FLITECAM Redux Pipeline Users Manual

SOF-US-HBK-OP10-2004

Date: January 7, 2015
Revision: -

AFRC
Armstrong Flight Research Center
Edwards, CA 93523

ARC
Ames Research Center
Moffett Field, CA 94035

German Space Agency, DLR
Deutsches Zentrum für Luft und Raumfahrt

FLITECAM Redux Users Manual

SOF-US-HBK-OP10-2004

AUTHOR:

William Vacca, USRA, SOFIA Senior Scientist	Date
Melanie Clarke, USRA, SOFIA Pipeline Engineer	Date
Sachin Shenoy, USRA, SOFIA Pipeline Scientist	Date

CONCURRENCE:

R.Y. Shuping, USRA, SOFIA DPS Lead	Date

APPROVAL:

William Reach, USRA, SOFIA Associate Director for Science	Date
William Szolnoki, USRA Business Manager	Date

REVISION HISTORY

REV	DATE	DESCRIPTION
-	1/7/15	Initial Release

FLITECAM Redux Users Manual
SOF-US-HBK-OP10-2004

Table of Contents

FLITECAM Redux Users Manual
SOF-US-HBK-OP10-2004

1 INTRODUCTION

The SI Pipeline Users Manual (OP10) is intended for use by both SOFIA Science Center staff during routine data processing and analysis, and also as a reference for General Investigators (GIs) and archive users to understand how the data in which they are interested was processed. This manual is intended to provide all the needed information to execute the SI Level 2 Pipeline, flux calibrate the results, and assess the data quality of the resulting products. It will also provide a description of the algorithms used by the pipeline and both the final and intermediate data products.

A description of the current pipeline capabilities, testing results, known issues, and installation procedures are documented in the SI Pipeline Software Version Description Document (SVDD, SW06, DOCREF). The overall Verification and Validation (V&V) approach can be found in the Data Processing System V&V Plan (SV01-2232). Both documents can be obtained from the SOFIA document library in Windchill at location: / Software Management Development or Verification / Pipelines (DPS).

This manual applies to FLITECAM Redux version 1.0.0.

2 SI OBSERVING MODES SUPPORTED

2.1 FLITECAM instrument information

The First Light Infrared Test Camera (FLITECAM) is an infrared camera, operating in the 1.0 - 5.5 µm range. It has a set of broadband filters for imaging, and a set of grisms and order sorting filters for medium resolution spectroscopy.

The FLITECAM imaging mode provides seeing-limited images at 1 – 3 □m and diffraction-limited images at 3 - 5.5 □m (McLean, et al. 2006). The array (InSb ALADDIN III) size is 1024 x 1024 with a pixel scale of 0.47” per pixel. This configuration results in a Field-Of-View (FOV) of ≈ 8.0’ but a circular stop and coma at the edges of the image restrict the usable FOV to 5.6’ (see the figure below). FLITECAM has two filter wheels that contain a set of broadband imaging filters, a suite of broad order sorting filters for spectroscopy and a few narrowband-imaging filters. Available broadband imaging filters for FLITECAM are J, H, and K. The narrow band filters are $\text{Pa}_{\square}$, $\text{Pa}_{\square}$ Continuum, 3.0 □m ice, and 3.30 □m PAH. In future the instrument team may offer other filters (H_{wide}, K_{wide}, K_{long}, L, nbL, L’, M, nbM, and L+M) to general observers. Detailed information about filter characteristics, saturation limits, sensitivities and observation planning can be found in the FLITECAM chapter of the SOFIA Observer’s Handbook (SOFIA User Support Group 2013).

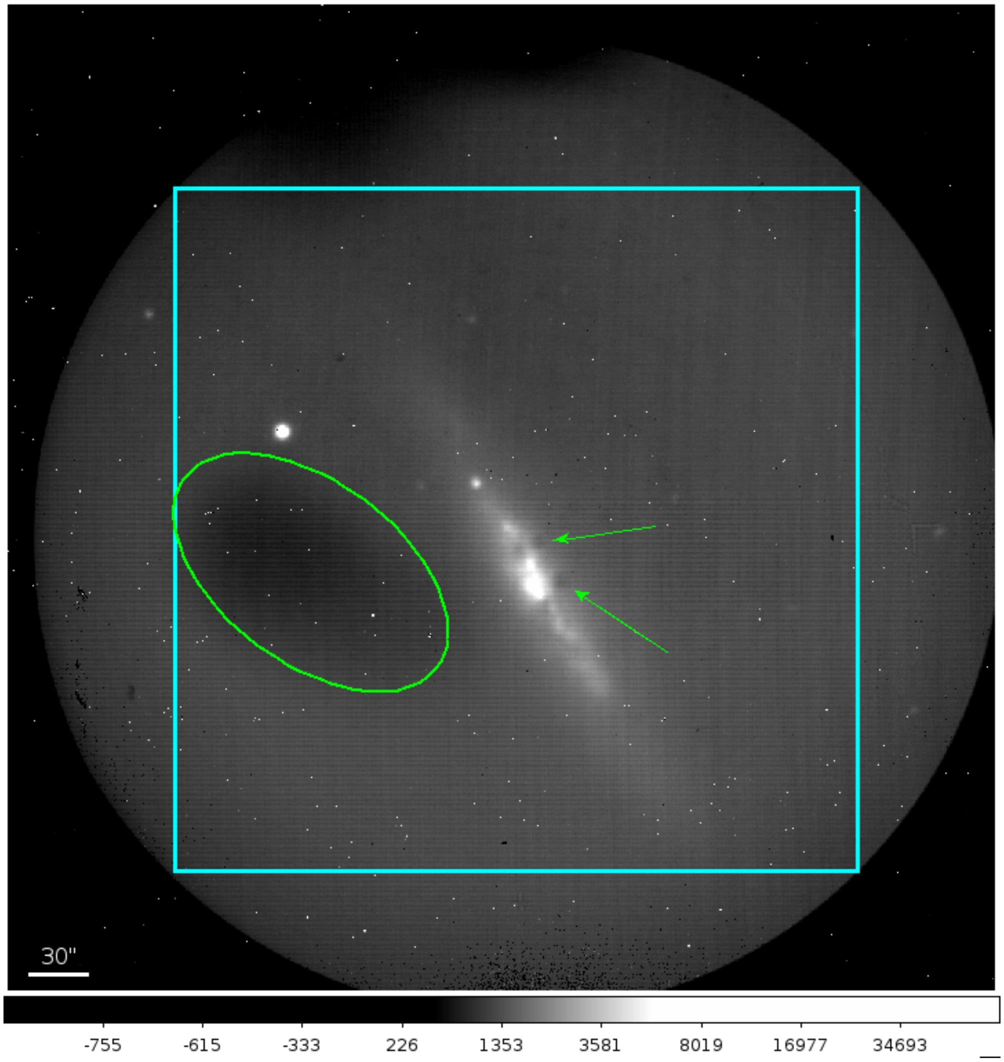


Figure 1: A typical FLITECAM image obtained during a commissioning flight. The cyan box represents the usable portion of the array. Sources that fall outside of this box show too much coma for accurate PSF measurement. The small white dots visible in the image are hot pixels. The green ellipse encompasses a region of low quantum efficiency and the green arrows show obscurations in the optical path.

FLITECAM also has three grisms and five order-sorting filters that can be combined in different ways in order to access the full wavelength range. It has two slits, a narrow slit (1”) and a wide slit (2”), which allow for higher or lower resolution, respectively. The higher resolution mode is not currently used.

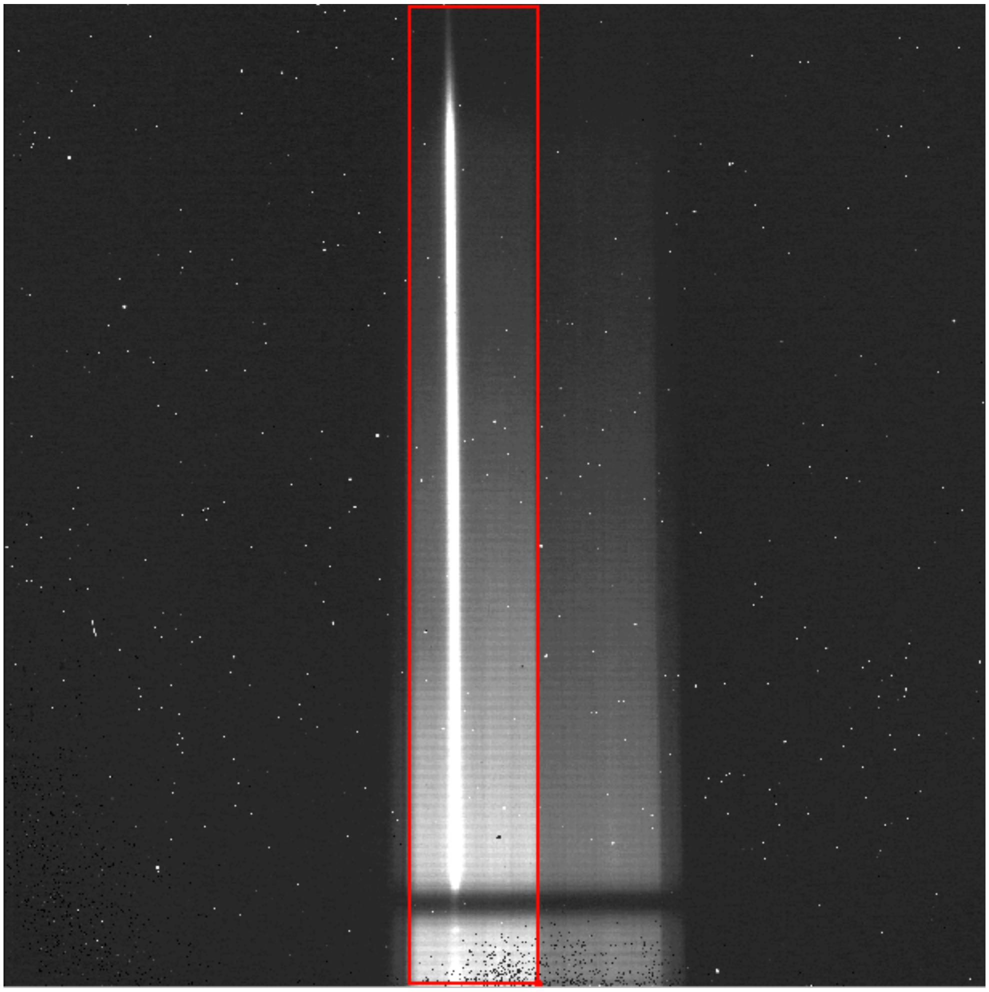


Figure 2: A typical FLITECAM grism observation taken with the wide slit. As with the raw imaging frame, there are hot pixels scattered across the frame. The wide-slit region is outlined in red; the narrow-slit region is visible to the right of the wide-slit region.

2.2 FLITECAM observing techniques

In any given pixel on a detector, the total number of counts is given by the sum of the counts from the dark current, telescope emission, sky background, and the target itself. Since the sky level can vary significantly over short timescales in the near infrared, it is typical to take pairs of observations back-to-back and subtract them in order to remove the dark current, telescope signal, and sky signal (to first order). An A frame exposure of the target is immediately followed by a B frame containing only sky at the target's position:

$\mathrm{A} = \mathrm{dark} + \mathrm{telescope} + \mathrm{sky}_\mathrm{A} + \mathrm{target}$
$\mathrm{B} = \mathrm{dark} + \mathrm{telescope} + \mathrm{sky}_\mathrm{B}$
$\mathrm{A\text{-}B} = \mathrm{target} + (\mathrm{sky}_\mathrm{A}\text{-}\mathrm{sky}_\mathrm{B})$

Note that it is assumed that the sky level may vary somewhat between frame A and frame B, but the dark current and telescope emission do not. For the spectroscopy mode, this residual sky emission is typically removed in the extraction process.

For FLITECAM imaging, there are presently two observing modes: stare and nod-off-array. Both of these modes can be run with or without dithers. We anticipate that most of the observations using FLITECAM will be done either in stare with dither mode for un-crowded Field-Of-View (FOV) or nod-off-array with dither for crowded and extended emission FOV. In the first case, all dither positions are combined to produce a sky frame that can be subtracted from each image. In the second case, only the sky frames are combined and subtracted from the on-source frames.

FLITECAM grism observations offer two modes for producing the AB pairs. In the nod-along-slit mode, the A frame is taken with the target positioned one-third to one-quarter of the distance along the slit. After the A frame is complete, the telescope moves to place the target approximately the same distance from the other end of the slit. The exposure taken in this configuration is the B frame. It is typical, then, to repeat AB pairs in either an A-B-A-B or A-B-B-A sequence until the total desired integration time is achieved. In this mode, the A frame provides the sky measurement at the target position for the B frame, and the B frame provides the sky for the A frame. This mode is useful as long as the target is compact, relative to the slit length.

In the second mode, nod-off-slit, the A frame is taken with the target at the center of the slit. The B frame is taken with the target completely off the slit, so that the exposure contains only the sky signal. In this mode, the B frame exists only to provide the sky measurement at the target position for the A frame, which may be useful if the target is significantly extended. In this mode, too, either the A-B-A-B or A-B-B-A observing sequence can be used.

For spectroscopy, it is necessary to observe a flat field and a telluric standard star alongside the science observations. Optionally, for either imaging or grism mode, a dark frame may be observed as well. Standards should be taken in flight, near in time to the science observation. Flat fields and dark frames may be taken from the ground, but ideally should be taken the same day as the science observations. The flat field is used to correct for differences in instrumental response across the detector as well as to determine the location of the slit image on the detector. The telluric standard is used to correct for absorption from the Earth's atmosphere, as well as to assign physical flux units to the final science spectrum. The dark frame is used to correct for dark current in the spectroscopic flat frames. It is not required for science processing because science frames are subtracted, which automatically removes the dark current. It is optional for the flat field processing because the dark current for FLITECAM is expected to be small.

3 ALGORITHM DESCRIPTION

3.1 Overview of data reduction steps

This section will describe, in general terms, the major algorithms used by Redux to reduce a FLITECAM observation. See the figures below for flow charts of how these algorithms fit together for spectral and imaging modes.

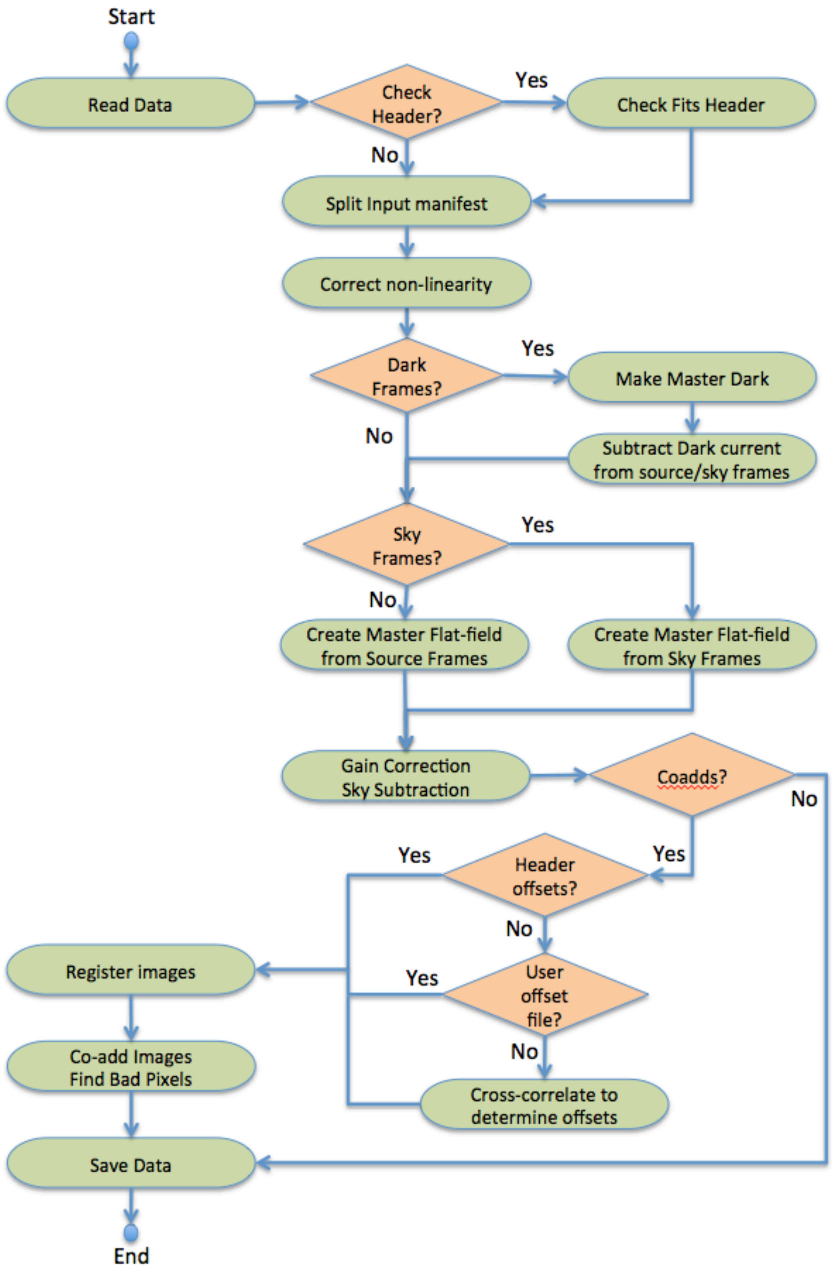


Figure 3: Flowchart for imaging reduction

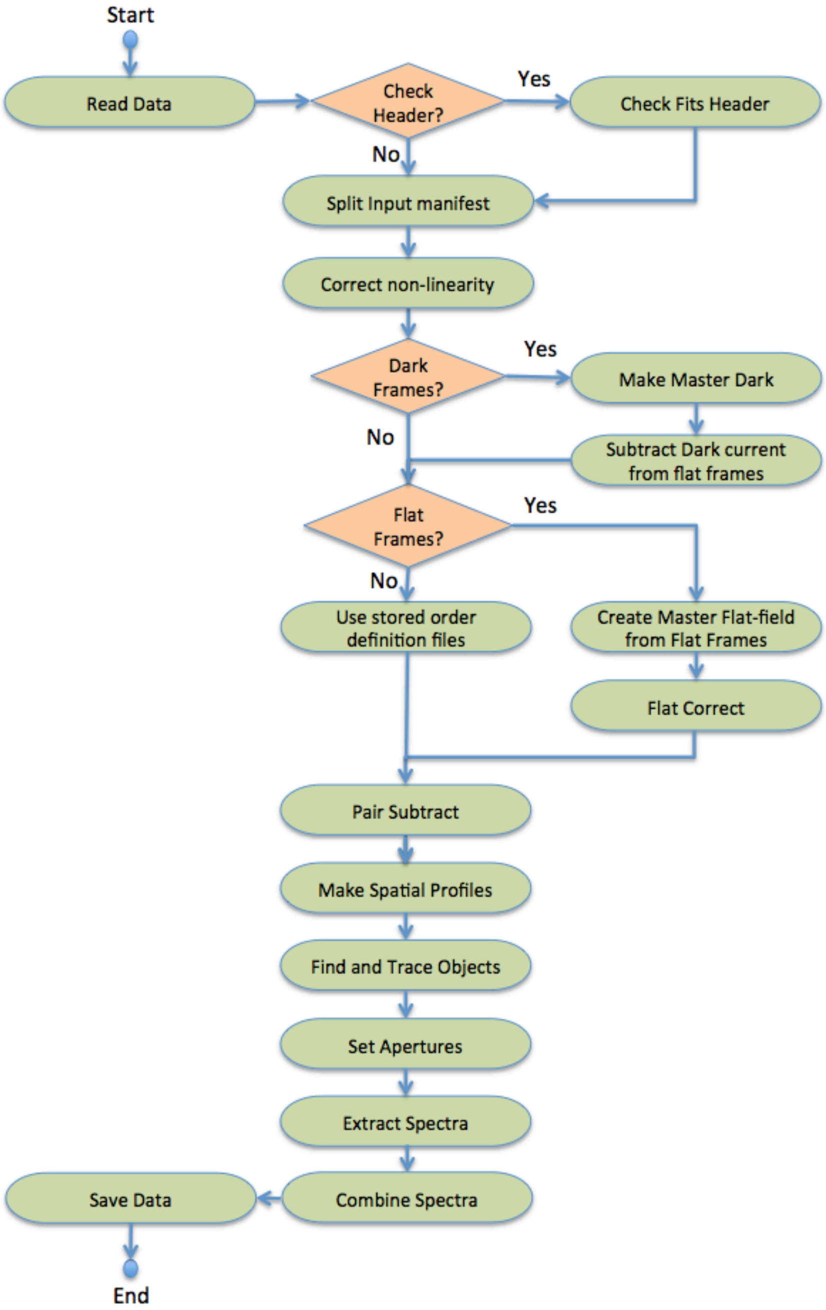


Figure 4: Flowchart for spectroscopy reduction

3.2 Imaging Reduction algorithms

The following subsections detail each of the data reduction pipeline steps outlined in the imaging flowchart above. All algorithms in this section are carried out by the FDRP package.

3.2.1 Nonlinearity correction

In imaging mode, the pipeline reads the FITS header data to sort the input files by observation type (dark, source, or sky, for imaging). Optionally, it can check the FITS header keywords for compliance with requirements at this stage, and abort processing if the requirements are not met. It then reads the data into memory, corrects each frame for nonlinearity, and generates an uncertainty image that associates an error with each pixel in the raw image.

The nonlinearity coefficients for FLITECAM were determined by taking a series of flat exposures with varying exposure times. The count rates at each pixel in the flats were fit with a fourth order polynomial, and the resulting coefficients were stored in a FITS image file as a 3D data cube, where each plane corresponds to a different coefficient in the polynomial fit.

Following the Spextool nonlinearity paper (Vacca et al., 2004; see Other Resources, below), the coefficients are applied to raw data as follows. First, the flat pedestal is determined from the first plane of the linearity coefficients:

$$p_{flat} = C_0 \, \delta t_{flat},$$

where C_0 is the first coefficient plane, and δt_{flat} is the readout time for the flats used to determine the linearity coefficients. The pedestal for the image to be corrected is determined iteratively. The first estimate of the pedestal is:

$$p^{(1)} = \frac{S_{tot}\,\delta t}{n_r n_c \Delta t}\left(\frac{n_r+1}{2} - f\right),$$

where S_{tot} is the raw counts in the image, n_r is the number of readouts, n_c is the number of hardware coadds, δt is the readout time, Δt is the integration time, and f is a fractional value indicating how long it takes for an individual pixel to be readout. Rigorously, f varies for each pixel, depending on its position in the array; for FLITECAM, an average value of $f=0.5$ is used for all pixels. Using this pedestal estimate, the signal for an individual readout is estimated as:

$$s^{(1)} = \frac{S_{tot}}{n_r n_c} + p^{(1)}$$

and both pedestal and signal are corrected for nonlinearity. In order to account for the pedestal value of the flats used to determine the linearity coefficients, the coefficients are normalized by the first coefficient plane, and the polynomial is applied to the value to correct, minus the flat pedestal:

$$p^{(2)} = p^{(1)} \frac{C_0}{C_{nl}(p^{(1)} - p_{flat})}$$

$$s^{(2)} = s^{(1)} \frac{C_0}{C_{nl}(s^{(1)} - p_{flat})},$$

where C_{nl} is the full correction polynomial for each pixel in the image. This process is then repeated once more, replacing S_{tot} with

$${S_{tot}}^{(2)} = n_c n_r s^{(2)} - n_c n_r p^{(2)} \; .$$

The final value for S_{tot}, calculated from $s^{(3)}$ and $p^{(3)}$, is the linearity corrected image.

After linearity correction, the variance is calculated for each pixel from:

$$V = \frac{S_{tot}}{g\, n_r\, n_c^2 \Delta t^2}\left[1 - \frac{\delta t\left(n_r^2 - 1\right)}{3\, \Delta t\, n_r}\right] + \frac{2\, \sigma_r^2}{g^2\, n_r n_c\, \Delta t^2},$$

where g is the electronic gain and σ_r is the read noise for the detector (Vacca et al., 2004). This variance is propagated through all remaining reduction steps and its square root (the uncertainty) is recorded in all output files along with the image, as a separate plane in the data array.

For the imaging pipeline, before proceeding, the linearity-corrected images and the corresponding uncertainty planes are typically clipped to the size of the useful part of the detector (the cyan box in Figure 1).

3.2.2 Dark correction

The imaging pipeline provides, as an optional step, a method to combine dark frames and subtract them from individual science frames. This is intended to correct for any dark current in the detector. However, FLITECAM's dark current has been shown to be negligibly small, and in the course of normal reductions, the median dark current is automatically subtracted off along with the background level in the sky subtraction step anyway. Therefore, dark correction is not usually used.

If dark frames are provided, they are scaled to the exposure time of the science observation and averaged together before being subtracted from the science frames.

3.2.3 Gain correction

As with all modern near-IR detectors, raw images produced by FLITECAM are afflicted with hot and cold pixels, and significant pixel-to-pixel gain variations. In addition, FLITECAM's detector has a large region of low quantum efficiency in the third quadrant and the top of the fourth quadrant of the detector, as shown in Figure 1. These gain variations can be corrected by dividing by a normalized flat field image.

Imaging flats for FLITECAM are made from images of the sky. In the nod-off-array mode, dithered sky images are used to generate the flat. In the stare mode, the dithered source images themselves are used to generate the flat. In either case, the algorithm to generate the flat is the same

First, bad pixels are identified in each frame and recorded in a set of bad pixel masks. Then, all images are combined into a "draft" flat with a high-low rejection algorithm. The draft flat is used to flat-correct all input images, and each image has its median value subtracted off, as a quick background subtraction. These draft images are then used to create object masks that identify any point sources in the frame; the object mask is then added to the bad pixel mask. The raw images are then re-combined, ignoring any pixels identified in the bad pixel/object mask for each frame.

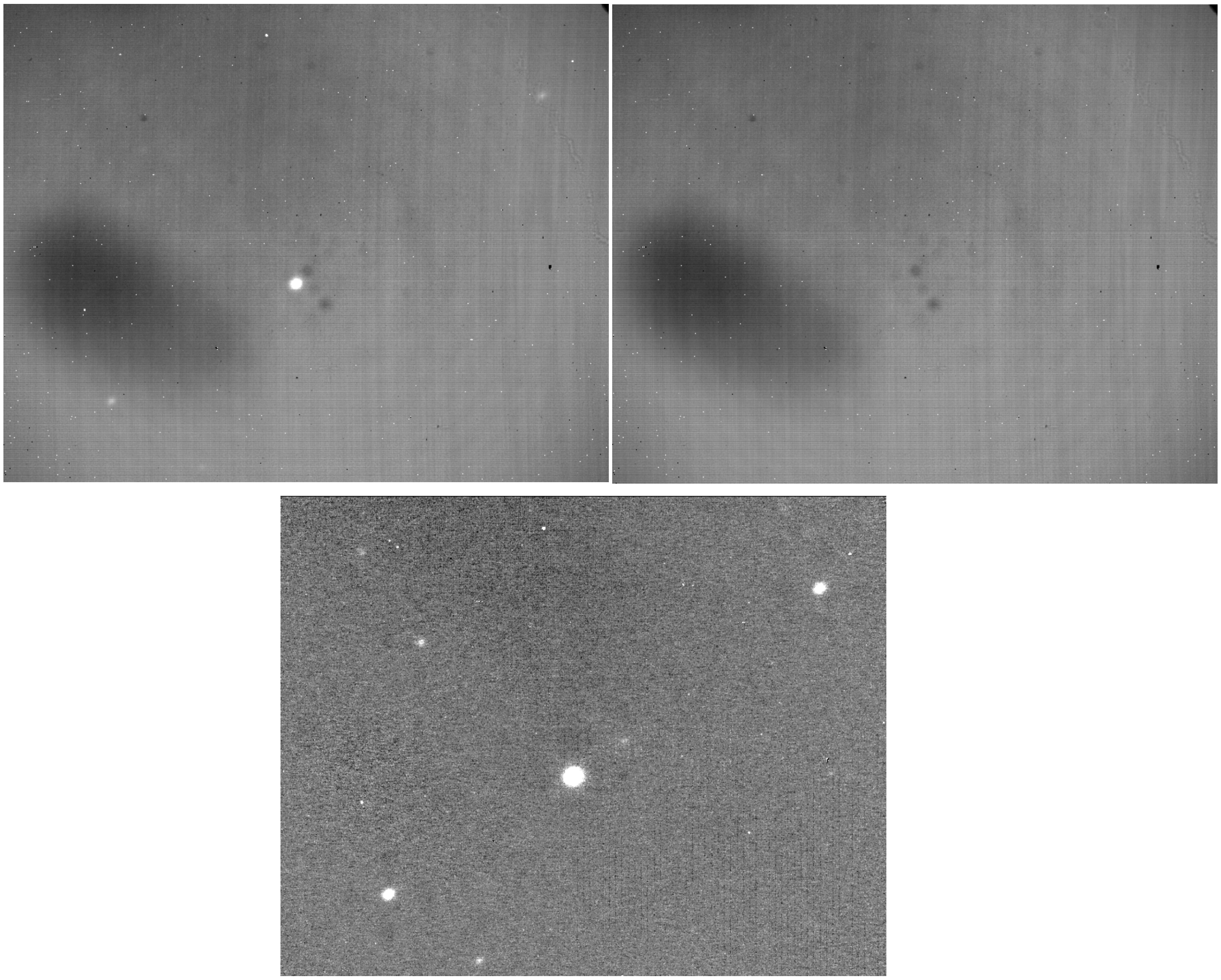


Figure 5: Top left: raw frame at a single dither position, in stare mode. Top right: flat frame made from all dither positions. Bottom: Gain corrected image.

When the final flat frame has been created, it is divided by its median value to normalize it, then it is divided into all input source images to correct for gain variations.

3.2.4 Sky subtraction

The sky background level must then be subtracted for each image. For imaging frames, since the flat is made from the sky images, the average sky level can be found by taking the median of the unnormalized flat. This sky level is subtracted from each source image. Optionally, the median value of each individual image can be subtracted, in place of the median level from the flat.

3.2.5 Image Registration and Coaddition

In order to coadd dithered images, the images must be shifted into a common grid, so that the sources appear at the same location on the array. FDRP has three methods available to determine the offsets required to shift each image to the reference frame of the first image:

- Read the dither parameters from the header of the files. The executed offsets should be recorded in the DITHERX and DITHERY keywords. This method is limited by the accuracy of the values recorded in the header.
- Compute the cross-correlation of the first frame with each subsequent frame, and use the results to calculate the offsets. This method can take a significant amount of time to complete.
- Provide offsets directly, computed in any manner the user chooses, via a text file.

The default option is to use the header keywords if they are available, and the cross-correlation if they are not. It is also possible for the user to specify that registration and coaddition are not desired, and should be skipped.

After the offsets have been determined, they are applied to each image as an integer pixel shift. No interpolation is performed. Then, if desired, all input images are combined with a robust weighted mean algorithm, which helps to mitigate bad pixels in the final output. Optionally, a median combination algorithm may be used instead.

3.3 Spectroscopy Reduction algorithms

The following subsections detail each of the data reduction pipeline steps outlined in the grism flowchart below. All algorithms in this section are carried out by the FSpextool package. This section borrows heavily from the Spextool paper; see the Other Resources section, below, for a reference to this paper.

3.3.1 Flat Field Processing

FSpextool's first step in reducing spectroscopic frames is to prepare a master flat frame. In order to get a high signal-to-noise ratio in this processed flat field, it is typical to take multiple flat-field exposures and combine them together. If there is a significant dark current, it may also be beneficial to take a number of dark frames, combine them together using a median statistic, and subtract the result from the individual flat frames. After dark subtraction, if necessary, FSpextool calculates the median signal of each frame and scales them all to a common level (the median of the individual median levels). It then combines all the flats, typically using a median statistic.

FSpextool then uses this master flat to determine the edges of the spectral order on the detector. It draws an approximate center position for the order from a configuration file on disk. Starting from this position, it uses an edge detection algorithm to find the top and bottom of the order at regular intervals across the dispersion direction. The edge points are then fit with a low-order polynomial to give the edge coefficients of the order. These coefficients are later used in the source extraction process to reconstruct the location of the order at any given column.

Finally, the flat field is normalized to unity so that when it is divided into the source spectrum, it does not affect the signal-to-noise estimate for the spectrum. In order to correct for the spectral shape of the flat source, the normalization begins by fitting a low-order two-dimensional surface to a straightened, resampled flat image. This fit is then divided into the unresampled master flat to normalize it.

FSpextool also calculates the variance of the master flat frame. The variances of the input flat and dark frames depend on the RMS read noise of the detector, the detector gain, the integration time per coadd, and the number of counts in each pixel. The variance for the master flat is the combination of the variances of all the input frames, propagated through dark subtraction, scaling, median-combination, and the normalization process.

For some observations, there may not be appropriate flat fields available. In this case, FSpextool provides a default flat for each grism mode that defines the edges of the spectral order, but should not be divided into the data. This will at least allow processing to continue through extraction.

3.3.2 Image Processing

As for the FLITECAM imaging mode, FSpextool first corrects the input images for detector nonlinearity and creates a variance plane, using the algorithm described above. Then, the pipeline does A-B pair subtraction of all the input images. It then converts the units in the image from raw counts to counts per second by dividing by the integration time per coadd (Δt). It also divides by the normalized flat, if available. The resulting signal in the 2-D spectrum is:

$$\mathrm{S_{AB}} = \frac{\mathrm{S_A - S_B}}{\Delta \mathrm{t} \cdot \mathrm{flat}}$$

where S_A is the raw counts in frame A, S_B is the raw counts in frame B, and *flat* is the normalized flat image.

Alongside the image processing, the individual variances for the A frame, B frame, and flat are combined as follows to get the total variance for the 2-D spectrum:

$$\mathrm{V_{AB}} = \frac{\mathrm{V_A + V_B}}{\Delta \mathrm{t}^2 \cdot \mathrm{flat}^2} + \frac{\mathrm{V_{flat}} \cdot \mathrm{S_{AB}^2}}{\mathrm{flat}^2}$$

where V_A is the variance of frame A, V_B is the variance of frame B, and V_{flat} is the variance of the normalized flat image.

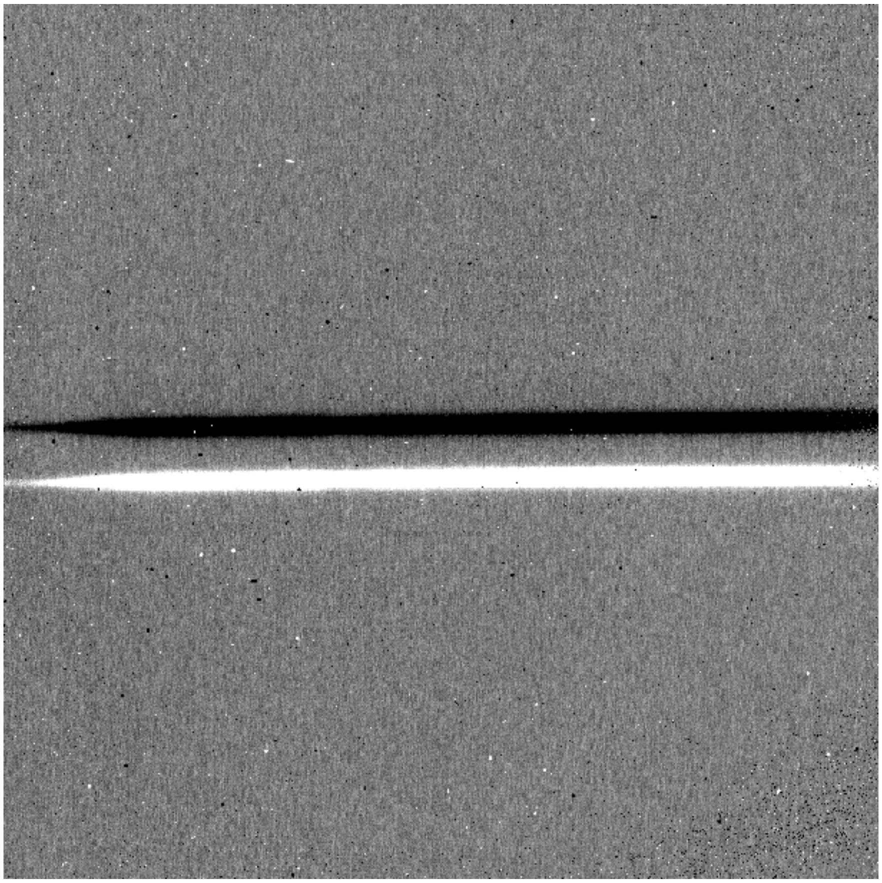


Figure 6: Pair-subtracted image in nod-on-slit mode, rotated to align the dispersion axis with the x-axis.

3.3.3 Spatial Profile Construction

In order to aid in spectral extraction, FSpextool constructs a smoothed model of the relative intensity of the target spectrum at each spatial position, for each wavelength. This spatial profile is used to compute the weights in optimal extraction or to fix bad pixels in standard extraction (see below). Also, the user must refer to the median profile, collapsed along the wavelength axis, to define the extraction parameters. Since FSpextool rotates input data so that dispersion is along the x-axis, we will here refer to a wavelength point as a column, and a spatial point as a row.

To construct the spatial profile, FSpextool first subtracts the median signal from each column in a spectral order to remove the residual background. It then resamples the order onto a regular grid of wavelength columns and spatial rows; this has the effect of removing any curvature of the order across the detector. The intensity in this resampled image in column i and row s is given by

$$\mathrm{O_{i,s} = f_i P_{i,s}}$$

where f_i is the total intensity of the spectrum at wavelength i, and $P_{i,s}$ is the spatial profile at column i and row s. To get the spatial profile $P_{i,s}$, we must approximate the intensity f_i. To do so, FSpextool computes a median of the columns of the order image to get a first-order approximation of the median spatial profile P_s. Assuming that

$$\mathrm{O_{i,s} \approx c_i P_s},$$

FSpextool uses a linear least-squares algorithm to fit P_s to $O_{i,s}$ and thereby determine the coefficients c_i. These coefficients are then used as the first-order approximation to f_i: the resampled order image $O_{i,s}$ is divided by f_i to derive $P_{i,s}$. FSpextool then fits a low-order polynomial along the columns at each spatial point s in order to smooth the profile and thereby increase its signal-to-noise. The coefficients of these fits can then be used to determine the value of $P_{i,s}$ at any column i and spatial point s (see Figure 7, left). The median of $P_{i,s}$ along the wavelength axis generates the median spatial profile, P_s (see Figure 7, right). Finally, $P_{i,s}$ is resampled back onto the pixel grid (i,j) to derive a profile $P_{i,j}$ that matches the coordinates of the raw image.

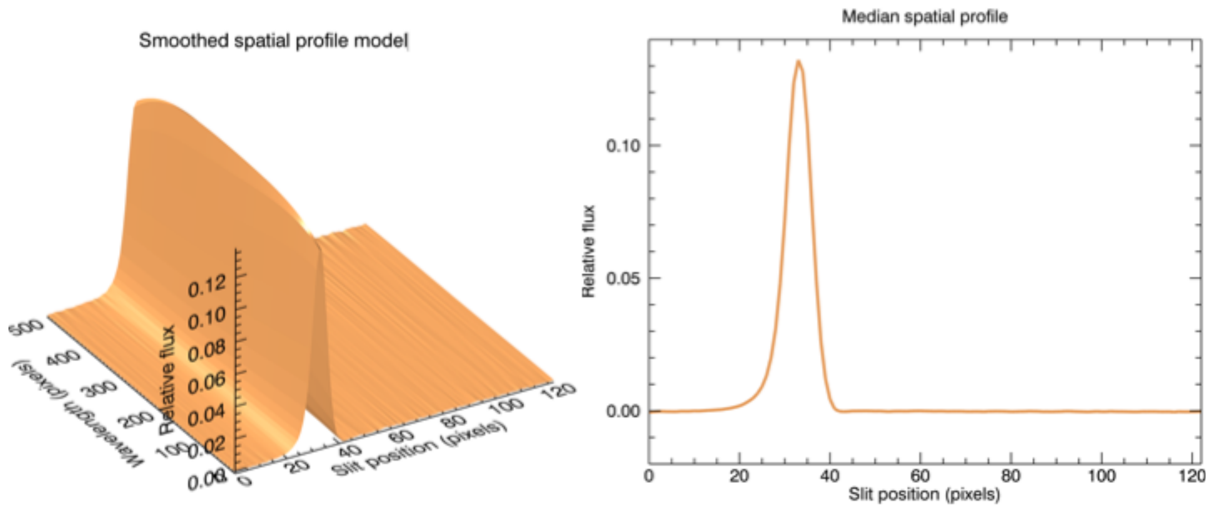


Figure 7: Spatial profile image as a function of wavelength and slit position (left), and median spatial profile as a function of slit position (right)

3.3.4 Aperture Location and Tracing

The median spatial profile can now be used to identify extraction apertures for the target. The aperture centers can be identified automatically by iteratively finding local maxima in the absolute value of the spatial profile, or can be specified directly by the user.

The true position of the aperture center will, however, vary somewhat with wavelength, as a result of small optical effects or atmospheric dispersion. To account for this variation, FSpextool traces the spectrum across the array. For a point source, it fits a Gaussian in the spatial direction, centered at the specified position, at regular intervals in wavelength. The centers of these fits are themselves fitted with a low-order polynomial; the coefficients of these fits give the trace coefficients that identify the center of the spectral aperture at each wavelength. For an extended source, the continuum cannot generally be directly traced. Instead, FSpextool uses the edges of the order to find the pixel position of the user-specified aperture center at each wavelength, and fits these positions with a low-order polynomial to get the trace coefficients.

Besides the aperture centers, FSpextool also requires that the user specify a few more aperture parameters. For point sources, optimal extraction requires a PSF radius, corresponding to the distance from the center at which the flux from the source falls to zero. For all extractions (point source or extended source, optimal or standard), FSpextool requires an aperture radius, corresponding to the edge of the extraction aperture. It is also possible to specify a background region outside of any extraction apertures, for fitting and removing the residual sky signal.

3.3.5 Spectral Extraction

Before extracting a spectrum, FSpextool first uses the identified background regions to find the residual sky background level. For each column, it fits a low-order polynomial to the values in the specified regions, as a function of slit position. This polynomial determines the wavelength-dependent sky level ($R_{i,j}$) to be subtracted from the spectrum ($D_{i,j}$).

FSpextool offers two methods for extracting a spectrum from the sky-subtracted image: optimal and standard extraction. The standard extraction method uses values from the spatial profile image ($P_{i,j}$) to replace bad pixels and outliers, then sums the flux from all pixels contained within the aperture radius. The flux and variance at column i is then:

$$\mathrm{f}_{\mathrm{i}}^{\mathrm{sum}} = \sum_{\mathrm{j}=\mathrm{j}_1}^{\mathrm{j}_2} \epsilon_{\mathrm{i,j}} \left(\mathrm{D}_{\mathrm{i,j}} \text{-} \mathrm{R}_{\mathrm{i,j}}\right)$$

$$\mathrm{V}_{\mathrm{i}}^{\mathrm{sum}} = \sum_{\mathrm{j}=\mathrm{j}_1}^{\mathrm{j}_2} \epsilon_{\mathrm{i,j}}{}^2 \left(\mathrm{V}_{\mathrm{D}_{\mathrm{i,j}}} + \mathrm{V}_{\mathrm{R}_{\mathrm{i,j}}}\right)$$

where j_1 and j_2 are the upper and lower limits of the extraction aperture (in pixels), and $\varepsilon_{i,j}$ is the fraction of the pixel (i, j) that is contained within the extraction aperture. This extraction method is the only algorithm available for extended sources.

Point sources may occasionally benefit from using standard extraction, but optimal extraction generally produces higher signal-to-noise ratios for these targets. This method works by weighting each pixel in the extraction aperture by how much of the target's flux it contains. FSpextool first normalizes the spatial profile by the sum of the spatial profile within the PSF radius defined by the user:

$$\mathrm{P}_{\mathrm{i,j}}^{'} = \frac{\mathrm{P}_{\mathrm{i,j}}}{\sum_{\mathrm{k}_1}^{\mathrm{k}_2} \mathrm{P}_{\mathrm{i,k}}}$$

where k_1 and k_2 are the upper and lower limits given by the PSF radius. $P'_{i,j}$ now represents the fraction of the total flux from the target that is contained within pixel (i,j), so that $(D_{i,j} - R_{i,j}) / P'_{i,j}$ is a set of j independent estimates of the total flux at column i. FSpextool does a weighted average of these estimates, where the weight depends on the pixel's variance ($V_{Dij} + V_{Rij}$), as well as the normalized spatial profile ($P'_{i,j}$). Then the flux and variance at column i is:

$$\mathrm{f}_{\mathrm{i}}^{\mathrm{opt}} = \frac{\sum_{\mathrm{j}_1}^{\mathrm{j}_2} \mathrm{M}_{\mathrm{i,j}} \mathrm{P}_{\mathrm{i,j}} (\mathrm{D}_{\mathrm{i,j}} \text{-} \widetilde{\mathrm{R}}_{\mathrm{i,j}}) / (\mathrm{V}_{\mathrm{D}_{\mathrm{i,j}}} + \mathrm{V}_{\widetilde{\mathrm{R}}_{\mathrm{i,j}}})}{\sum_{\mathrm{j}_1}^{\mathrm{j}_2} \mathrm{M}_{\mathrm{i,j}} \mathrm{P}_{\mathrm{i,j}}{}^2 / (\mathrm{V}_{\mathrm{D}_{\mathrm{i,j}}} + \mathrm{V}_{\widetilde{\mathrm{R}}_{\mathrm{i,j}}})}$$

$$\mathrm{V}_{\mathrm{i}}^{\mathrm{opt}} = \frac{1}{\sum_{\mathrm{j}_1}^{\mathrm{j}_2} \mathrm{M}_{\mathrm{i,j}} \mathrm{P}_{\mathrm{i,j}}{}^2 / (\mathrm{V}_{\mathrm{D}_{\mathrm{i,j}}} + \mathrm{V}_{\widetilde{\mathrm{R}}_{\mathrm{i,j}}})}$$

where $\mathrm{M}_{\mathrm{i,j}}$ is a bad pixel mask and j_1 and j_2 are the upper and lower limits given by the aperture radius. Note that bad pixels are simply ignored, and outliers will have little effect on the average because of the weighting scheme.

After extraction, spectra from separate apertures or separate images may be combined together to increase the signal-to-noise of the final product. The default combination statistic is a robust weighted mean.

3.3.6 Wavelength Calibration

FSpextool can extract spectra from images where the dispersion axis is well aligned with the image rows. In this case, each column represents a single wavelength bin. The wavelength calibration is one-dimensional, and can be derived from an extracted spectrum of an arc lamp or sky lines. FSpextool can also extract spectra from images where the dispersion axis does not align with the image rows: that is, there may be a significant curvature or tilt to the spectral lines with respect to the columns. In this case, the wavelength calibration must be two-dimensional, and can be derived from a two-dimensional spectral image of an arc or sky lines. FSpextool provides an interactive GUI, called *xwavecal2d*, which allows the user to identify spectral lines at regular intervals across the image, fit a surface to the wavelength and spatial data, and produce a map of the wavelength and spatial coordinates at each pixel.

Two-dimensional wavelength calibration requires an extra step in the extraction process, in order to ensure that each pixel contains only data from a single wavelength bin. FSpextool accomplishes this by first interpolating the x and y image coordinates onto a regularized grid of wavelength and spatial coordinates, as read from the wavelength calibration map. For each wavelength bin in this grid, it steps through the spatial elements, adding up the flux of any pixel or partial pixel contained within the given spatial and wavelength bin (the shaded area in the figure below). The resultant sums construct the flux profile with respect to the spatial coordinate, analogous to a slice along a column in the case of one-dimensional wavelength calibration. This profile is then used to perform either standard or optimal extraction as detailed above.

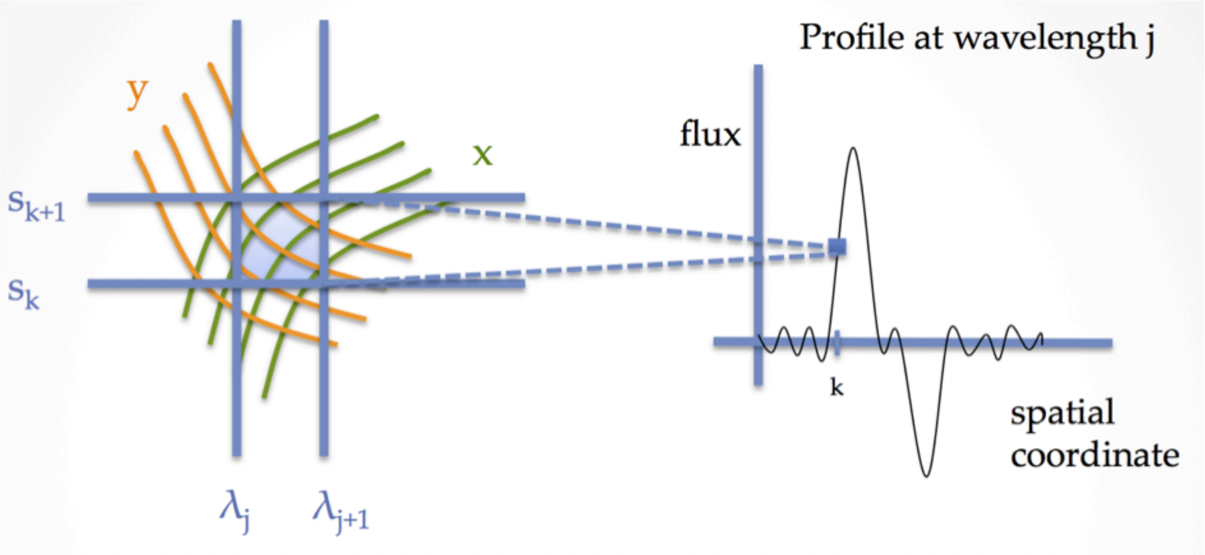


Figure 8: Extraction with two-dimensional wavelength calibration. The flux for the pixel at spatial element k and wavelength j is the weighted sum of the flux from pixels contained, or partially contained, within the grid element.

Commissioning data appears to show that FLITECAM spectral lines have significant curvature or tilt with respect to array columns, particularly for the narrow slit. Therefore, two-dimensional wavelength calibration maps are required to correctly extract and calibrate the spectra. However, it is not expected that wavelength calibration will change significantly between observations. Therefore, FSpextool provides a default wavelength calibration map for each grism mode that can be used to extract FLITECAM data. The *xwavecal2d* GUI can be used to produce new maps as needed.

The default wavelength calibration is expected to be good to within one pixel of dispersion, provided that the grisms have not shifted since the wavecal maps were made. As a final step, the pipeline offers the user the chance to identify spectral features in the extracted spectrum. Using these features, the pipeline calculates a zero-point shift to apply to the wavelengths after extraction. This step can correct for any minor shifts in the wavelength calibration.

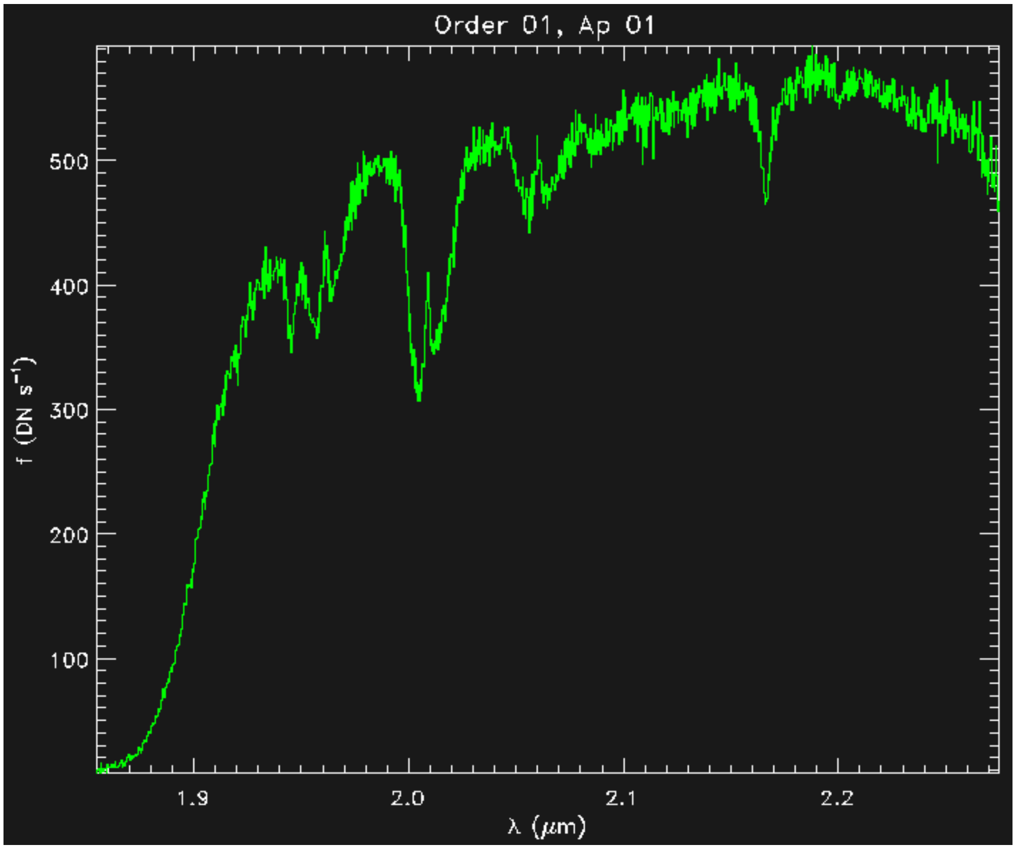


Figure 9: Final extracted, calibrated spectrum.

3.3.7 Mosaicking spectra

It may sometimes be useful to stitch together separate spectra to view the entire wavelength coverage of an observation in a single plot. FSpextool provides a tool called *xmergeorders* to perform this task. This tool can, for example, combine extracted spectra from several different grism modes, using any overlapping regions to scale the spectra to each other.

3.4 Other Resources

For more information on the instrument itself, see the FLITECAM paper:

FLITECAM: a 1-5 micron camera and spectrometer for SOFIA, Ian S. McLean, et al. (2006, SPIE 6269E, 168).

or the SOFIA observer's handbook:

Sofia Observer's Handbook for Cycle 2: v2.1.2, SOFIA User Support Group (2013, SOFIA Science Center).

For more information on how to run the FSpextool interactive tools (*xspextool, ximgtool*, *xvspec*, *xwavecal2d, xcombspec*, *xtellcor_general*, and *xcleanspec*), see the help files distributed with the FSpextool code, under *fspextool/Helpfiles*.

For more information on the reduction algorithms used in FSpextool, see the Spextool papers:

Spextool: A Spectral Extraction Package for SpeX, a 0.8-5.5 Micron Cross-Dispersed Spectrograph, Michael C. Cushing, William D. Vacca and John T. Rayner (2004, PASP 116, 362).

A Method of Correcting Near-Infrared Spectra for Telluric Absorption, William D. Vacca, Michael C. Cushing and John T. Rayner (2003, PASP 115, 389).

Nonlinearity Corrections and Statistical Uncertainties Associated with Near-Infrared Arrays, William D. Vacca, Michael C. Cushing and John T. Rayner (2004, PASP 116, 352).

For more information on the IDL Astronomy User's Library, used in both the imaging and grism pipelines, see:

The IDL Astronomy User's Library, W.B. Landsman (1995, ASPC 77, 437).

4 FLUX CALIBRATION

4.1 Imaging

The calibration of Level 2 imaging data is carried out in several steps that in practice are performed iteratively. The first step consists of measuring the photometry of all the standard stars for a specific mission. Calibration factors are then derived from the measured photometry and the known fluxes of the standards. For each object, the calibration factors from all the standards on a flight are adjusted to account for the differences between the target airmass and altitude and those of the standards, and then averaged. The pipeline then inserts this value and its uncertainty into the headers of the Level 2 data files. After all calibration factors are derived for a mission, the final step requires studying the calibration values. The calibration factor for each instrument configuration should be consistent within a mission and even between consecutive missions. Values that are not consistent may come from bad observations of a standard star. Bad standard stars are removed, and the reduction process is repeated.

The determination of the calibration factors begins with the analysis of the reduced Level 2 images of the standard stars observed on a given flight. Photometry is done on these images using an aperture of 12 pixels and a background region of 25-35 pixels. The measured count rate is then divided by the predicted flux in Jy for each star in each filter. The predicted fluxes were computed by multiplying the model stellar spectrum by estimates for the overall filter+instrument+telescope+atmosphere response curve and integrating over the filter passband to compute the mean flux in the band. The adopted filter throughput curves are those provided by the vendor. The instrument throughput is calculated by multiplying an estimate of the instrumental optics transmission (0.80) and the detector quantum efficiency (0.56). The FLITECAM overall throughput is (0.285). The telescope throughput value is assumed to be constant (0.85) across the entire FLITECAM wavelength range. The atmospheric transmission is computed using the ATRAN code (Lord 1992) for a range of observatory altitudes (corresponding to a range of overhead precipitable water vapor values) and telescope elevations. The equations used for determining the mean wavelength and the mean flux of the standard at the mean wavelength are given in Tokunaga & Vacca (2005). The equations for deriving the flux for a flat spectrum source are the same as those given in Herter et al. (2013) for FORCAST.

Photometric standards for FLITECAM have been chosen from three sources: (1) bright stars with spectral classifications of A0V as listed in SIMBAD; (2) Landolt SA stars (K giants and A0-4 main sequence stars) listed as 'supertemplate' stars in Cohen et al. (2003); K giant stars listed as 'spectral template' stars in Cohen et al. (1999). For all of these objects, models are either available (from the Cohen papers) or derivable (from a model of Vega for the A0V). Use of the A0V stars requires scaling the Vega model to the observed magnitudes of the target and reddening the model to match the observed color excess of the target. It should be noted that A0V stars should be used to calibrate the Pa alpha filter, as these objects have a strong absorption feature in this band. The models of the spectral template K giants listed in Cohen et al. (1999) extend down only to 1.2 microns, and therefore cannot be used to calibrate the J band filter.

Using the measured photometry of the standard, N_e^{std}(in counts/s), and the predicted mean fluxes of the standards in each filter, $\langle F_\nu^{std}\rangle$ (in Jy), the flux of a target object is

$$F_\nu^{nom,obj}(\lambda_{ref}) = \frac{N_e^{obj}}{C}$$

where $N_e^{obj}$is the count rate in counts/s detected from the source, C is the calibration factor (counts/s/Jy), and $F_\nu^{nom,obj}(\lambda_{ref})$ is the flux in Jy of a nominal, “flat spectrum” source (for which $F_\nu \sim \nu^{-1}$) at a reference wavelength λ_{ref}. The calibration factor, C, is computed from

$$C = \frac{N_e^{obj}}{F_\nu^{nom,obj}(\lambda_{ref})} = \frac{N_e^{std}}{\langle F_\nu^{std}\rangle}\frac{\lambda_{piv}^2}{\langle\lambda\rangle\lambda_{ref}}\frac{R_\lambda^{obj}/R_\lambda^{ref}}{R_\lambda^{std}/R_\lambda^{ref}}$$

with an uncertainty given by

$$\left(\frac{\sigma_C}{C}\right)^2 = \left(\frac{\sigma_{N_e^{std}}}{N_e^{std}}\right)^2 + \left(\frac{\sigma_{\langle F_\nu^{std}\rangle}}{\langle F_\nu^{std}\rangle}\right)^2$$

Here, $\lambda_{piv}$is the pivot wavelength of the filter, $\langle\lambda\rangle$ is the mean wavelength of the filter, and the ratio $R_\lambda/R_\lambda^{ref}$ accounts for differences in system response (transmission) between the actual observations and those for a ‘reference’ altitude of 41K and a telescope elevation of 45°.

The values of C, σ_C , and λ_{ref} are written into the headers of the Level 3 data as the keywords CALFCTR, ERRCALF, and LAMREF, respectively. The reference wavelength λ_{ref} for these observations was taken to be the mean wavelengths of the filters, $\langle\lambda\rangle$. Note that σ_C currently assumes no uncertainty on the stellar models and the values of $\langle F_\nu^{std}\rangle$. Typical uncertainties on the MID stellar model fluxes are expected to be on the order of 5-10% (see Herter et al. 2013) based on our previous experience with FORCAST at SOFIA. We expect these uncertainties to be lower than 5% in NIR.

Each observation of a standard provided a value of the calibration factor in the various filters. The values of C were examined across all of the Cycle 1 flights to check for consistency. Discrepant values signaled problems with the standard star data and those images were then excluded from the calibration process.

An observer often wishes to determine the true flux of an object at the reference wavelength, $F_\nu^{obj}(\lambda_{ref})$, rather than the flux of an equivalent nominal, "flat spectrum" source. To do this, we define a color correction K such that

$$K = \frac{F_\nu^{nom,obj}(\lambda_{ref})}{F_\nu^{obj}(\lambda_{ref})},$$

where $F_\nu^{nom,obj}(\lambda_{ref})$ is the flux density one obtained by measurement on a data product. Divide the measured values by K to obtain the "true" flux density. In terms of the wavelengths defined above,

$$K = \frac{\langle\lambda\rangle\lambda_{ref}}{\lambda_{piv}^2}\frac{\langle F_\nu^{obj}\rangle}{F_\nu^{obj}(\lambda_{ref})}.$$

We give values for K for power-law and blackbody spectral shapes in the Tables below. For most filters and spectral shapes, the color corrections are small (<10%).

FLITECAM										
Filter	J	H	Hwide	K	Klong	Kwide	L	Lprime	LplusM	M
$\langle\lambda\rangle$	1.242	1.632	1.793	2.105	2.485	2.308	3.535	3.848	4.087	4.844
$\lambda(piv)$	1.239	1.63	1.787	2.103	2.482	2.297	3.531	3.844	4.01	4.841

FLITECAM								
Filter	NbL	NbM	Ice	Pa	PaCont	Pa+H	PaC+H	PAH
$\langle\lambda\rangle$	3.603	4.807	3.05	1.874	1.9	1.874	1.9	3.301
$\lambda(piv)$	3.602	4.806	3.05	1.874	1.9	1.874	1.9	3.301

FLITECAM+HIPO										
Filter	J	H	Hwide	K	Klong	Kwide	L	Lprime	LplusM	M
$\langle\lambda\rangle$	1.246	1.632	1.794	2.106	2.486	2.31	3.535	3.849	4.092	4.844
$\lambda(piv)$	1.244	1.63	1.788	2.103	2.482	2.299	3.531	3.845	4.015	4.841

FLITECAM+HIPO								
Filter	NbL	NbM	Ice	Pa	PaCont	Pa+H	PaC+H	PAH
$\langle\lambda\rangle$	3.603	4.807	3.05	1.874	1.9	1.874	1.9	3.301
$\lambda(piv)$	3.602	4.806	3.05	1.874	1.9	1.874	1.9	3.301

Table 1: Filter Wavelengths, Cycle 2

Filter	α						
$\langle\lambda\rangle$	-3	-2	-1	0	1	2	3
1.251	1.004	1.000	1.000	1.004	1.012	1.036	1.040
1.642	1.002	1.000	1.000	1.002	1.007	1.022	1.024
1.802	1.007	1.000	1.000	1.007	1.022	1.066	1.073
2.120	1.002	1.000	1.000	1.002	1.007	1.021	1.024
2.500	1.003	1.000	1.000	1.003	1.008	1.023	1.026
2.325	1.009	1.000	1.000	1.009	1.028	1.084	1.095
3.555	1.002	1.000	1.000	1.002	1.006	1.019	1.021
3.867	1.002	1.000	1.000	1.002	1.005	1.016	1.018
4.101	1.037	1.000	1.000	1.039	1.122	1.403	1.457
4.873	1.001	1.000	1.000	1.001	1.003	1.009	1.010
3.607	1.000	1.000	1.000	1.000	1.001	1.002	1.002
4.807	1.000	1.000	1.000	1.000	1.000	1.001	1.001
3.052	1.002	1.000	1.000	1.000	1.001	1.002	1.002
1.874	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.903	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.874	1.000	1.000	1.000	1.000	1.000	1.000	1.000
1.903	1.000	1.000	1.000	1.000	1.000	1.000	1.000
3.301	1.000	1.000	1.000	1.000	1.000	1.000	1.001

$\langle\lambda\rangle$/ T_{BB}(K)	50000	20000	10000	5000	3000	1000	750	500	300	200
1.251	1.022	1.019	1.013	1.003	0.993	1.071	1.196	1.662	4.194	17.077
1.642	1.014	1.012	1.010	1.005	0.999	1.010	1.047	1.183	1.789	3.674
1.802	1.041	1.037	1.031	1.017	1.000	1.014	1.093	1.419	3.195	11.652
2.120	1.014	1.012	1.011	1.010	1.002	0.998	1.013	1.080	1.380	2.202
2.500	1.015	1.014	1.012	1.029	1.004	0.994	1.003	1.048	1.273	1.920
2.325	1.053	1.050	1.043	1.009	1.011	0.985	1.025	1.215	2.214	6.165
3.555	1.012	1.011	1.011	1.008	1.006	0.996	0.995	1.005	1.068	1.239
3.867	1.011	1.010	1.009	1.183	1.006	0.997	0.996	1.001	1.045	1.165
4.101	1.250	1.239	1.221	1.005	1.133	0.946	0.917	0.990	1.696	4.609
4.873	1.006	1.006	1.005	1.001	1.004	0.999	0.998	0.998	1.009	1.045
3.607	1.001	1.001	1.001	1.000	1.001	0.999	0.999	1.000	1.008	1.026
4.807	1.000	1.000	1.000	1.001	1.000	1.000	1.000	1.000	1.001	1.004
3.052	1.001	1.001	1.001	3000	1.001	0.999	1.000	1.002	1.013	1.041
1.874	20000	10000	3000	0.993	1.000	1.000	1.000	1.001	1.002	1.006
1.903	1.019	1.013	0.993	0.999	1.000	1.000	1.000	1.001	1.003	1.007
1.874	1.012	1.010	0.999	1.000	1.000	1.000	1.000	1.001	1.002	1.006
1.903	1.037	1.031	1.000	1.002	1.000	1.000	1.000	1.001	1.003	1.007
3.301	1.012	1.011	1.002	1.004	1.000	1.000	1.000	1.000	1.003	1.009

Table 2: Color Corrections, Cycle 2. Note: for power law spectral shapes, $F_\nu \sim \nu^\alpha$

4.2 Spectroscopy

FLITECAM grism data will be telluric corrected and flux calibrated in the same manner as is done for ground-based NIR spectroscopy. This entails observing a “telluric standard” (usually an A0V star) whose spectrum can be modeled. A model of Vega is smoothed to the appropriate resolution, adjusted for extinction, broadened to account for rotation, and scaled to match the observed mag of the target star. The ratio between this model and the observed spectrum then constitutes (to first order) the correction spectrum for both telluric absorption and instrumental response. Additional corrections will have to be made to account for the possibly different widths of the H lines between the model and the actual star observed; without these adjustments, residual artifacts arising from the H lines will be present in the ratio. Those corrections can be made semi-automatically in some cases, manually in others. The final correction curve can then be applied to the science object spectrum to flux calibrate it. The *xtellcor* software, incorporated in the FSpextool package, provides a means of deriving the correction/calibration curve and applying it to the science object's spectrum.

An additional difficulty, avoided in ground-based observations whenever possible, arises from the fact that the telluric standard and the science object will almost certainly be observed at different airmasses. Variations in the telluric absorption (relative to that at 41K feet) as a function of wavelength for the complete range of possible airmasses (zenith angles between 30 and 70) will be pre-computed and stored. These additional corrections will be applied during the application of the telluric correction/calibration curve. Similar corrections are applied during the Level 3 processing of images.

5 DATA PRODUCTS

5.1.1 Filenames

FLITECAM output files from Redux are named according to the convention:
FILENAME = F[flight]_FC_IMA|GRI_AOR-ID_SPECTEL1SPECTEL2_Type_FN1[-FN2].fits, where flight is the SOFIA flight number, FC is the instrument identifier, IMA or GRI specifies that it is an imaging or grism file, AOR-ID is the 8 digit AOR identifier for the observation, SPECTEL1 and SPECTEL2 are the keywords specifying the filter, grism, or slit used, Type is three letters identifying the product type (listed in the table below), and FN1 is the file number corresponding to the input file. FN1-FN2 is used if there are multiple input files for a single output file, where FN1 is the file number of the first input file and FN2 is the file number of the last input file.

5.1.2 Pipeline Products

The following table lists all intermediate products generated by Redux for FLITECAM, in the order in which they are produced. The product type is stored in the FITS headers under the keyword PRODTYPE. By default, for imaging, only the final coadded product is saved. For spectroscopy, the image, flat, rectified image, and final combined spectrum are saved by default. Specifying the appropriate option in either the automatic or interactive modes will save all products.

Mode	Description	PRODTYPE	ID	Saved by default?
Imaging	Non-linearity corrected	lincor	LIN	N
Imaging	Clipped to usable portion of detector	clipped	CLP	N
Imaging	Bad pixel masks	bpmsk	BPM	N
Imaging	Initial (draft) flat field	flat_draft	DFL	N
Imaging	Initial (draft) flat fielded images	ffimg_draft	DFF	N
Imaging	Flat-fielded with DFL and sky subtracted	ffsub_draft	DFS	N
Imaging	Object plus bad pixel masks	obmsk	OBM	N
Imaging	Final flat/sky image	sky	SKY	N
Imaging	Final normalized flat field	flat	FLT	N
Imaging	Flat fielded images	ffimg	FFI	N
Imaging	Flat-fielded and sky subtracted	ffbsub	FBS	N
Imaging	Registered images	imreg	RIM	N
Imaging	Final coadded image	coadd	COA	Y
Grism	Linearized, flat-corrected, pair-subtracted 2D spectrum	image	IMG	Y
Grism	Normalized flat field	flat	FLT	Y
Grism	Rectified image (produced during extraction)	rectimg	RIM	Y
Grism	Extracted spectrum, containing all apertures	spec	SPC	N
Grism	Combined spectrum from all images	combspec	CMB	Y

Table 3: Final and intermediate data products

5.1.3 Data Format

All files produced by the pipeline are FITS single-extension image files. All image products (for both imaging and spectroscopy modes) are 3-D arrays of data, where the first plane is the image frame and the second plane represents the uncertainty associated with each pixel in the image. For imaging products, the uncertainty plane is the 1-sigma uncertainty; for grism images, the plane contains the variance (the square of the 1-sigma uncertainty). The final coadded image for the imaging pipeline and the rectified image for the spectroscopy pipeline also have a bad pixel mask appended in a third plane. The fourth plane in the coadded image is an exposure map, indicating the number of exposures at each pixel; multiply this frame by the value of the keyword EXPTIME to get the total integration time at each pixel. The extracted spectral products are one-dimensional spectra, stored in three rows of data. The first row is the wavelength, the second is the flux, and the third is the error (standard deviation). If there were multiple apertures selected (e.g. for the nod-on-slit mode), then the spectrum for each aperture is stacked into a different plane. The length of the row varies depending on the data, but is typically around 1000 pixels.

6 GROUPING LEVEL_1 DATA FOR PROCESSING

For FLITECAM's imaging mode, there are three kinds of input data: darks, sky frames, and sources. Darks are always optional. The sky frames and source frames should share the same exposure time and number of hardware coadds, as well as instrument configuration and filters. In order to be calibrated together, they must also be taken at similar altitudes and zenith angles. It may also be necessary to require that they are taken on the same flight leg, or with the same line-of-sight rewind.

For the grism mode, there are darks, flats, and sources. Darks are optional, but if used, they should match the exposure time and number of coadds of the flats. Flat frames are recommended, but may be replaced with default flats if appropriate frames were not taken with the science. If provided, flat frames should match the filter, grism, and slit of the source observation. All source observations must share the same target, filter, grism, slit, exposure time, and coadds, and must be taken at similar altitudes and zenith angles.

These grouping requirements translate into a set of FITS keywords that must match in order for a set of data to be grouped together. These relationships are summarized in the figures and tables below.

Keyword	Datatype	Match Criterion
OBSTYPE	STR	Exact
OBJECT	STR	Exact
INSTCFG	STR	Exact
SPECTEL1	STR	Exact
EXPTIME	FLT	Exact
COADDS	INT	Exact
MISSN-ID	STR	Exact
ALTI_STA	FLT	+/- 500
ALTI_END	FLT	+/- 500
ZA_START	FLT	+/- 2.5
ZA_END	FLT	+/- 2.5
PLANID	STR	Exact
AOR_ID (optional)	STR	Exact
LASTREW (optional)	STR (DATE/TIME)	Exact

Table 4: Grouping criteria for FLITECAM imaging

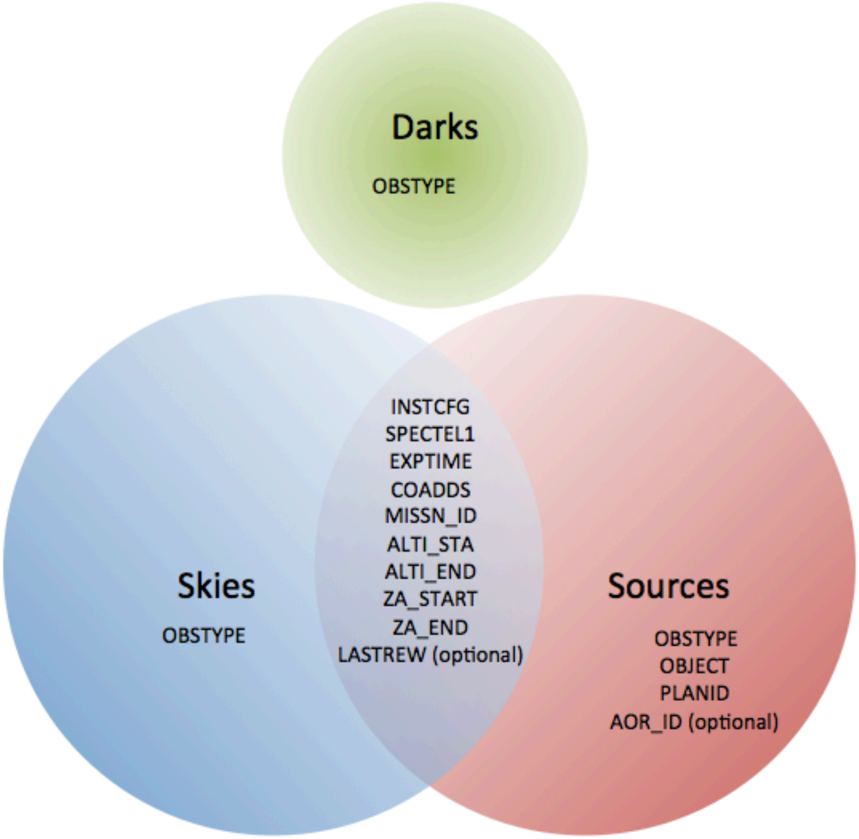


Figure 10: Grouping keyword requirements for FLITECAM imaging mode

Keyword	Datatype	Match Criterion
OBSTYPE	STR	Exact
OBJECT	STR	Exact
INSTCFG	STR	Exact
SPECTEL1	STR	Exact
SPECTEL2**	STR	Exact
EXPTIME	FLT	Exact
COADDS	INT	Exact
MISSN-ID	STR	Exact
ALTI_STA	FLT	+/- 500
ALTI_END	FLT	+/- 500
ZA_START	FLT	+/- 2.5
ZA_END	FLT	+/- 2.5
PLANID	STR	Exact
AOR_ID (optional)	STR	Exact
LASTREW (optional)	STR (DATE/TIME)	Exact

Table 5: Grouping criteria for FLITECAM grism. **If SPECTEL2 is in use (value!='NONE' or 'UNKNOWN'), the observation should always be included in the grism plan, regardless of INSTCFG.

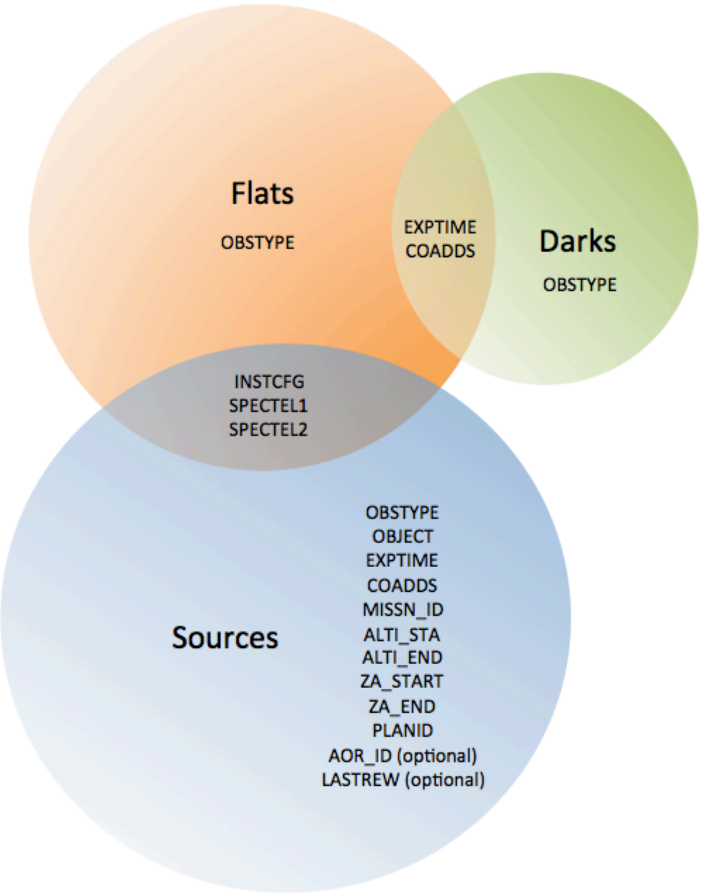


Figure 11: Grouping keyword requirements for FLITECAM grism mode

7 CONFIGURATION AND EXECUTION

7.1 Installation

Redux is a software package written in IDL that is designed to be a framework for executing any number or combination of data reduction algorithms. For FLITECAM, it has been developed to support seamlessly running image processing algorithms from the FDRP package to reduce imaging data, alongside spectral extraction algorithms from the FSpextool package for spectroscopy data. FDRP is an IDL package developed specifically for FLITECAM imaging, while FSpextool was originally developed for use with the SpeX instrument (as Spextool), then was adapted for FLITECAM. Redux can run in an automatic batch mode, integrated with the SOFIA Data Pipeline System (DPS), or it can run with a graphical front end as a quick-look data viewer in flight or during manual data reduction and analysis. Redux with FDRP and FSpextool was developed under Linux and MacOS X operating systems. Other operating systems may also work, but have not been tested.

Running Redux requires IDL 8.1 or later, as well as the latest version of the IDL Astronomy User's Library, the Coyote graphics library, the FDRP package, the FSpextool package, and the Redux code. FDRP, FSpextool, and Redux are under SOFIA DPS revision control and can be obtained directly from git repositories there. The IDL Astronomy User's Library (astrolib) is publicly available, and can be downloaded from the website at the following URL:
http://idlastro.gsfc.nasa.gov/homepage.html.
The Coyote graphics library (coyote) is also publicly available and can be downloaded from:
http://www.idlcoyote.com/documents/programs.php.
When these packages have been installed, their locations should be added to the IDL_PATH environment variable, so that their procedures are accessible to Redux.

SOFIA may distribute the Redux, FDRP, and FSpextool codes as gzipped tar files. If so, unpack them, as, for example:

```
tar xvzf redux.tar.gz
tar xvzf fdrp.tar.gz
tar xvzf fspextool.tar.gz
```

This will create directories called *redux*, *fdrp*, and *fspextool*, which will contain a number of subdirectories. Each of these package directories should be added to the IDL_PATH as well.

7.2 Configuration

For FSpextool algorithms, default options are specified in a configuration file called *FLITECAM.dat*, located in the *fspextool* package directory. This file contains keyword-value pairs, in the format *parameter=value*. The parameters must all be present and in the correct order, but can have any number of spaces or comments between them. Comment lines begin with the % or # character. See Appendix A for a sample of this configuration file.

In automatic pipeline mode, parameters set in the configuration file are the values actually used by the pipeline, unless overridden by an input parameter file. In interactive mode, the configuration files set the default values, but the parameter values used can be modified at run-time.

FDRP does not have a configuration file for parameter definition.

7.3 Input data

Redux takes as input raw FLITECAM FITS data files containing 1024x1024 pixel image arrays. The FITS headers contain data acquisition and observation parameters and, combined with the pipeline configuration files, comprise the information necessary to complete all steps of the data reduction process. Some critical keywords are required to be present in the raw data in order to perform a successful grouping, reduction, and ingestion into the SOFIA archive. These keywords are listed in Appendix B.

It is assumed that the input data have been successfully grouped before beginning reduction: Redux considers all input files in a reduction to be part of a single homogeneous reduction group, to be reduced together with the same parameters.

7.4 Automatic mode execution

Redux is an object-oriented program whose basic unit is a reduction object (*flitecam_imaging_reduction__define.pro* or *flitecam_grism_reduction__define.pro*). To run the pipeline from the IDL command line as a DCS black box pipeline, we run the pipeline wrapper (*redux_pipe.pro*). This wrapper takes as input the path to an input manifest file. This text file should contain a line specifying the number of input files, then the relative path to each input file, one per line. The script then reads these input files, instantiates the appropriate reduction object according to the mode specified in the input FITS headers, then calls the object's reduce method. This method calls each processing step in order, as appropriate for the given mode. Finally, the wrapper script will write an output manifest called *outfiles.txt* containing the names of the produced data files.

This wrapper can be invoked from the IDL prompt, as

```
IDL> redux_pipe, ‘infiles.txt’
```

or directly from a terminal as

```
localhost$ echo “redux_pipe, ‘infiles.txt’” | idl
```

The wrapper accepts a single input parameter on the command line, which allows the user to give it a parameter file that specifies any desired reduction parameters. This option is given as, for example:

```
IDL> redux_pipe,‘infiles.txt’,PARAM_FILE=‘param.json’
```

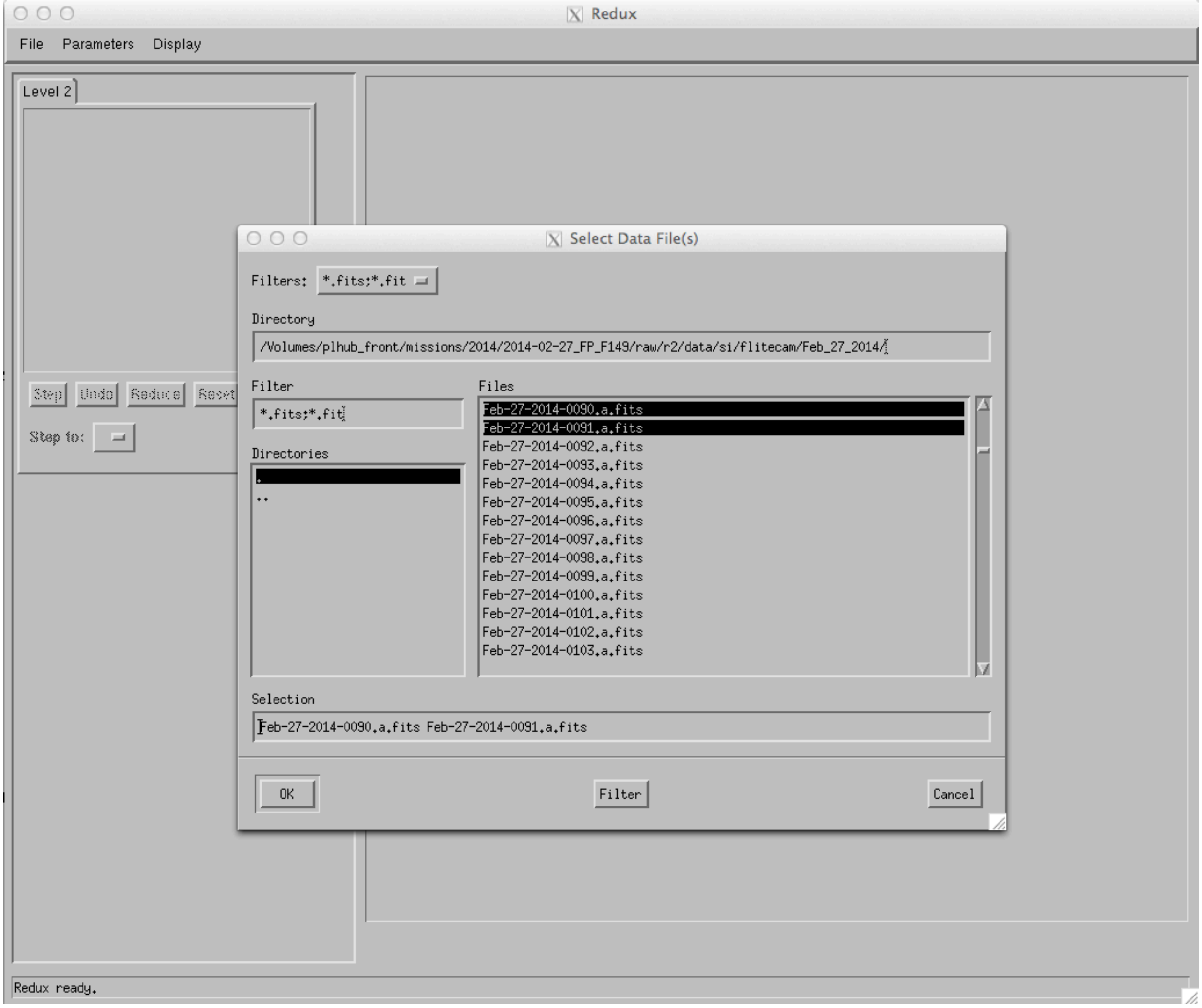
where *param.json* is a file that lists parameter keywords and values in JSON format for any step in the pipeline reduction. Parameter files can be generated interactively with the GUI (see below), then saved and fed to the automatic pipeline for batch reduction of a large number of files.

7.5 Manual mode execution

It is also possible to run the pipeline interactively, using a graphical user interface. The IDL command `redux`, called without arguments, will launch the Redux GUI.[1]

7.5.1 Basic workflow

To start an interactive reduction, open a set of FLITECAM files, using the File menu (**File->Open New Reduction**). This will bring up a file dialog window (see figure, below). All files selected will be reduced together as a single reduction set. If flats or darks are available for the reduction, they should be selected along with the source files.



Filters: *.fits;*.fit

Directory
/Volumes/plhub_front/missions/2014/2014-02-27_FP_F149/raw/r2/data/si/flitecam/Feb_27_2014/

Filter
.fits;.fit

Directories
.
..

Files
```
Feb-27-2014-0090.a.fits
Feb-27-2014-0091.a.fits
Feb-27-2014-0092.a.fits
Feb-27-2014-0093.a.fits
Feb-27-2014-0094.a.fits
Feb-27-2014-0095.a.fits
Feb-27-2014-0096.a.fits
Feb-27-2014-0097.a.fits
Feb-27-2014-0098.a.fits
Feb-27-2014-0099.a.fits
Feb-27-2014-0100.a.fits
Feb-27-2014-0101.a.fits
Feb-27-2014-0102.a.fits
Feb-27-2014-0103.a.fits
```

Selection
Feb-27-2014-0090.a.fits Feb-27-2014-0091.a.fits

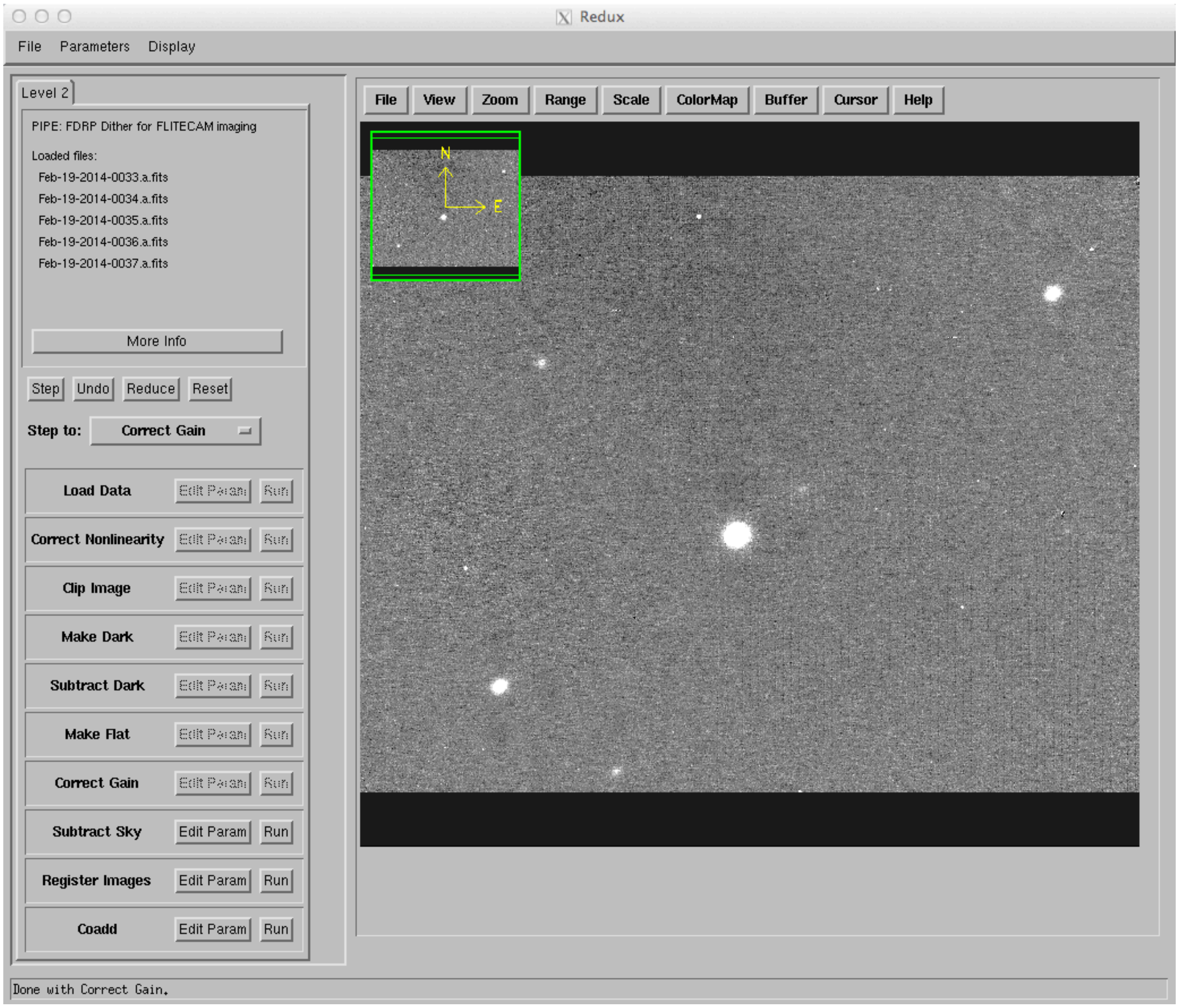
Redux ready.

Figure 12: Open New Reduction

---

[1] The *redux* GUI has a single, rarely-used command-line parameter: if called with the parameter /*SMALL*, it will produce an initial window that is sized for small laptop screens.

Redux will decide the appropriate reduction steps from the input files, and load them into the GUI, as in the figures below.



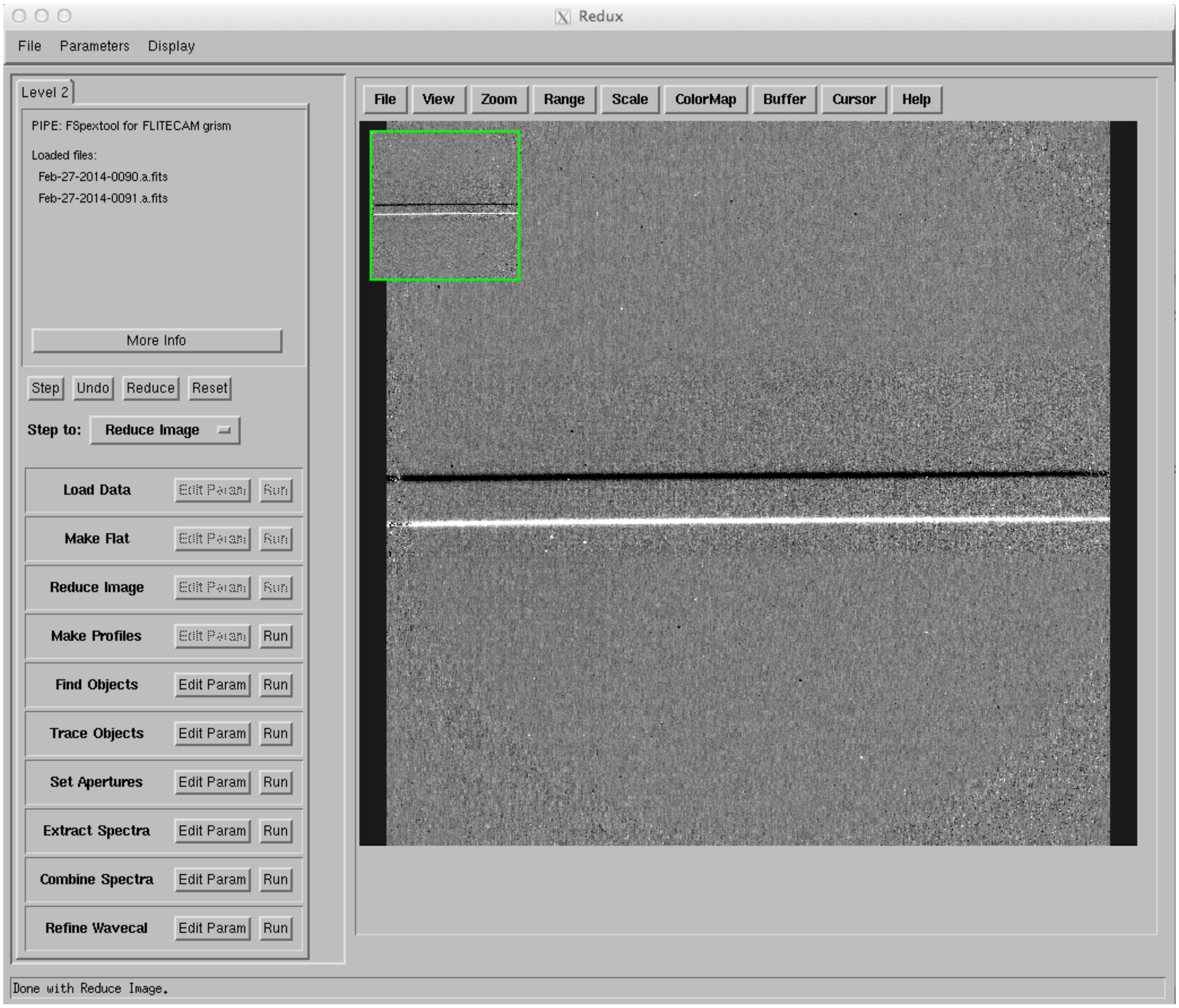
PIPE: FDRP Dither for FLITECAM imaging

Loaded files:
Feb-19-2014-0033.a.fits
Feb-19-2014-0034.a.fits
Feb-19-2014-0035.a.fits
Feb-19-2014-0036.a.fits
Feb-19-2014-0037.a.fits

Step to: Correct Gain

```
Done with Correct Gain.
```

Figure 13: Sample reduction in imaging mode.



PIPE: FSpextool for FLITECAM grism

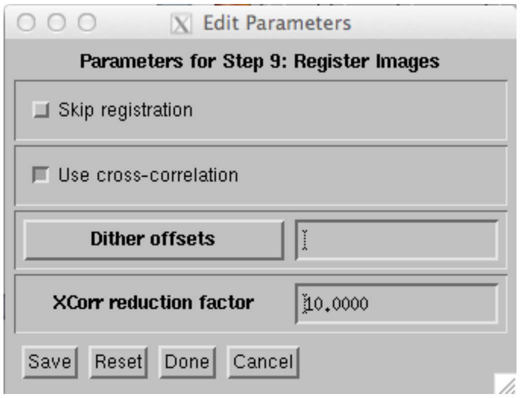
Loaded files:
Feb-27-2014-0090.a.fits
Feb-27-2014-0091.a.fits

Done with Reduce Image.

Figure 14: Sample reduction in grism mode.

Each reduction step has a number of parameters that can be edited before running the step. To examine or edit these parameters, click the **Edit Param** button next to the step name to bring up the parameter editor for that step. Within the parameter editor, all values may be edited, but will not be used unless **Save** or **Done** is selected. Clicking **Save** will leave the parameter editor window open; clicking **Done** will save values and close the window. Clicking **Reset** will restore any edited values to their defaults; clicking **Cancel** will discard all changes to the parameters.

The current set of parameters can be displayed, saved to a file, or reset all at once using the **Parameters** menu. A previously saved set of parameters can also be restored for use with the current reduction (**Parameters->Load Parameters**). Note that edited parameters will retain their values for future reductions unless they are manually reset.



Parameters for Step 9: Register Images

☐ Skip registration

☑ Use cross-correlation

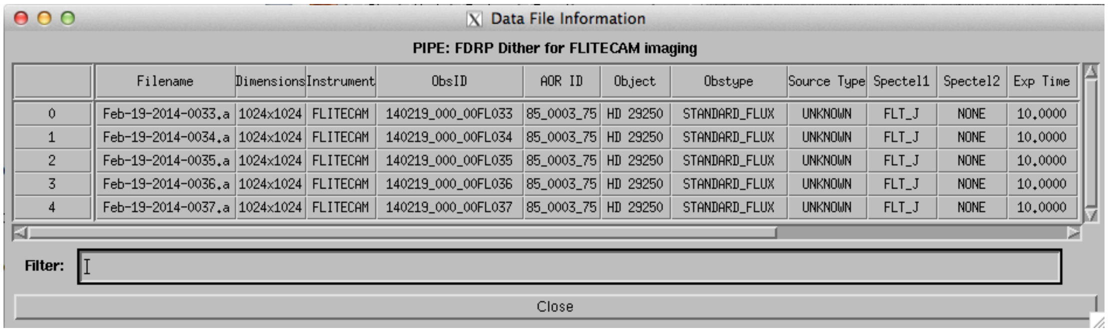
Dither offsets:

XCorr reduction factor: 10.0000

Figure 15: Sample parameter editor (for Register Images step)

After all parameters for a step have been examined and set to the user's satisfaction, a processing step can be run on all loaded files either by clicking **Step**, or the **Run** button next to the step name. Each processing step must be run in order, but if a processing step is selected in the **Step to:** widget, then clicking **Step** will treat all steps up through the selected step as a single step. When a step has been completed, its buttons will be grayed out and inaccessible. It is possible to undo one previous step by clicking **Undo**. All remaining steps can be run at once by clicking **Reduce**. After each step, the results of the processing will be displayed in the display window. Clicking **Reset** will restore the reduction to the initial state, without resetting parameter values.

Files can be added to the reduction step (**File->Add Files**) or removed from the reduction set (**File->Remove Files**), but either action will reset the reduction for all loaded files. Selecting **Display->Display File Information**, or the **More Info** button, will pull up a table of information about the currently loaded files (Figure 17). The table rows displayed can be filtered by entering a search string into the **Filter** text box.



PIPE: FDRP Dither for FLITECAM imaging

	Filename	Dimensions	Instrument	ObsID	AOR ID	Object	Obstype	Source Type	Spectel1	Spectel2	Exp Time
0	Feb-19-2014-0033.a	1024x1024	FLITECAM	140219_000_00FL033	85_0003_75	HD 29250	STANDARD_FLUX	UNKNOWN	FLT_J	NONE	10.0000
1	Feb-19-2014-0034.a	1024x1024	FLITECAM	140219_000_00FL034	85_0003_75	HD 29250	STANDARD_FLUX	UNKNOWN	FLT_J	NONE	10.0000
2	Feb-19-2014-0035.a	1024x1024	FLITECAM	140219_000_00FL035	85_0003_75	HD 29250	STANDARD_FLUX	UNKNOWN	FLT_J	NONE	10.0000
3	Feb-19-2014-0036.a	1024x1024	FLITECAM	140219_000_00FL036	85_0003_75	HD 29250	STANDARD_FLUX	UNKNOWN	FLT_J	NONE	10.0000
4	Feb-19-2014-0037.a	1024x1024	FLITECAM	140219_000_00FL037	85_0003_75	HD 29250	STANDARD_FLUX	UNKNOWN	FLT_J	NONE	10.0000

Figure 16: File information table

7.5.2 Display features

Redux displays images using ximgtool, a full-featured display tool distributed with FSpextool. For more information, see the ximgtool help file, available from Redux via the **Help** button just above the display. See below for a quick listing of the most useful ximgtool features.

Feature	Menu button	Keyboard shortcut
Load new file	**File->Load FITS**	--
Load file into new frame	**File->New Frame**	--
View FITS header	**File->View Header**	--
Zoom	**Zoom->Zoom In**, **Zoom Out**, **Zoom To Fit**	Press **z** to enter zoom mode, then **i** to zoom in, **o** to zoom out, or **t** to zoom to fit
Color stretch	**Cursor->Stretch**	Press **s** to enter stretch mode, click and drag to change brightness and contrast
Set display range	**Cursor->Range**	Press **r** to enter range mode, click and drag to select the box that sets the display range
Display distance	--	Press **d** to enter distance mode, then click and drag to identify start and end points
Line cut	--	Press **l** to enter line cut mode, then click and drag to identify start and end points
Display image statistics	--	Press **m** to enter moments mode, then click and drag to identify box for which the statistics should be calculated
Photometry	--	Press **a** over a star to do basic photometry.
Clear current mode	--	Press **c**
Buffer select	**Buffer->Buffer 1, Buffer 2…**	Press **f** to move to the next buffer, **b** to move to the previous buffer.
Buffer math	**Buffer->Buffer Math**, then select buffers and arithmetic operation to perform	--
Blink buffers	**Buffer->Blink Buffers**	--
Cycle frames	**Buffer->Cycle frames**	Press **n** to move to the next available frame, **p** to move to the previous frame

Table 6: Useful ximgtool features

Ximgtool has five buffers available for simultaneous display of images. If there are more than five files loaded into Redux, they can only be viewed by selecting **Display->Quick Look** from the Redux menu. This will cycle through the data in its current processing state, allowing interaction and analysis with each image in turn. To move between images, click the **Next File** or **Previous File** buttons, below the image. Click **Cancel** to quit the quick look display.

In spectroscopy mode, median spatial profiles generated from the 2D spectra are displayed in an independent plot window. When the aperture parameters are set, they will be overplot on the spatial profile in this window.

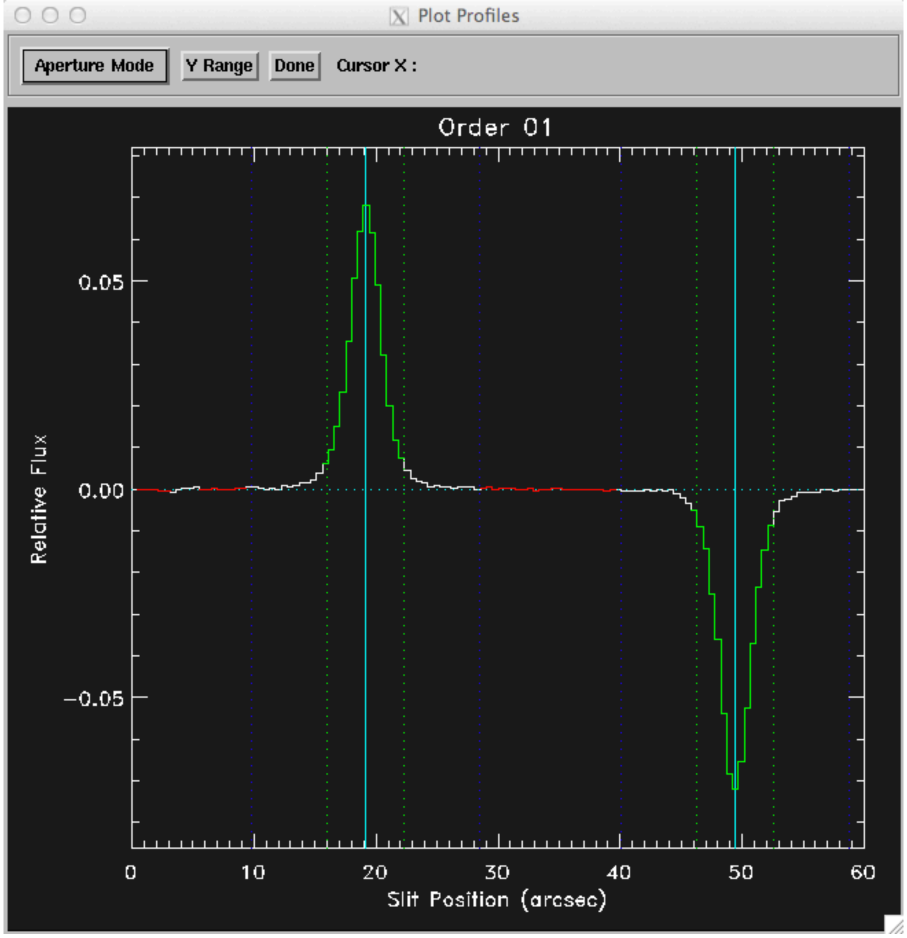


Figure 17: Aperture locations identified and overplotted on the spatial profile for a nod-on-slit observation. Aperture centers are identified with light blue lines, aperture radius with green lines, PSF radius with dark blue lines, and background regions with red lines.

Extracted spectra are displayed using xvspec, a display tool packaged with FSpextool. Xvspec typically displays only one spectrum at a time, but each loaded spectrum can be examined by using the **Quick Look** feature, or else additional spectra may be directly loaded for display by clicking **Load FSpextool FITS** or **File->Add Spectra**. For more information on xvspec features, use the Help button in the xvspec window.

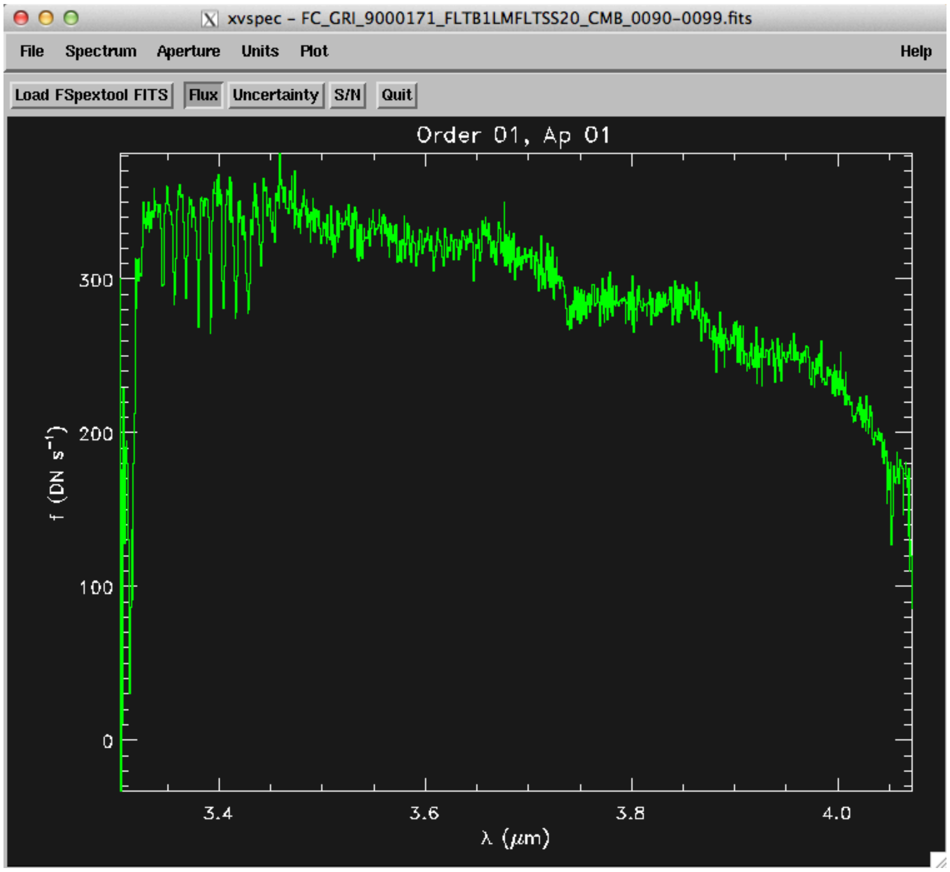


Figure 18: Final extracted spectrum, displayed in xvspec.

7.6 FLITECAM Reduction

FLITECAM data reduction with Redux follows the data reduction flowcharts given in section 3. At each step, Redux attempts to determine automatically the correct action, given the input data and default parameters, but each step can be customized as needed. Some key parameters to note are listed in the following subsections.

7.6.1 Imaging Parameters

- **Load Data:**
    - *Save all intermediate files*: Check this box to save a file after each reduction step.
    - *Check headers*: By default, Redux will abort reduction if the input header keywords do not meet requirements. Uncheck this box to attempt the reduction anyway.
- **Correct Nonlinearity:**
    - *Skip linearity correction*: If checked, Redux will skip applying the linearity coefficients to the raw data. This option should only be used for testing.
    - *Linearity coefficients*: If desired, a file containing a non-default set of linearity coefficients may be selected. This option will also generally be useful only for testing.
- **Clip Image:**

    - *Use header data section*: Check to use a data section defined in the ARRAY0 keyword in the header of the first file in the reduction set. The keyword should contain the x-size and y-size of the data section to use. For example, ARRAY0='100,200' will clip the data to a 100x200 pixel section centered on the center of the array.
    - *Data section (x)*: Enter a pixel range to use as the data section range in the x-direction. Values should be entered as *x1-x2*, e.g. 170-930 will use the pixels x=170 to x=930.
    - *Data section (y)*: Enter a pixel range to use as the data section range in the y-direction. Values should be entered as *y1-y2*, e.g. 200-800 will use the pixels x=200 to x=800.
- **Make Flat:**
    - *PSF FWHM*: Set to the expected FWHM of point sources to mask in the input flat frames.
- **Subtract Sky:**
    - *Subtract median sky*: If checked, the background will be determined from the median of each frame, rather than from the flat/sky frame.
- **Register Images:**
    - *Skip registration*: If checked, the pipeline will not attempt to shift the input images into a common reference frame.
    - *Use cross-correlation*: If checked, the pipeline will use cross-correlation as the method for determining registration offsets. If unchecked, the default is to use header keywords if found, and cross-correlation if not found.
    - *Dither offsets*: Select a file containing registration offsets. This should be a text file with three columns: the file number, the x-offset, and the y-offset for each input file. If provided, these offsets will override any other registration method.
    - *XCorr reduction factor*: Set to 10, 50, or 100. Larger values start with a coarser grid to determine the initial guess at the offset shifts, but will then require more iterations to complete the calculation. Starting with a larger grid may make the registration algorithm complete faster. Default value is 10.
- **Coadd:**
    - *Skip coadd*: If checked, input files will not be coadded. Bad pixel masks will be computed and attached to the background-subtracted images and written as the final output.

7.6.2 Spectroscopy Parameters

- **Load Data:**
    - *Save all intermediate files*: Check this box to save a file after each reduction step.
    - *Check headers*: By default, Redux will abort reduction if the input header keywords do not meet requirements. Uncheck this box to attempt the reduction anyway.
    - *Wavecal file*: Enter a wavelength calibration file to use in place of the default file. This option will generally be useful only for testing.
- **Reduce Image:**
    - *Pair subtract*: If checked, files will be pair-subtracted in the order they are given.

  - *Divide by flat*: If checked, and a flat is loaded along with the sources, the input images will be divided by the processed flat. If the default flat is being used, it will not be used to flat-correct the data, regardless of the value of this parameter.
  - *Coadd images*: If checked, all input images will be coadded, after pair subtraction (if desired). This option may be useful for extracting spectra from faint sources.
  - *Combination statistic*: Select the algorithm to use to coadd images.
- **Find Objects:**
  - *Number of apertures*: Enter the number of apertures to try to find automatically. The default is to look for two apertures for nod-on-slit mode, and one otherwise. If the observation has SRCTYPE=EXTENDED_SOURCE, the default is to fix the aperture to the center of the slit.
  - *Guess*: Enter a guess value for the aperture to use as a starting point. Values are in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons. For example, *3,8;2,7* will look for apertures near 3” and 8” in the first image and 2” and 7” in the second image. If there are multiple files loaded, but only one aperture list is given, the aperture parameters will be used for all images.
  - *Fix*: Enter a value to use as the aperture center. The format of this parameter is the same as for the *guess* parameter, but no fit will be done to attempt to refine the position: it will be used as entered.
  - *Exclude orders*: Enter comma-separated numbers to identify specific orders to exclude from spectral extraction. This option is not used for FLITECAM, as it has only a single order.
- **Trace Objects:**
  - *Fit trace?*: Select the algorithm to use to trace the aperture center across the array. Auto will use the SRCTYPE keyword to decide: if SRCTYPE=EXTENDED_SOURCE, it will fix the trace to the aperture center. Otherwise, it will fit Gaussians to the continuum to determine the trace. Select *Fix* or *Fit* to override the automatic choice.
- **Set Apertures:**
  - *Auto*: If checked, the pipeline will use the header parameters to determine how to set the apertures automatically. If SRCTYPE=EXTENDED _SOURCE, it will perform a full-slit extraction. Otherwise, it will automatically determine a PSF radius, aperture radius, and background regions to use in optimal extraction. Uncheck *auto* to use either of the next two parameters.
  - *Subtract background*: If checked, and *auto* is not checked, then the pipeline will attempt to determine suitable background regions from the median spatial profile.
  - *Set PSF radius for optimal extraction*: If checked, and *auto* is not checked, then the pipeline will attempt to determine a PSF radius from the median spatial profile. If a PSF radius is not set at this step, standard extraction will be used as the extraction method.
  - *Override* parameters: If numbers are entered for any of the following parameters, they will be used as the aperture values. No automatic fit will be done for the specified parameter. For each one, values should be given in arcseconds across the slit (refer to the spatial profile). Separate multiple apertures by commas; separate values for multiple files by semi-colons.

        - *Override aperture radius*: Enter a value to use as the aperture radius.
        - *Override PSF radius*: Enter a value to use as the PSF radius. This number should be **larger** than the aperture radius.
        - *Override background regions*: Enter a range to use as the background region. For example, *0-1,8-10* will use the regions between 0" and 1" and between 8" and 10" to determine the background level to subtract in extraction.
    - *Override aperture signs*: enter either 1 or -1 to override the automatic determination of the aperture sign from the spatial profile. If the value is -1, the spectrum will be multiplied by -1. Separate multiple apertures by commas; separate values for multiple files by semi-colons.
- **Extract Spectra:**
    - *Extraction algorithm*: If set to *auto*, the pipeline will use standard extraction for SRCTYPE=EXTENDED_SOURCE and optimal extraction otherwise. To override this, select either *optimal* or *standard*.
    - *Use median profile*: By default, the pipeline uses a wavelength-dependent spatial profile for extraction, but this method may occasionally give poor results, if the signal-to-noise in the profile is low. Check this option to use the median spatial profile across all wavelengths instead.
    - *Background fit order*: Set to a number greater than or equal to zero for the polynomial order of the fit to the background regions. The default is zero for FLITECAM.
    - *Bad pixel threshold:* Enter a value for the threshold for a pixel to be considered a bad pixel. This value is multiplied by the standard deviation of all good pixels in the aperture at each wavelength bin.
- **Combine Spectra:**
    - *Combine apertures*: If unchecked, spectra from separate files will be combined, but separate apertures will remain separate in the output file.
    - *Scale to median*: If checked, each spectrum will be scaled to the median across all spectra before combination.
    - *Correct spectral shape*: If checked, each spectrum's shape will be corrected to the shape of the first spectrum before combination.
    - *Combination statistic*: Select the combination method. The default is a robust weighted mean.
- **Refine Wavecal:**
    - *Select feature interactively?*: If unchecked, a default zero-point shift will be looked up in a configuration table. If checked, a GUI window will pop up, allowing selection and editing of any number of spectral feature. To select the feature, click on the spectrum, then in the new window, type *f* and click on either side of the desired feature. From the GUI window, click **Add from Plot**, then edit the wavelength to the known value. Click **Done** when finished. The mean shift calculated from all selected features will be applied as a zero-point shift to the wavelength calibration of all output spectra.
    - *Override shift*: Enter a value in dispersion units (e.g. μm) to use as the zero-point shift in wavelength calibration.

8 DATA QUALITY ASSESSMENT

After the pipeline has been run on a set of input data, the output products should be checked to ensure that the data has been properly reduced.

For imaging data:

- Check the final flat frame by comparing it with the raw image. All instrumental artifacts (areas of low quantum efficiency, obscurations in the optical path, and other systematics) that are present in the raw frame should be present in the flat. Sources in the raw image should not appear in the flat. Also check that the flat-fielded image does not contain any residual artifacts.
- Check for excessive hot or cold pixels in the coadded image. Bad pixels should be ignored in the coadding process.
- Check that the background was correctly subtracted. The counts in regions containing no sources should be zero, within the standard deviation.
- Check that the registration process calculated offsets correctly: look at all the registered images to verify that all sources appear at the same location on the array.

For grism data:

- Check the output to the terminal (or the log, in the case where the pipeline has been run by the automatic DPS system) for warnings or errors. Non-fatal warnings will be prepended with the string *WARNING*. Fatal errors will be prepended with the string *ERROR*.
- Check that the expected files were written to disk: there should, at a minimum, be a 2D image (*IMG*) and an extracted spectrum (*SPC* or *CMB)*.
- Look at the intermediate extracted spectra in *xvspec*: the positive and negative apertures of the same object should look similar. Spectra from separate observations of the same source in the same mode should look similar.
- Compare the signal-to-noise in the extracted spectrum to the reduced image. If the target seems bright in the reduced image, but the spectrum looks noisy, it may be that the aperture centers were not defined correctly.
- Check the aperture parameters recorded in the pipeline output and in the output file headers. The calculated aperture and PSF radii should have similar values for all apertures.
- Check the extracted spectra for excessive outliers. Most bad pixels should be removed in the extraction process. If they are not, it may be helpful to use the median spatial profile, or set the bad pixel threshold lower.

9 APPENDIX A: SAMPLE CONFIGURATION FILES

Sample FSpextool instrument configuration file, usually located at *fspextool/Instruments/FLITECAM/Data/FLITECAM.dat*.

```
#
# Generated by W. Vacca, 19 March 2012
#
# This is the calibration file for the FLITECAM spectrograph on SOFIA.
# Note the values must be in the correct order, but can have any number of
# spaces/comments between them.
#
#============================================================================
#
 INSTRUMENT=FLITECAM
 NCOLS=1024
 NROWS=1024
 STDIMAGE=1024
 PLOTWINSIZE=700 512
 FILENAME=FILENAME
 EXPTIME=EXPTIME
 TIME=TIME_OBS
 POSANGLE=ROT_ANGL
 HA=None
 AIRMASS=None
 NINT=4
 BADPIXMASK=None
%
%CAL BASE
%
 CALMODULE=mc_flitecamcals1d
%
%FILE READ MODE
%
 FILEREADMODE=Filename
 IPREFIX=F
 OPREFIX=r
 SUFFIX=*.fits*
 FITSREADPROGRAM=mc_readflitecamfits
 HEADCOMBPROGRAM=mc_flitecamdcshdr
 YUNITS=DN/s
 YTITLE=f (!5DN s!u-1!n)
 XUNITS=um
 XTITLE=!7k!5 (!7l!5m)
%
% Reduction Mode
%
 REDUCTIONMODE=A-B
%
% Combine Base Information
%
 COMBMODE=A
 COMBSTAT=Median (Median Error)
```

```
 COMBTHRESH=8.0
 COMBODIR=proc/
%
%  Sky Base Information
%
 SKYSTAT=Robust Weighted Mean
 SKYTHRESH=8.0
%
% Profile Parameters
%
 YBUFFER=2
 OVERSAMP=1
 ATMOSTHRESH=0.7
%
% Point Source Base
%
PSNAPS=2
PSPSFRAD=9.0
PSAPRAD=3.0
PSBGSUB=1
PSBGSTART=9.0
PSBGWIDTH=10
PSBGDEG=0
PSBGMULT=2.0
%
% Extended Source Base
%
XSBGSUB=1
XSBG=0-9,24-36,54-60
XSBGDEG=0
%
% Additional processing base
%
 ADDLMODULE=None
%
%Other Base Parameters
%
 TRACEDEG=2
 TRACESTEP=7
 TRACESUMAP=7
 TRACESIGTHRESH=1
 TRACEWINTHRESH=5
 BADPIXELTHRESH=4
 PLOTSATURATEDPIXELS=1
 SATURATION=6000
 CHECKSEEING=0
 SEEINGTHRESH=3
 LINCORRECT=1
 ERRORPROPAGATION=1
 FLATFIELD=1
 FIXBADPIXELS=1
 OPTIMALEXTRACTION=1
%
% FITS Header Keywords to Grab
% ---------------------------
```

[…]

Here follows a list of all keywords required to appear in FSpextool output files. This list includes all keywords listed in Appendix B, plus a few more keywords useful for informational purposes.

10 APPENDIX B: REQUIRED KEYWORDS

Keyword	Type	Allowed range	Condition	Required for:
ALTI_END	float	> 0, < 60000	*	Grouping
ALTI_STA	float	> 0, < 60000	*	Grouping
AOR_ID	string		*	Grouping
CYCLES	int	> 0	*	Data reduction
COADDS	int	≥ 1	*	Data reduction
DATASRC	string	ASTRO, CALIBRATION, LAB, TEST, OTHER, FIRSTPOINT	*	Archiving
DATATYPE	string	IMAGE, SPECTRAL	*	Grouping
DATE_OBS	string	yyyy-mm-ddThh:mm:ss[.sss]	*	Archiving
DITHERX	float		Dithering	Data reduction
DITHERY	float		Dithering	Data reduction
DIVISOR	int	≥ 1	*	Data reduction
EXPTIME	float	> 0	*	Data reduction
INSTCFG	string	IMAGING, GRISM, SPECTROSCOPY	*	Grouping
INSTRUME	string	FLITECAM	*	Archiving
INSTMODE	string	STARE, NOD_OFFARRAY, NOD_ALONG_SLIT, NOD_OFF_SLIT	*	Grouping
IRAFNAME	string		Imaging	Data reduction
ITIME	float	> 0	*	Data reduction
MISSN_ID	string		*	Archiving
NDR	int	≥ 1	*	Data reduction
OBJECT	string		*	Grouping

OBS_ID	string		*	Archiving
OBSTYPE	string	OBJECT, STANDARD_FLUX, STANDARD_TELLURIC, FLAT, DARK, SKY	*	Grouping
PROCSTAT	string	LEVEL_0, LEVEL_1, LEVEL_2, LEVEL_3, LEVEL_4	*	Archiving
SLIT	string	NONE, FLT_SS10, FLT_SS20	*	Grouping
SPECTEL1	string	NONE, FLT_DRK, FLT_J, FLT_H, FLT_K, FLT_ICE_308, FLT_PAH_329, FLT_Pa, FLT_Pa_cont, FLT_NbL, FLT_NbM, FLT_L, FLT_Lprime, FLT_M, FLT_B3_J, FLT_C4_H, FLT_A3_Hw, FLT_B2_Hw, FLT_C3_Kw, FLT_A2_KL, FLT_C2_LM, FLT_B1_LM, FLT_A1_LM	*	Archiving, grouping
SPECTEL2	string	NONE, FLT_SS10, FLT_SS20	*	Archiving, grouping
SRCTYPE	string	POINT_SOURCE, EXTENDED_SOURCE, OTHER, UNKNOWN	Spectroscopy	Data reduction
TABLE_MS	float	> 0	*	Data reduction
ZA_END	float	>0, <90	*	Grouping
ZA_START	float	>0, <90	*	Grouping