The CIRSI LCIRS Pipeline

November 30, 1999

Andrew Firth

Overview of the Pipeline

The pipeline currently consists of four tasks cirproc, cirdither, wcsfit and cirknit.

- cirproc is responsible for preprocessing to remove reset residuals, flat fielding and sky subtraction. Sky subtraction involves either fitting a background map to each frame or making a sky frame from the image frames themselves. Objects are then located and masked and the preprocessing, flat fielding and sky subtraction are repeated.
- cirdither sorts the frames into dither sequences and adds the frames in each dither, resulting in 16 images for a full mosaic.
- wcsfit matches objects with the APM catalogue to determine wcs coordinates which it then writes to the image headers.
- cirknit makes the final mosaic using rotations and sub-pixel translations as necessary.

Parameters[1]

Each field is observed on 3-6 nights in each filter. One night of observations typically comprises four pointings (SE, NE, NW and SW) to complete a 13' × 13' mosaic with 5 or 9 dithers per pointing and 3 loops of 90s in J or 4 loops of 45s in H per dither, for a total of 20 or so frames per pointing per chip and 80 or so frames per field per night per chip.

The current reduction of the NTT field (tile 1) involved[2]:

1. No preprocessing to remove reset residuals.
2. Flat fielding using 3 σ-clipped median with loop 1 of the $\approx$ 80 frames per field per night per chip separated from loops 2 and 3.
3. Sky subtraction using 3 σ-clipped median of the $\approx$ 20 frames per pointing per night per chip.
4. Repeat flat fielding and sky subtraction with object masking.
5. Coadding of loops and dithers for each night using 3 σ-clipped means.
6. Mosaicing the 60 or so images being 4 chips × 4 pointings × 4 nights using 3 σ-clipped means and weighting images by the inverse square of the sky rms as computed with phiirs.iterstat.

[1] A full list of parameters used is contained in Appendix A.
[2] /data/H_cass19e/aef/NTT_again/Dithers/mosaic2.fits.

The final mosaic is approximately 5400s in J. This may be compared with the 4000s H and 8000s J reductions of the IoA1511 field (tile 1)[3]which were processed in essentially the same way (though a different routine was used to generate object masks for sky subtraction) except that medians were used instead of means in cirdither and cirknit. This results in a cosmetically rather nicer picture but seems to affect the photometry. Hopefully the reset correction algorithm, pixel-by-pixel weight maps in cirknit and optimized σ-clipping will tidy up the image while allowing the use of means instead of medians.

Pipeline problems

The following is a list of items that I think need investigating:

1. A pre-pipeline stage could be introduced to sort out, with minimal user input, a whole night's data dumped onto disk. The data may include fields, standards, twilight and dome flats, focus tests, other tests and bad data. Some image headers may be wrong (such as `filter`, obstype, imagtype which are entered by the observer) so `int_time`, `ra`, dec and sky counts per second may be used to check the observation type, field and filter etc. (More ambitiously one could also try to pick out smeared images or ones with satellite trails.)

2. BPMs need updating nightly since dust on the chips moves on time-scales of one to a few nights. It would be useful if these could be generated automatically in the pipeline from the data or from dome flats obtained each night.

3. The chip 4 problem with the reset correction algorithm needs fixing. This occurs because vignetting at the left-hand side of chip 4 on one night goes to -1000 counts. When cirproc fits a polynomial to remove the reset residual this results in an image with -ve sky. When it makes flat-fields it doesn't know what to do with the scaling and weighting. This particular problem should be removed with a nightly updated BPM. More generally after the polynomial fit is subtracted the sky should be left at the level of the majority of the chip rather than the level corresponding to the bottom of the ramp. Otherwise the flats will be messed up as well since the separation of additive and multiplicative components will be incorrect.

4. What is the optimum way to scale and or weight images when making flats and skies? Similarly when adding data images. A noise-estimate (the sky median from iterstat or similar) could be written into the headers before sky-subtraction and some sort of average written into the headers when images are added by cirdither. What about scaling/weighting for airmass and seeing and what is the procedure (if any) for not-entirely photometric data?

5. Are medians photometrically accurate or should we stick with means? What is the optimum degree of σ-clipping?

6. Super sky flats versus dome flats versus twilight flats or some combination. Super sky flats include fringes which are additive not multiplicative so may mess up the photometry. Twilight flats are a different colour and the counts may be much higher and exposure times much shorter than in data frames so may be subject to non-linear effects in the detectors. Dome flats are also a different colour and there may be large scale variation. If we use super sky flats then do we use a single field, a whole night or the entire data set to make them. Or we could make super sky flats from 3am-6am frames when the fringes are reduced compared to straight after sundown.

7. cirdither could check that offsets for the different chips are identical and use this to self correct errors. Similarly in wcsfit, a limit could be imposed on the skewing and

[3]/data/mds13/aef/Tar/H_Mos.fits and /data/mds13/aef/Tar/J_Mos.fits.

offsets from the expected position to reduce the number off mismatches, and/or the noisy edges could be ignored during object detection. cirdither needs an interactive option (involving clicking on stars, for example, or inputing a coordinate list which the user has constructed by inspection of the offsets of other chips and other loops).

8. If flats are not normalized to the standard star chip in the initial processing stages then this should be incorporated later on, eg. in cirknit. The necessary information could be written to the headers at the cirproc stage.

9. Similarly (but not really a pipeline problem) if the QE varies significantly over a single chip then the fact that standards are taken with only a small section of chip 1 needs to be taken into account.

10. When sky-subtracting other people tend to use a sequence of say 15 adjacent images, not including the particular image being processed and discarding images with vastly different sky-values. Do we want to do this?

11. Which option is best for sky-subtraction? The 'back' option (fitting a surface) may not be able to fit fine fringes. The 'frame' option (subtracting medianed average of consecutive frames) may suffer if the sky counts change a lot during the night if a long sequence of images are used to make the sky frame or if a short sequence is used the skies will suffer from too much noise.

12. What is the optimum level of masking? What currently happens if by chance there is no data in a pixel due to a combination of masked images and BPMs?

13. At the edges of a frame produced from combining a dither sequences each pixel will have been derived from only 3 or so images while in the centre each pixel will have been derived from about 15 images. When these frames are combined into a mosaic by cirknit the pixels need to be weighted accordingly. A map of exactly how many pixels were used to produce each pixel of the final frame needs to be kept. Eventually when the mosaics from several nights are combined the sum of these maps (weighted appropriately) will give an exposure/noise map for the entire field.

14. To what degree is non-linearity of pixels as a function of flux and waveband, not to mention loop number and temperature, a problem? Is it possible to produce a response curve for every pixel and develop a look-table for processing data? (Chris Sabbey is looking at this.)

A Full listing of pipeline parameters

PACKAGE = cirdr
TASK = cirproc

runs	=	1753-1788	Gimme a range of runs
preproce	=	yes	Preprocess to remove reset residuals?
dostripe	=	no	Destripe the images?
transpos	=	none	Image transposition algorithm
darkcor	=	no	Correct for dark current?
darkfram	=		Mean dark frame root name
flatcor	=	yes	Flat field correct frames?
meanflat	=	flat	Mean flat field root name
ssalgo	=	back	Sky subtraction algorithm
meansky	=	sky	Mean sky frame root name
boxback	=	64	Smoothing box size for background subt
findobje	=	yes	Locate objects?
(preproc	=	)	Parameters for residual reset removal
(stripep	=	)	Parameters for destripe
(mkflatp	=	)	Parameters for making mean flat
(mkskypa	=	)	Parameters for making mean sky files
(apmpars	=	)	Parameters for object location
(list	=	)	
(mode	=	ql)	

PACKAGE = cirdr
TASK = preprocpars

(xord	=	7)	X order for surface fit
(yord	=	7)	Y order for surface fit
(niter	=	2)	Number of rejection cycles
(rejthre	=	3.)	Rejection threshold for bad pixels
(box	=	16)	Smoothing box size
(mode	=	ql)	

PACKAGE = cirdr
TASK = stripepars

(ifilt	=	3)	Filter X, Y or Both (1, 2 or 3)
(iback	=	yes)	Do background correction
(iscale	=	16)	Size of smoothing box for background correction
(ixstart	=	INDEF)	Starting X position for destripe
(ixend	=	INDEF)	Ending X position for destripe
(iystart	=	INDEF)	Starting Y position for destripe
(iyend	=	INDEF)	Ending Y position for destripe
(mode	=	ql)	

PACKAGE = cirdr
TASK = mkflatpars

(maxcomb	=	100)	Maximum number of frames to combine (0 = all)
(plfile	=	)	List of output pixel list files (optional)
(logfile	=	)_.logfile)	Log file
(combine	=	median)	Type of combine operation
(reject	=	sigclip)	Type of rejection
(blank	=	0.)	Value if there are no pixels
(expname	=	INT_TIME)	Image header exposure time keyword
(lthresh	=	INDEF)	Lower threshold
(hthresh	=	INDEF)	Upper threshold
(nlow	=	1)	minmax: Number of low pixels to reject
(nhigh	=	1)	minmax: Number of high pixels to reject
(nkeep	=	1)	Minimum to keep (pos) or maximum to reject (neg)
(mclip	=	yes)	Use median in sigma clipping algorithms?
(lsigma	=	3.)	Lower sigma clipping factor
(hsigma	=	3.)	Upper sigma clipping factor
(rdnoise	=	0.)	ccdclip: CCD readout noise (electrons)
(gain	=	1.)	ccdclip: CCD gain (electrons/DN)
(snoise	=	0.)	ccdclip: Sensitivity noise (fraction)
(sigscal	=	0.1)	Tolerance for sigma clipping scaling corrections
(pclip	=	-0.5)	pclip: Percentile clipping parameter
(grow	=	0)	Radius (pixels) for 1D neighbor rejection
(mode	=	ql)	

PACKAGE = cirdr
TASK = mkskypars

(plfile	=	)	List of output pixel list files (optional)
(logfile	=	)_.logfile)	Log file
(combine	=	median)	Type of combine operation
(reject	=	sigclip)	Type of rejection
(blank	=	0.)	Value if there are no pixels
(expname	=	INT_TIME)	Image header exposure time keyword
(lthresh	=	INDEF)	Lower threshold
(hthresh	=	INDEF)	Upper threshold
(nlow	=	1)	minmax: Number of low pixels to reject
(nhigh	=	1)	minmax: Number of high pixels to reject
(nkeep	=	1)	Minimum to keep (pos) or maximum to reject (neg)
(mclip	=	yes)	Use median in sigma clipping algorithms?
(lsigma	=	3.)	Lower sigma clipping factor
(hsigma	=	3.)	Upper sigma clipping factor
(rdnoise	=	0.)	ccdclip: CCD readout noise (electrons)
(gain	=	1.)	ccdclip: CCD gain (electrons/DN)
(snoise	=	0.)	ccdclip: Sensitivity noise (fraction)
(sigscal	=	0.1)	Tolerance for sigma clipping scaling corrections
(pclip	=	-0.5)	pclip: Percentile clipping parameter
(grow	=	0)	Radius (pixels) for 1D neighbor rejection
(mode	=	ql)	

PACKAGE = cirdr
TASK = apmpars

(thresh	=	20.)	Object threshold
(keepbac	=	no)	Keep the background map?
(pnop	=	20)	Minimum number of pixels per object
(grid	=	64)	Size of grid background sampling
(mode	=	ql)	

PACKAGE = cirdr
TASK = cirdither

runlist	=	1753-1788	Input run list
chipno	=	0	Which chip (0 == all)?
output	=	c	Output file name root
(mkdithp	=	)	
(fd	=	)	
(fd2	=	)	
(fd3	=	)	
(mode	=	ql)	

PACKAGE = cirdr
TASK = mkdithpars

(logfile	=	)_.logfile)	Log file
(combine	=	average)	Type of combine operation
(reject	=	sigclip)	Type of rejection
(blank	=	0.)	Value if there are no pixels
(lthresh	=	INDEF)	Lower threshold
(hthresh	=	INDEF)	Upper threshold
(nlow	=	1)	minmax: Number of low pixels to reject
(nhigh	=	1)	minmax: Number of high pixels to reject
(nkeep	=	1)	Minimum to keep (pos) or maximum to reject (neg)
(mclip	=	yes)	Use median in sigma clipping algorithms?
(lsigma	=	3.)	Lower sigma clipping factor
(hsigma	=	3.)	Upper sigma clipping factor
(rdnoise	=	0.)	ccdclip: CCD readout noise (electrons)
(gain	=	1.)	ccdclip: CCD gain (electrons/DN)
(snoise	=	0.)	ccdclip: Sensitivity noise (fraction)
(sigscal	=	0.1)	Tolerance for sigma clipping scaling corrections
(pclip	=	-0.5)	pclip: Percentile clipping parameter
(grow	=	0)	Radius (pixels) for 1D neighbor rejection
(mode	=	ql)	

PACKAGE	= cirdr		
TASK	= wcsfit		
inlist	=	c*.fits	List of files or range of run numbers
(intype	=	file)	Input type
(ithresh	=	0.75)	Fit error threshold for interactive refit
(survey	=	ukst)	Name of survey to use
(nst	=	50)	Number of stars to use
(giverou	=	yes)	Give files a rough WCS to start with?
(fd1	=	)	
(fd2	=	)	
(mode	=	ql)	

PACKAGE	= cirdr		
TASK	= apm		
inlist	=		List of input files
(thresh	=	-2.)	Detection threshold
(keepbac	=	no)	Keep a copy of the background map
(pnop	=	5)	Minimum number of pixels to define an object
(grid	=	64)	Grid size for background map
(verbose	=	no)	Verbose output
(domask	=	no)	Make a mask of pixels used for objects?
(mode	=	ql)	

PACKAGE = cirdr
TASK = cirknit

inlist	=	c*.fits	List of files or range of run numbers
intype	=	file	Input type
output	=	mosaic	Output file
(cirknit	=	)	Knitting parameters
(fd	=	)	
(mode	=	ql)	

PACKAGE = cirdr
TASK = cirknitpars

(logfile	=	)_.logfile)	Log file
(plfile	=	)	List of output pixel list files (optional)
(combine	=	average)	Type of combine operation (median—average)
(reject	=	sigclip)	Type of rejection
(masktyp	=	goodvalue)	Mask type
(maskval	=	0.)	Mask value
(blank	=	0.)	Value if there are no pixels
(scale	=	none)	Image scaling
(zero	=	none)	Image zero point offset
(weight	=	none)	Image weights
(lthresh	=	1.)	Lower threshold
(hthresh	=	INDEF)	Upper threshold
(nlow	=	1)	minmax: Number of low pixels to reject
(nhigh	=	1)	minmax: Number of high pixels to reject
(nkeep	=	1)	Minimum to keep (pos) or maximum to reject (neg)
(mclip	=	yes)	Use median in sigma clipping algorithms?
(lsigma	=	3.)	Lower sigma clipping factor
(hsigma	=	3.)	Upper sigma clipping factor
(rdnoise	=	0.)	ccdclip: CCD readout noise (electrons)
(gain	=	1.)	ccdclip: CCD gain (electrons/DN)
(snoise	=	0.)	ccdclip: Sensitivity noise (fraction)
(sigscal	=	0.1)	Tolerance for sigma clipping scaling corrections
(pclip	=	-0.5)	pclip: Percentile clipping parameter
(grow	=	0.)	Radius (pixels) for neighbor rejection
(mode	=	ql)	