOPPED_FRAME_ primitive as used by the NOD_CHOP family of recipes has new options. These comprise centroiding with validation, the method by which the individual chopped and nodded images are combined, the ability to apply block smoothing for faint sources, and the removal of pickup and bias variations. The removes column then row patterns, by subtracting the median of each column or row from all the values in that column or row.

J   Release Notes—V3.0-3

The main changes are the addition of recipes for Michelle nodded and chopped observations, and a rearranged directory structure for new and multi-mode instruments.

J.1   New recipes

ARRAY_TESTS This is for Michelle. It derives a mean bias file, and calculates and reports the read noise, using the latter pair of minimum-exposure frames in the array tests sequence. Both the bias frame and readnoise are recorded in the calibration system.

NOD_CHOP Reduction of nodded and chopped data, specifically for Michelle. Each recipe cycle comprises four frames located at two nod positions and the A and B beams. The individual beams are differenced and then so are successive pairs of observations. The differenced images are then combined into a mosaic. On successive cycles the mosaics are co-added.

NOD_CHOP_APHOT As NOD_CHOP, but it also extracts, registers, and combines the two positive and two negative images of the source. Then it performs 3-arcsecond aperture photometry on the combined image, logging to a small text list. It compares the object name against a file of N, Q standards to determine the zero point. No extinction correciton is currently applied, as the coefficients have yet to be determined.

MOVING_NOD_CHOP As NOD_CHOP, but mosaic registration is adjusted to track the motion of an asteroid or compact comet using ephemeris data.

POL_ANGLE_NOD_CHOP Reduces chopped and nodded poliarmetry data of point and small ($< \sim 10$ arcsec) extended sources, specifically for Michelle. The data are expects to iterate over waveplate angle before the telescope is nodded. The recipe forms integrating mosaics as NOD_CHOP, for each of four waveplate angles. For each mosaic it combines the two positive and two negative images of the source. The recipe then calculates polarisation frames and catalogues of vectors from these as earlier recipes like POL_JITTER.

POL_NOD_CHOP As POL_ANGLE_NOD_CHOP, except the data are ordered such that the telescope performs its nodding pattern before the waveplate is turned.

J.2   Modified recipes

- ARRAY_TESTS   Instrument-specific versions for UFTI and IRCAM. Both record the read noise in the calibration system.

J.3   Global changes

The main changes from a user perspective were as follows.

- Correct data variance creation for UKIRT infra-red data. The previous calculations were for CCD data.

- Easier to switch on data-variance processing. This combined with handling of chopped data has caused some reordering of the early steps; for instance the editing of the world co-ordinate system (WCS) occurs just before dark subtraction, rather than immediately after the _IMAGING_HELLO_ preliminaries. Thus a few of the early frames such as _db and _bp now do not have a WCS defined.

- More-efficient masking using EXTRACTOR instead of PISA. Some of the acceleration is because some of the steps to sweeten the data for PISA, such as the removal of the bad pixels, are now unnecessary.

- Flat-field creation now uses an unweighted median to give approximately equal weighting to the contributing frames. The previous version of MAKEFLAT was supposed to weight the values but had bugs. The corrected task weights by the data value in the absence of a variance array assuming pure Poisson statistics, which is in appropriate for infra-red data. The unweighted median gives similar results to earlier ORAC-DR versions. There is a beneficial exception where the earlier MAKEFLAT biassed towards certain contributing frames, and the unweighted median gives a more equitable division.

- An improved option for registration is available, although it is not the default. It uses the WCS to only compare sources in the overlap regions, and permits a match using a single source, provided it is within 12 pixels of the nominal WCS position.

- For moving-target registration and the ephemeris file, a bug affecting object names containing spaces is fixed.

- Better formatting of output with blank lines to block related output, and some of the commentary contain more details. The historical and unnecessary "Orac says: " prefix was removed from the commentary.

- The nearest-neighbour registration immediately prior to forming a mosaic had assumed that the brightest object with identification number 1 would be present in all the offset files. In rare cases, this may not be true. For instance, if the centred target is a faint or low surface-brightness galaxy at low galactic latitude, brighter stars relegate the galaxy to a high identification number. So the code now checks for a common-denominator object between all the frames with a higher identification number.

- The mosaic-making primitive has been partly restructured to make the code more obvious, and it also rationalises the naming. Gone is the `_mu` file. The mosaic with bad pixels filled has its own `_fb` suffix. All individual-cycle mosaics are retained. The only disadvantage is that for a single-cycle observation, there are two copies of the same mosaic, one with suffix `_mos` and the other with suffix `_mos_0`; the recipe cannot know if there is a second cycle to come.

- The mosaics have the world co-ordinate system domain set to sky so that displays with GAIA and KAPPA have sky co-ordinates.

- The start and end UT times for mosaics are updated to that of the first- and last-contributing frames.

- The photometry results file now reports the sky in counts per second (previously just counts), and the exposure time. The alignment of the columns is thus slightly altered.

- The recipe tidying has been improved to remove all unwanted files. Some of the registration text files, certain suffices, and later-cycle frames were being missed.

- In the polarimetry calculations, the chip position angle is added to the offset of north with respect to the analyser, so that the vectors are also measured with respect to north. This omission had been giving vector orientations approximately 89 degrees too high for IRCAM.

- More allowance for occasional problems with the headers in the raw data, *e.g.* the base R.A. in degrees not hours and defunct instrument read modes.

- In earlier versions, there were fatal errors which the recipes reported, but allowed the pipeline to continue. These are now corrected so that the recipe ends its processing.

- Expanded tutorial documentation, particularly of the description of algorithms used by the primitives and more hyperlinks, and updated for the new directory structure. Various minor improvements, new hyperlinks and corrections to the recipe documentation.

The main changes from a programmer perspective were as follows.

- The primitives access user headers for steering.

- The FITS header information is accessed through generic user headers with the prefix `ORAC_`. This insulation permits common code for multiple instruments.

- There were upgrades for the latest versions of KAPPA and ARD, notably for further use of the world co-ordinate system. In general recipes use the pixel domain for tasks like CENTROID and PSF, recording the former domain, and then resetting WCS frame after using the KAPPA task.

- There is a reorganised directory structure for the recipes and primitives. These are divided into instrument-specific; general; or by topic, such as imaging or spectroscopy. The topic directories also have subdirectories for specific instruments. This restructuring permits code reuse for current and new instruments. It is also now possible to have recipes and primitives with the same name for different topics or instrument. Such scripts do have similar functions, but the exact processing depends on the data topic or some instrument attribute.

K   Release Notes—V2.1-0

The main changes are the addition of spatially jittered Fabry-Perot recipes, and recipes for compact comets.

K.1   New recipes

FP_JITTER Reduction of a Fabry-Perot observation, comprising eight frames, on and off the source, and on and off the spectral line both to the blue and to the red. This is repeated for a series of spatial positions of the source.

FP_JITTER_NO_SKY Reduction of a Fabry-Perot observation, comprising four frames, all on the source, and on and off the spectral line both to the blue and to the red. This is repeated for a series of spatial positions of the source.

MOVING_QUADRANT_JITTER As QUADRANT_JITTER but registration is adjusted to track the motion of a comet using ephemeris data. The comet should <45 arcsec diameter for UFTI, or <10 arcsec for IRCAM. Compared with MOVING_JITTER_SELF_FLAT, this recipe avoids cometary artifacts appearing in the flat field.

QUADRANT_JITTER_TELE As QUADRANT_JITTER, but registers using the telescope offsets. This is used for observing compact comets (limiting angular sizes as above), when the telescope has tracked the nucleus.

K.2   Modified recipes

- ARRAY_TESTS For UFTI, the ADU conversion is obtained from the GAIN header, rather than being hardwired at 7.0.

- CHOP_SKY_JITTER This now functions correctly for multiple cycles of the recipe.

- FP Documentation improvements especially in the description. Primitive _FP_STEER_ has a new steering parameter, NPAIRS, and parameter NUMBER has changed to its normal meaning.

- NIGHT_LOG This can start from observation numbers other than 1. A bug has been fixed where the dimensions appeared as zero for UFTI. It arose because certain headers no longer existed after 2000 August.

- SKY_FLAT_FP This is no longer limited to eight frames.

- SKY_AND_JITTER A bug has been fixed where some intermediate files were not being removed for this recipe and its variant.

K.3   Global changes

The main changes from a user perspective were as follows.

- The aperture for _APHOT recipes is now 5 arcseconds.

- A new photometry catalogue `fs2001.dat`, supplied in original form by Sandy Leggett, is used by the _APHOT recipes. The new catalogue contains *IZLM* magnitudes for the first time, and the *JHK* data have been refined to account for recent observations. The 2000 edition is accessed for *JHK* photometry if the 2001 catalogue is unavailable.

- Another new UFTI bad-pixel mask. Old masks are available on request.

- A bug that affected some rare mixed-method registrations has been fixed. The y offset had the wrong sign.

- Allow for a special case in the mosaicking registration, when the nearest-neighbour method is used to align the various images, *and* an automatic multiple-object registration was found. It was possible not to find a common object identifier in all the fields when the target is a low-surface brightness galaxy. While this is still possible, it is far less likely than before.

- The IRCAM saturation level was refined upwards to 20000, or 33000 if header SPD_GAIN is "Deepwell". SPD_GAIN is created by the recipe when it is absent, based upon the value of the detector bias.

- The message concerning an AST SKY Frame creation, which could be confused with a data frame of blank sky, has been clarified. The same script allows for a missing CROTA2 header in old data when inserting world co-ordinate system headers.

- Various minor improvements and corrections to the documentation, such as correcting the former prefix for UFTI mosaics in the reference section, and excluding references to IRCAM in the FP recipes.

L   Release Notes—V2.0-1

The major changes are the move to generic recipes, and the introduction of many new recipes, especially for polarimetry and Fabry-Perot data.

L.1   New recipes

CHOP_SKY_JITTER Reduction of alternating sky-target jitters for extended objects of size comparable to the detector's field of view.

CHOP_SKY_JITTER_BASIC A basic (faster) version of CHOP_SKY_JITTER.

FP Reduction of a Fabry-Perot observation, without jittering.

JITTER_SELF_FLAT_NCOLOUR Reduction of multi-colour standard jitters. This will become the new JITTER_SELF_FLAT once the recipes are colour generic.

POL_ANGLE_JITTER Reduces an imaging polarimetry observation, in which the waveplate angle iterates at each jitter position.

POL_EXTENDED Reduces an imaging polarimetry observation of an extended source.

POL_JITTER Reduces an imaging polarimetry observation, in which the spatial is position jittered before moving the waveplate angle.

SKY_FLAT_FP Creates and files a Fabry-Perot flat field derived from eight frames, using object masking.

SKY_FLAT_POL Creates a polarimetry flat field derived from eight frames (two at each waveplate angle), using object masking. It copies the flat for each waveplate angle and files them.

SKY_FLAT_POL_ANGLE Creates and files polarimetry flat fields derived from jittered frames at each waveplate angle, using object masking.

L.2   Modified recipes

Many former recipes had numerous variants for different jitter sizes. These have largely been superseded by generic equivalents. Only the EXTENDED recipes await conversion. The families of recipes changed are listed below.

- JITTER_SELF_FLAT (Six recipes in the family) There were three-, five-, seven-, and nine-point versions, but most were only available for one or two recipe variants.

- MOVING_JITTER_SELF_FLAT (Two recipes) These were limited to nine-point jitters.

- NOD_SELF_FLAT_NO_MASK (Four recipes) There were limited to fixed sizes of four- and eight-point jitters.

- SKY_AND_JITTER (Two recipes) Only five-point jitters were available.

In addition certain recipes had a fixed jitter size, but no longer. These families are as follows.

- BRIGHT_POINT_SOURCE (Two recipes) These were formerly restricted to five-point jitters.

L.3   Global changes

The main changes from a user perspective were as follows.

- Recipes and primitives are instrument generic.

- Improved registration allowing mixed solutions. There is a new offset type—beam separations—for combining polarimetry mosaics.

- Thresholding of dark-subtracted data to prevent bizarre values affecting the pipeline processing. In addition JITTER_SELF_FLAT and SKY_FLAT recipes and their variants which use object masking to make a flat, now have the deviant pixels of the initial flat-field frame reflagged as bad.

- Editing of the FITS headers to create a world co-ordinate system which GAIA and KAPPA recognise. Also pre-ORAC IRCAM data have their headers structured and comments edited to bring them closer towards the UKIRT FITS standard, and to make the headers easier to read and comprehend.

- The names of flats have changed. The filter name is included for easy identification. Certain characters have special meaning to HDS, therefore `[]{}` are removed and a decimal point becomes a `p` in the flat's name. The `_cycle` suffix is shortened to `_c`, and all `_subgrp` strings removed from the group number. Flats are are not combined over multiple cycles over recipes. SKY_FLAT and SKY_FLAT_MASKED are limited to jitters between three to five points for compatibility with other ORAC tools. It is still possible to make private variants of these recipes in which the number of jitter positions is set by the NUMBER steering parameter.

- Mosaics are combined using the mean at each pixel. This was formerly the median. The change was made to correct the photometry. Poor registration from telescope offsets due to sparse fields leads to multiple peaks in the mosaic's grid, and given the steep point-spread function's profile, the median preferentially selects pixels not at a peak. This resulted in a typical underestimate of the flux of standard stars by 1–3%. The _MAKE_MOSAIC_ primitive now has an argument to select various estimators should you prefer not to use the mean.

- For the aperture photometry the default sky annulus radii have been increased. This reduces the error estimating the sky level, both statistically and from the extend low-level pedestal in the point-spread function. The area is increased $2.9\times$ for IRCAM and $5.6\times$ for UFTI.

- To counteract the spike artifact in the histogram of sky values of flat-fielded frames where the pre-flattened frame itself contributed to the flat ('self-flat' recipes), the mode is now calculated using multiple standard-deviation clipping for JITTER_SELF_FLAT_APHOT and NOD_SELF_FLAT_NO_MASK_APHOT. The previous estimation was from iterative

application of Chauvenet's criterion and using the $3 * \text{median} - 2 * \text{mean}$ formula, and lead to a underestimate of the sky level. For typical standard stars this systematic error led to a brightening of 1–2%. The latter method is still used for photometric recipes which do not self flat, such as BRIGHT_POINT_SOURCE_APHOT. If you create and apply a 'superflat', the artifact is much reduced and therefore, the former estimator is appropriate.

- More and better processing-status messages. For example, all floating-point numbers are now reported with a sensible number of decimal places, and the names of calibration frames are reported.

- Addition of FWHM to the aperture photometry results and small text list. The file name column is 3 characters wider to accommodate the positive and negative suffices of recipe NOD_SELF_FLAT_NO_MASK_APHOT.

- The saturation level in the aperture photometry was a constant. Now it is set to the appropriate value by instrument and mode.

- It is possible to use versions of KAPPA other than the latest. The changed argument lists in various tasks are adjusted for the KAPPA version.

- More intermediate files, mostly the text files, are removed.

- Pipeline activates two more application engines: POLPACK and CATSELECT.

- A new UFTI bad-pixel mask.

- Added waveplate angle to flat rules file (`$ORAC_DATA_CAL/rules.flat`). The angle defaults to zero if it does not have a value in the FITS headers.

- Bug fixes and documentation improvements, especially links in the Perl POD.