IDL Image Reduction

I. Introduction: Should write a brief intro here.

II. Requirements:

- >300M RAM (this can be modified)
- the XIDL package
- the IDL coyote package
- the Goddard IDL package
- the Schlegel IDL package (compiled)

III. Setup

(a) Create a Raw/ directory and put all the raw data in it. Move to the directory above Raw

(b) **xdimg_strct** :: This creates the IDL structure which drives all of the subsequent programs. It also creates a fits file of the structure ("struct.fits") and a summary in the ASCII file "image.list". After running the program, one should modify the resulting structure as needed. A brief summary of the structure is given below. An example of this step is xdimg_setup.pro

Example: **xdimg_strct**, n1_strct, 'LRISR', 'Keck', /MKDIR

Time : $<$ 5s per image

(c) **distruct**

Tag	Type	Comment
flg_anly	0	Analysis flag 0=Don't Analyse
frame	0L	FRAME Number
naxis1	0L	NAXIS1 is helpful for determining sub-chip readouts, esp for NIRC2
naxis2	0L	NAXIS2
exp	0.d	Exposure time
filter	' '	Filter : U,B,V,R,I,H,Kp
type	' '	ObjTyp : OBJ, STD, TWI, DRK, ZRO, DFT, SPS
Obj	' '	Object Name
Equinox	0.	EQUINOX
Date	0.0d	Date of Obs
UT	' '	UT
RA	' '	RA
DEC	' '	DEC
TEL	' '	Telescope
AM	0.	Airmass
RMS	0.	RMS
extinct	0.	Extinction (e.g. cirrus)
CCD	' '	CCD : LRISR, NIRC2, SITe3, etc.
namp	0	Namps
coadds	0L	Number of coadds (For IR images)
sampmode	0L	Sampmode
multisam	0L	Multisam
statsec	lonarr(4)	Stat section
gain	0.	Gain
readno	0.	Read Noise
satur	0.	Saturation
rootpth	' '	Path of the Root
img_root	' '	Root name (usually in Raw directory)
med_raw	0L	Median of the Raw image
min_raw	0L	Min of the Raw image
max_raw	0L	Max of the Raw image
flg_ov	0L	OV FILE? 0=No, 1=Yes
img_ov	' '	Name of OV file (with directory)
med_ov	0.	Median of OV file
flg_drk	0L	Does necessary dark file exist? 0=NO, 1=Yes
img_drk	' '	Name of dark file which should be used with this image. (with directory)
flg_drksub	0L	Has image been dark subtracted? 0=NO, 1=Yes
img_drksub	' '	Name of the dark subtracted (and coadd divided) file (with directory)
med_drksub	0.	Median of dark subtracted, coadded divided file
flg_msk	0L	Mask FILE? 0=No, 1=Yes
img_msk	' '	Name of Mask file
flg_skymsk	0L	Sky Mask FILE? 0=No, 1=Yes
img_skymsk	' '	Name of Sky Mask file
flg_final	0L	Final File? 0=No
img_final	' '	Name of Final img

IV. CALIBRATIONS ::

The calibration section is split into two parts, the first for creating calibration files for use with optical images and the second for creating calibration files for use with infrared images.

Note that at any point during the reductions you can exit out of IDL, and re-start the calibrations later by reading in the structure as follows: n1_strct=mrdfits('struct.fits',1). The structure is re-saved at each step in the process.

OPTICAL CALIBRATIONS ::

Create Biases and Flats, including a Super-sky flat. An example of these steps is given in Notes/pro/xdimg_calibs_opt.pro. You will need one set of flats but otherwise these steps are optional. For the flats, a non-linear correction is applied where applicable (e.g. SITe3)

(a) **xdimg_bias** :: Create the Bias frame. This routine ov subtracts the bias images and median combines. It then deletes the ov images unless instructed otherwise. Currently, the 'Tek5', 'Site#1', 'LRISR' chips are supported. Investigate the bias as you like (e.g. xatv, 'Bias/Bias.fits')
Example: xdimg_bias, n1_strct
Time : 5 min

(b) **xdimg_dflat** :: Create the Dome Flats. This routine ov subtracts the dome flat images and median combines. It then deletes the ov images unless instructed otherwise. It creates one normalized flat per filter. Investigate the flats as you like (e.g. xatv, 'Flats/DFlatNR.fits').
Example: xdimg_dflat, n1_strct
Time : 10 min

(c) **xdimg_twiflt** :: Create the Twilight Flats. This routine ov subtracts the twilight flat images and median combines. It then deletes the ov images unless instructed otherwise. It creates one normalized flat per filter. Investigate the flats as you like (e.g. xatv, 'Flats/TFlatNR.fits').
Example: xdimg_twiflt, n1_strct
Time : 10 min

INFRARED CALIBRATIONS ::

Create darks, flats, and a bad pixel mask. An example of these steps is given in Notes/pro/xdimg_calibs_ir.pro. You will need one set of flats for each filter, and a super-dark frame for each exposure time of science data. The bad pixel mask is helpful but not necessary.

(a) **xdimg_badpix** :: Create a bad pixel mask. This routine uses the dark frames to pick out hot pixels and it uses well-exposed flat fields to pick out dead pixels. Therefore the structure must include at least one of these types of frames. You can set the DEADPERCENT and HOTPERCENT keywords to change the threshold of what is considered a bad pixel. If multiple flats and darks are present the program goes through each frame looking for dead and hot pixels. In this way it can locate "sometimes" dead and hot pixels. If running this program on NIRC2 data, you must run it once for the narrow camera and once for the wide camera since these are treated as separate instruments.
Example: xdimg_badpix, n1_struct, CCD='NIRC2W'

(b) **xdimg_dark** :: Create a super-dark frame for every type of dark frame that we have. For example, there will be one super-dark frame for each unique combination of exposure time, coadds, sampmode, and nreads. This routine median combines all the dark exposures of the same type. Investigate the super-darks as you like (e.g. xatv, 'Darks/Dark_w10.000_6_3_8.fits') The darks are saved with the NIRC2 camera name, the exposure time, the number of coadds, the sampmode, and the number of reads (in that order).
Example: xdimg_dark, n1_struct

(c) **xdimg_dflat** :: Create the dome flat. When used with the /IR flag, this routine dark subtracts the flats with the super-dark frame of the same type, and then median combines, scaling by the median value in a small portion of the image. It then deletes the dark subtracted images unless instructed otherwise. It creates one normalized flat per filter, and per camera, of course. Investigate the flats as you like (e.g. xatv, 'Flats/DFlat_w_NKp.fits').

Example xdimg_dflat, n1_strct, /IR, CCD='NIRC2W', BPM='PixMask/BadPixMask_w.fits'

V. Sky Flat ::

This is the preferred flat for analysis. Although strictly a calibration file, this step can be more interactive than the previous and deals with the data images. Ideally one has > 5 well dithered images per filter. The routine:

(a) OV subtracts the object images (usually interactively with the /INTER keyword). If the /IR flag is supplied then the routine dark subtracts instead of OV subtracting.

(b) Runs a gui (**xmkmask**) designed to make sky masks (suppressed with the /NOMSK keyword).

(c) It then median combines with clipping.

The ov images are generally kept. The routine creates one normalized flat per filter. Investigate the flats as you like (e.g. xatv, 'Flats/SkyFltNR.fits' OR xatv, 'Flats/SkyFlt_w_Nkp.fits').

Note: If xdimg_skyflt fails due to problems with dynamic linking, it is possible that your installation of XIDL was faulty. Update the XIDL CVS repository, and be sure to type bin/evilmake clean followed by bin/evilmake all while in XIDL.

OPT Example: xdimg_skyflt, n1_strct, /INTER

IR Example: xdimg_skyflt, n1_strct, /IR, CCD='NIRC2W', /NOMSK

Time : 45min (masks), 10min combining

- **xmkmask** :: IDL gui for creating masks of a set of images. Use the keyword /SKYMSK to create a skymask. User can create rectangular, circular, and line masks.
  - Rectangle: LMB, LMB on corners of rectangle
  - Circular: LMB, LMB on center then outer radius of circle
  - Line: LMB, LMB, LMB on center of line, then one edge of the line and then the other center

  After saving the first Mask, one can dither the same Mask on the next image using the dx, dy values [Note: minus signs require you type a non-zero number first; this is confusing]. Set them one by one and hit return to update the image+mask. Iterate. Click on SVDITH when done or on NODITH to delete the current Mask. You can then add new sub-masks to the current image as desired. The Masks are written to the Masks/ directory by default and gzipped to save memory.

VI. PHOTOMETRIC CALIBRATION ::

(a) See xdimg_stds.pro for an example.

(b) **xdimg_proc** :: Process the standard field images. Identify all STD images with the **where** command, then run **xdimg_proc**. You will need to choose which image to flatten with and input the root name (e.g. 'Flats/TSkyFltN' or 'Flats/DFlatN'). Notice you do not use the entire flat name, just the filterless portion. If dealing with infrared data, you will need to use the /IR flag, and you should provide the name of a bad pixel mask if you have one. The resulting images are in the Final/ directory with an 'f_' prefix. Note that this procedure does NOT multiply by the gain by default. Make sure the date is properly set for each image or is set globally!

OPT Example: STDS = where(n1_struct.type EQ 'STD' AND n1_struct.flg_anly NE 0)
xdimg_proc, n1_struct, STDS, 'flatroot'

IR Example: STDS = where(n1_struct.type EQ 'STD' AND n1_struct.flg_anly NE 0 and n1_struct.ccd eq 'NIRC2W')

xdimg_proc, n1_struct, STDS, 'flatroot', /IR, BPM='PixMask/BadPixMask_w.fits'
Time : Approx 15 sec / image

(c) **xdimg_starid** :: Identify the stars in each standard field. This routine displays the image, allows one to flip it, directs the reader to read in a star list relevant to the field, and then facilitates the identification of all stars in that field. It is not necessary to have the standard fields in the local List directory before starting this procedure. (By default, the program will search the /IMG/Photometry/Lists directory.) But it is advised to use the LST_PATH keyword and copy over the list locally. You'll want to be familiar with the order of the stardards in the list, because you'll have to click on the stars in the image one by one, in the order that they appear in the starlist. Once you have identified and centered the stars, you can save the list which is convenient given the poor quality of the Landolt coordinates and proper motions. This program drives the program **x_starid** which is an IDL gui. See its help page for more details. Play around with the inorient keyword to get the standard stars to line up correctly on the image. After running xdimg_starid, there should be a new file in the Photo directory called std_****.dat, and this file should contain the pixel positions of the standard stars.
Example: **xdimg_starid**, n1_struct, inorient=4, LST_PATH='/Users/pollack/foo/Lists/'
Time : Mostly human labor (<1 min / image)

(d) **xdimg_stdmag** :: Calculate magnitudes for the stars. This routine takes the star positions from the previous step, reads in each image and calculates $7''$ aperture magnitudes. The output is written to Photo/ by default (mag_*). Note: If you found that some of the standard stars are staruated, or fall on a bad column, or whatever, you may want to edit the "mag" files in the Photo directory before continuing on to the next step of photcal. You'll have a chance in xdimg_photcal to remove bad data points, but it may not be obvious which ones are bad.
Example: **xdimg_stdmag**, n1_struct
Time : Fast, 5s/image

(e) **xdimg_photcal** :: Derive the final solution. This routine takes the magnitudes from the previous step, reads in the Landolt info, and then creates and plots up the solution. One can set an AM term instead of fitting for it with the SETAM keyword. (For a good discussion on airmass, and to find values of AM to use in case you can't fit for it yourself, see Prochaska et al. 2001, AJ, 123, 2206.) Also the default is to only include Landolt stars with at least 4 measurements. This can be changed with the MIN_NOBS and MIN_MOBS keywords. The output is written as 'soln_*' in the Photo directory.
Example: **xdimg_photcal**, n1_struct, setam=0.7, min_nobs=1, /NCLR
Time : Fast

(f) Checkout 'soln_*' in the Photo directory for the final photometric solutions. The output given is the filter for the solution, and the color, followed by the ZP, AM, CLR fitted parameters (with corresponding 1σ errors on the last line). The final magnitudes are given by:

$$\text{Mag} = \text{m} + 25.0 + \text{ZP - AM*obs\_AM - CLR*obs\_clr}$$

Of course, the +25.0 is simply the default when you use x_aper or daophot to do standard aperture photometry. i.e. If you look at the mag_****.dat files which were created by xdimg_stdmag, which are stored in the Photo directory, these values are already (m + 25.0) in the above equation. Note: the quoted zero point applies to the images processed by xdimg_proc, that is images which already have had the gain applied!

VII. Coadd the Images

(a) **xdimg_proc** :: Process the data images. Make sure to set the correct flat root. Use the /IR flag if dealing with infrared data, and also provide the name of a bad pixel mask if one exists. If you want to later coadd the final images using Josh's code, include a FILEOUT name to be written with a list of final files.

OPT Example: IMS = where(n1_struct.type EQ 'OBJ' AND n1_struct.flg_anly NE 0)
xdimg_proc, n1_struct, IMS, 'flatroot'
IR Example: IMS = where(n1_struct.type EQ 'OBJ' AND n1_struct.flg_anly NE 0 and n1_struct.ccd eq 'NIRC2W')
xdimg_proc,
n1_struct, IMS, 'Flats/SkyFlt_w_N', /IR, BPM = 'PixMask/BadPixMask_w.fits', FILEOUT = 'stackFile'
Time : Approx 20 sec / image

(b) **x_offset** :: This routine takes a series of images (in a list) and calculates the integer offset between number 2-n and number 1. It uses sExtractor to identify objects and then cross-correlates the images in a very simple way. Copy the default.* files from $XIDL/IMG/Notes/files to a new directory (e.g. '[obj]/offsets') and create a file lists the final images. The default output is a file 'offsets' with integer offsets. Example: **x_offset** 'obj.list' INSTR='SITe3'
Time : 1min/image

(c) Combine! :: **THIS IS WRONG!!!** Use IRAF's *combine* to register the processed images. Be sure to weight by exposure and scale by the median. This has the same effect as just summing each image straight-up (i.e. preserving total counts). **After a long discussion with Hsiao-Wen, we determined that what you should really do is subtract the median from each each, and then coadd the median-subtracted images by weighting by the inverse variance and scaling by the exposure time.**

VIII. Measuring Magnitudes

We will use sextractor and in order for it to work properly the images must be in counts per second (i.e. counts for which Poisson Statistics applies), have the same dimensions and be registered [these last requirements may no longer be true, I can edit the in-house code to handle different sizes and registrations]. We will use the R band to select our images and set the aperture sizes. We then will use an in-house program to do the photometry

(a) Normalizing :: Divide each image by the value in TTIME in the header.
Example: imarith Q0951B / 600.0 Q0951NB
Time : Quick

(b) Registering :: Either use the Gawiser method where you give the offset program the final B and R images and calculate the offset and repeat for I or just use your eyeball and ximtool. Then use imcopy to copy subsections of each image into a final 3 images which have the same dimension and are registered.
Example: imcopy Q0951NB[71:1839,86:2269] Q0951XB
Example: imcopy Q0951NR[1:1769,1:2184] Q0951XR
Example: imcopy Q0951NI[2:1770,3:2186] Q0951XI

(c) Run Sextractor :: Copy all of the default.* files as well as the example .sex files from /home/xavier/Keck/LRIS/DLAImg/Sex/ into a subdirectory (Sex). Copy the *.sex files to the appropriate name. Edit the *.sex files changing CATALOG_NAME, MAG_ZEROPOINT (this is the photometric zero point taking airmass into account), SATUR_LEVEL (55000 / TTIME), SEEING_FWHM, and CHECKIMAGE_NAME. Finally, run sextractor on the frame for which you wish to identify objects and apertures. We have been using R.
Example: sex ../Q0132XR.fits -c Q0132+13R_ap.sex

(d) Run SBlimit :: Check the Surface brightness limit. Create a file SBlimit.inp in Input/ with (1) Npix in aperture e.g. $1''$, (2) Nfilt, (3-?) $b1, b2, XB$; σ^{DN} repeat
Ex: SBlimit

(e) Run lris_clrphot :: This program will parse the sextractor output and calculate mangitudes for BRI based on the output from sextractor. The key is that the apertures are of a fixed size for all of the images. Copy the input file lris_clrphot.inp from DLAImg/Input into your Input directory and edit everything. The first 9 lines are the fits file, total exposure time (s), and Airmass for each filter. The next two are the aperture and catalog files from sextractor. The last 3 lines are: (1) the number of dummy comments in the sextractor output, don't touch this; (2) the photometric calibration ($b1, \sigma(b1), b2, \sigma(b2), b3, \sigma(b3)$) and (3) the 1σ limiting magnitude which we will use in the case when nothing is detected. The output is fort.11 which has: (i) running number, (ii) x_pix, (iii) y_pix; (iv) Bmag, (v) Rmag, (vi) Imag, (vii) sex mag.

Example: lris_clrphot

(f) Run lris_3comp :: In theory you are ready to rock. Just copy over the lris_3comp.inp file and get to it.

Example: lris_3comp