 

**SPICA-VIS**

**Data Flow and Data Reduction Software**

Authors: Philippe Berio and Denis Mourard

Doc. No. : SPICA-VIS-0006

Issue: 2

Date: 10/07/2020

**CHANGE RECORD**

|  |  |  |  |
| --- | --- | --- | --- |
| **ISSUE** | **DATE** | **SECTION** | **COMMENTS** |
| 1 | 09/03/2020 | All | DM: creation |
| 1 | 18/03/2020 | 7.1 | PBE: Pipeline description |
| 2 | 30/06/2020 | All | PBE, DMO: ready for Design Review |

Table of Contents

[1. Scope 4](#_Toc44398373)

[2. Introduction 4](#_Toc44398374)

[3. Science Group - List of targets 4](#_Toc44398375)

[4. Generation of the list of OB and scheduling 5](#_Toc44398376)

[5. Observing Software 5](#_Toc44398377)

[6. SPICA Data Reduction Pipeline 5](#_Toc44398378)

[6.1. Data Reduction Cascade 6](#_Toc44398379)

[6.2. Processing Modules 7](#_Toc44398380)

[6.2.1. spica\_cal\_det 7](#_Toc44398381)

[6.2.2. spica\_disp\_law 7](#_Toc44398382)

[6.2.3. spica\_est\_kappa 7](#_Toc44398383)

[6.2.4. spica\_raw\_estimates 7](#_Toc44398384)

[6.2.5. spica\_cal\_oifits 8](#_Toc44398385)

[6.3. Automatic Pipeline 8](#_Toc44398386)

[6.4. Calibration Pipeline 8](#_Toc44398387)

[6.5. Database feedbacks 8](#_Toc44398388)

# Scope

The scope of this document is to describe the circulation of information from the science preparation to the data products.

# Introduction

The CHARA/SPICA instrument aims at measuring a large number (~1000) of stellar angular diameters over different regions of the HR diagram (see <https://lagrange.oca.eu/fr/spica-project-overview>). It will also serve different science programs with samples of a few tens to few hundreds of sources. The SPICA Science Group (SG) is preparing a dynamical list of targets and will manage the science products (*see Document SPICA-VIS-0020, Science Survey Management*). The general data flow cycle is presented in Figure 1:

*Figure 1: general cycle for the CHARA/SPICA science operation*

# Science Group - List of targets

The core science program of CHARA/SPICA is a survey of stellar parameters over a large fraction of the HR diagram. To manage this survey, the SG is considering the use of a database with a first prototype that will be based on TopCat, pending an evolution currently discussed with the JMMC. CHARA/SPICA will also serve other science programs and we may consider including or not these additional programs inside the survey database. An alternative would be that these additional programs define their own target lists and generate additional OB feeding the scheduler.

For the core science program, this work is done using Python Scripts sending requests to Vizier catalogues on different sets of parameters. It could also, in some cases, be based on a fixed list of already identified targets. The list of targets could be updated dynamically to benefit from the last update of the various Vizier database (Gaia DR3 for example). Details are given in the SSM document (*SPICA-VIS-0020*); this part of the work corresponds to Tasks 1 to 4.

# Generation of the list of OB and scheduling

This part of the program is based on ASPRO2. It includes the required functionalities for the correct setting of the instrument and an estimation of the exposure time and optimal setting of the observation. ASPRO2 will be feed on a selection of targets through search functions in the SPICA database and criteria depending on the period, strategy of observation, and a follow-up of previous observations. The new principle of the link between ASPRO2 and ObsPortal (archiving all the ESO VLTI observations) will be adapted together with the JMMC.

Based on the current list of OBs, this software will permit to extract the OBs that will be scheduled for a given observation. The system needs in input some criteria: range of declination, range of hour angle, range of magnitude, observing mode, priorities…

This part of the work corresponds to Task 5 of the *SSM document*.

# Observing Software

This software is described in *SPICA-VIS-0003*. It takes the list of OBs and execute them within the CHARA environment. It also feed the OB database with the information about the execution of the OBs. The outcome of this software is a list of data and metadata that are stored at the end of the night.

# SPICA Data Reduction Pipeline

The end-of-night pipeline executes a full data processing to assess the quality of the observations. It is connected to the OIDB database to send the L0 metadata.

The SPICA pipeline follows the architecture of the ESO pipelines. It organizes, associates and classifies the raw data, the calibration maps and the products following the ESO rules. These rules are based on specific keywords which are stored in the primary header of the FITS files.

The raw data are classified with 3 keywords: HIERARCH DPR CATG, HIERARCH DPR TECH and HIERARCH DPR TYPE (see grey box in Figure 1). The products are classified with 1 keyword: HIERARCH PRO CATG (see pink and yellow boxes in Figure 1).

## Data Reduction Cascade

The SPICA data reduction principle is presented in Figure 1.

Une image contenant capture d’écran

Description générée automatiquement

Figure 1: SPICA data reduction cascade

The match rules are based on the keywords contained in the primary header of the FITS files. The keywords to be checked are presented in Table 1.

|  |  |  |
| --- | --- | --- |
| **Match Rule** | **Keyword Name** | **Keyword Description** |
| Identical Readout | HIERARCH DET NAME  HIERARCH DET DIT  HIERARCH DET GAIN | Detector Name  Detector Frame Time  Detector Gain |
| Identical Mode | HIERARCH INS RESOL  HIERARCH INS CENTRAL WL | Spectral Resolution (LOW, MED, HIGH)  Central Wavelength |
| Identical Observation | Identical Readout +  Identical Mode +  HIERARCH TPL START  HIERARCH FT STATUS  HIERARCH NB TEL | Starting date of observation (difference between calibrator and target should be less than xx hours)  Fringe Tracker Status (NONE, TRACKING)  Number of telescopes |

Table 1: Match Rules Description

## Processing Modules

We defined 5 processing modules which allows to process all raw data and to calibrate the raw OIFITS files. The association between processing modules and raw data could be found in Figure 1 (see white boxes). All processing modules will be developed Python.

## spica\_cal\_det

This processing module process the data recorded during the template Detector Calibration (see SPICA-VIS-0001). The dark frames are processed in order to derive a BIAS map and the amplification gain of the EMCCD. The method for deriving the amplification gain is described in Martinod et al, 2018 (A&A). The sequence of processing is:

* Read the raw data
* Compute the BIAS map by averaging all frames
* Compute the histogram of the pixel values
* Determine the amplification gain from the histogram
* Save the BIAS\_GAIN map

## spica\_disp\_law

This processing module process the data recorded during the template Spectral Calibration (see SPICA-VIS-0001). The sequence of processing is:

* Read the raw data
* Correct the raw data from the BIAS map
* Average all frames
* Compute the spectrum
* Detect the spectral lines of the spectral sources
* Determine the polynomial law =f(pixel)
* Save the DISP\_LAW map

## spica\_est\_kappa

This processing module process the data recorded during the template Kappa Matrix Calibration (see SPICA-VIS-0001). The sequence of processing is:

* Read the raw data
* Correct the raw data from the BIAS map
* For all beams (6)
  + Average all frames
  + Compute the flux ratio between the photometric channel and the interferometric channel
* Save the KAPPA\_MATRIX map

## spica\_raw\_estimates

This processing module process the data recorded during the templates transfer Function and Observation (see SPICA-VIS-0001). The sequence of processing is:

* Read the raw data
* Correct the raw data from the BIAS map
* Compute the photometry of each beam
* Apply the KAPPA\_MATRIX map to the estimated photometry
* Compute the averaged power spectrum of each spectral channel for the fringes data and foreground data
* Compute the averaged bi spectrum of each spectral channel for the fringes data and foreground data
* Correct the averaged power spectrum and bi spectrum by the ones corresponding to the foreground data
* Estimate the squared visibility and closure phase
* Estimate the differential phase (method TBD)
* Save the raw OIFITS files (TARGET\_RAW\_INT or CALIB\_RAW\_INT)

## spica\_cal\_oifits

This processing module calibrate the interferometric estimates of the target by the ones of one or more calibrators. The sequence of processing:

* Read OIFITS files of target and calibrators
* Retrieve the angular diameter of the calibrators from JSDC catalog (or from other catalogues)
* Divide the visibility of calibrators by the visibility of a uniform disk model
* Interpolate the interferometric observables of the calibrators at the observing time of the target
* Calibrate the interferometric observables of the target by the interpolated ones.
* Save the calibrated OIFIST file (TARGET\_CAL\_INT)

## Automatic Pipeline

The automatic pipeline is a high-level script which groups the raw data following the matching rules and applies the corresponding processing modules to all raw data groups. This script will be developed in Python.

## Calibration Pipeline

The calibration pipeline is a high-level script which groups the products of spica\_raw\_estimates following the matching rules and applies spica\_cal\_oifits to all raw OIFITS groups. This script will be developed in Python.

## Database feedbacks

The final step of the pipeline will be to feed the different databases: OIDB for the L0 and possibly the L1 data with proprietary period, and the SPICA Survey Management Database.