

The SCUBA Data Reduction Pipeline: ORAC-DR at the JCMT

Tim Jenness, Frossie Economou

Joint Astronomy Centre, 660 N. A‘ohōkū Place, Hilo, HI 96720

Abstract. The ORAC data reduction pipeline, developed for UKIRT, has been designed to be a completely general approach to writing data reduction pipelines. This generality has enabled the JCMT to adapt the system for use with SCUBA with minimal development time using the existing SCUBA data reduction algorithms (SURF).

1. Introduction

ORAC-DR (e.g., Economou et al. 1998; Economou et al. 1999) is a flexible and modular pipeline originally developed by the United Kingdom Infrared Telescope (UKIRT) for the on-line reduction of data from infrared instruments as part of their ORAC project (Bridger, Wright & Economou 1998). This paper describes our experiences with using ORAC-DR at the James Clerk Maxwell Telescope (JCMT) for the reduction of SCUBA (Submillimetre Common User Bolometer Array) data.

The ability of the ORAC-DR pipeline to handle data from instruments as disparate as an infrared grating spectrometer and a submillimetre bolometer array is an indication of the flexibility of the system design.

2. Recipes

Recipes in ORAC-DR consist of a series of data reduction steps (primitives) containing instructions for the reduction of data. Recipes may be written or changed by support scientist with the aid of an editor or, shortly, a drag-and-drop-type GUI.

This is an example of a recipe used for reducing SCUBA photometry data:

```
_PRE_PROCESS_  
_FLAT_FIELD_  
_EXTINCTION_CORRECT_  
_CLIP_BOLOMETERS_ NSIGMA=3.0  
_REMOVE_SKY_NOISE_JIGGLE_ BOLOMETERS=all MODE=median  
_CALC_PHOT_RESULT_  
_CALIBRATE_PHOTOM_DATA_  
_PRINT_RESULTS_PHOTOM_FRAME_  
_PHOTOM_GROUP_
```

3. Primitives

ORAC-DR primitives contain perl code for the manipulation of data in a given state. They are written in an object-oriented fashion, manipulating objects associated with the individual data frames as well as groups of observations. The steps that involve actual processing of data (as opposed to housekeeping etc tasks) are done via a messaging request to an algorithm engine resident in memory.

Here is an example of the `_CLIP_BOLOMETERS_` primitive used by the above recipe. SCUBA has multiple sub-instruments, that simultaneously take data at different wavelengths (e.g., at 450 and 850 microns). This primitive performs a median clipping of the data from all sub-instruments associated with the current frame object. The actual clipping is performed by a task (in this case `scuclip`) in the SURF monolith. SURF (SCUBA user reduction facility, Jenness & Lightfoot 1998a,b) is written in FORTRAN and accepts ADAM messages. The details of the messaging are hidden from the primitive writer, who is typically a software engineer supporting an instrument.

```
# default the NSIGMA parameter if not specified
$nsigma = ($_CLIP_BOLOMETERS_{NSIGMA} || 5.0);

# loop over sub-instruments
for ($i = 0; $i < $Frm->nsubs; $i++) {
  # Generate output filename from the input
  ($in, $out) = $Frm->inout("_clip", $i+1);
  # Execute the action
  $Mon{surf_mon}->obeyw("scuclip",
    "in=$in out=$out nsigma=$nsigma");
  # Update the filename of the current frame object
  $Frm->file($i+1, $out);
}
orac_print "Bolometers clipped at $nsigma sigma\n";
```

Because primitives always manipulate objects associated with the pipeline, they are order-ignorant, i.e., no assumptions are made about the file number, file name, or filename convention. These behaviours are all handled by instrument-specific classes, thus allowing the re-use of primitives elsewhere in different recipes for different instruments.

4. Adding New Instruments

The object-oriented design of ORAC-DR allows new instruments to be added simply by sub-classing existing instrument objects and providing modified methods as needed. The only change required to the core pipeline code is to add the class definitions for the new instrument.

This section describes the modifications necessary to add SCUBA support to ORAC-DR.

4.1. Algorithm engines and messaging system

A single file is provided for each instrument that defines the necessary data reduction packages and the related messaging system. For SCUBA, it is necessary to use the SURF package (this provides the SCUBA specific data processing

tasks) and the Starlink KAPPA data reduction system (Currie & Berry 1998) (for general purpose data file manipulation). Both these packages use the NDF data format and the ADAM messaging system.

4.2. Frame definition

A SCUBA observation (or Frame) differs from a UKIRT observation in a number of ways and these differences are handled by the `ORAC::Frame` class. The SCUBA variant (`ORAC::Frame::SCUBA`) must do the following:

- Handle the different file naming convention
- Provide translation from SCUBA FITS keywords to ORAC keywords
- Cope with multiple sub-instruments

The first two are fairly simple to implement, but the final issue comes about because SCUBA can observe at up to 4 wavelengths simultaneously with all the data being stored in a single file. Mid-way through the data reduction a file is generated per wavelength and the pipeline must keep track of these files separately. Each primitive is then designed to process each sub-instrument.

4.3. Group handling

The SCUBA observing system does not have an inbuilt notion of an ‘observation group’. The major driving force behind the adoption of a pipeline for SCUBA was to provide a way of co-adding separate observations automatically in order to reduce the load on a service observer. The SCUBA implementation of ORAC-DR differs from the UKIRT system in that it must calculate group membership itself from header information (source name, observing mode, position etc) rather than simply querying the header for a group name. To deal with the occasion when an observer does not want an observation to be included in the group (even when ORAC-DR thinks it should be included) a facility is provided for removing certain observations from group operations.

Other differences between SCUBA and UKIRT groups relate to the sub-instrument problem. The group handling primitives must co-add all members of the group at all available wavelengths. Group handling is particularly complex in SCAN/MAP data reduction since there is the added complication of combining observations taken with different chop configurations

4.4. Calibration

Calibration of submillimetre data is completely different to near-infrared/optical calibration and a completely new calibration class was required for SCUBA. Two types of calibration are required:

- Correction for atmospheric extinction.
- Compensation for instrumental gain variations

The atmospheric extinction (sky opacity) can be determined by performing skydip observations (i.e., measuring the sky brightness temperature at different elevations and fitting the data with an atmospheric model) at a single wavelength and then extrapolated to the current wavelength.

Gain calibration is more complicated and has, as yet, only been implemented for photometry observations. In this case a flux calibrator (usually a planet) is observed and a gain derived that is used for further observations.

4.5. Recipes and primitives

New recipes and primitives had to be written since SCUBA is completely different to a near-infrared/optical instrument but this was possible without having to construct a data reduction pipeline from scratch. The major effort in adding SCUBA to ORAC-DR has therefore been entirely focused on the actual data reduction recipes rather than on infrastructure.

5. Conclusions

The JCMT have adopted UKIRT's ORAC-DR data reduction pipeline for processing SCUBA data. This pipeline is now in use at the telescope and provides useful information on the progress of observing in near-real-time. The modular and object-oriented design of the ORAC-DR pipeline has allowed the addition of SCUBA to be made without significant expenditure of effort on pipeline infrastructure. Programming effort has concentrated on writing data reduction recipes and is now benefiting from the decision to share pipeline code with another telescope.

The pipeline has been designed so that it can be used off-line or on-line and it is intended that observers will use the system at their home institution or as a quick-look system for database retrieval.

References

- Bridger, A. B., Wright, G. S., & Economou, F. 1998, in ASP Conf. Ser., Vol. 145, *Astronomical Data Analysis Software and Systems VII*, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 292
- Currie, M. J. & Berry, D. S. 1997, *KAPPA – Kernel Application Package User's Guide* (ver. 0.13) (Starlink User Note 95, CCLRC/Rutherford Appleton Laboratory)
- Economou, F., Bridger, A., Wright, G. S., Rees, N. P., & Jenness, T. 1998, in ASP Conf. Ser., Vol. 145, *Astronomical Data Analysis Software and Systems VII*, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 196
- Economou, F., Bridger, A., Wright, G. S., Jenness, T., Currie, M. J., & Adamson, A. 1999, this volume, 11
- Jenness, T., & Lightfoot, J. F. 1998a, in ASP Conf. Ser., Vol. 145, *Astronomical Data Analysis Software and Systems VII*, ed. R. Albrecht, R. N. Hook, & H. A. Bushouse (San Francisco: ASP), 216
- , 1998b, *SURF - SCUBA User Reduction Facility User's Manual* (ver. 1.3) (Starlink User Note 216, CCLRC/Rutherford Appleton Laboratory)