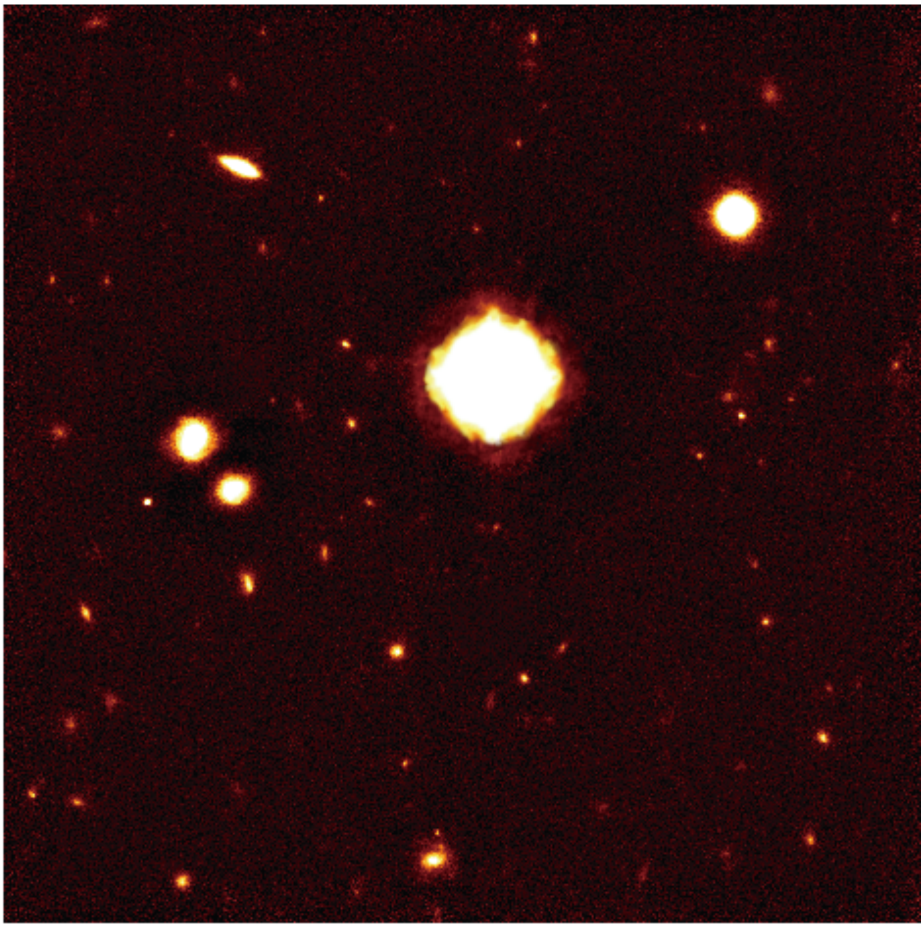
Subaru Data Reduction Cookbook: Imaging Observations with IRCS

— Version 2.0.3e (January 5, 2010) —



Based on the textbook in Japanese by Y. Minowa
for the Subaru Data Reduction School held in December 2006
Current Editor of English Version: R. S. Furuya,
with the combined effort of the past and current staff at Subaru Telescope

1 Foreward

This data reduction cookbook describes how to reduce your imaging data acquired with the Infrared Camera and Spectrograph (IRCS) on the Subaru Telescope. The basic concept and methods to reduce near-infrared (NIR) data from IRCS are essentially the same as those at optical regime taken with CCDs. NIR imaging differs from those at optical in the sense that background emission from the atmosphere and telescope is intrinsically higher than those at optical wavelength. In this COOKBOOK we will start to explain how to reduce your IRCS data by attracting your attention to such differences characterizing NIR imaging.

We will subsequently describe the practical tips for calibration of your data with IRAF. As for the details of IRAF commands, please refer to other documents available elsewhere. In this COOKBOOK, we will use originally written IRAF scripts of `ircs_imgred` written one of us (Y. Minowa), which can be obtained from the Subaru Data Reduction web page. This COOKBOOK had been originally written by Y. Minowa for the Subaru Data Reduction school in 2006, intending to give a basic knowledge of NIR data reduction for novices. We realize that this COOKBOOK is far from the complete and are more than happy getting your questions and positive feedbacks to improve this COOKBOOK.

2008 October 31

Y. Minowa (minoways at subaru.naoj.org)
R. S. Furuya (rsf at subaru.naoj.org)

Contents

Table 1: **Revision History**

Version	Date	
2.0.1e	2008 October 31	First release based on the Japanese version (2006 December 4) by R.S.F., & Y.M.
	2008 November 30	Language correction for *the GUAS Subaru Asia Winter School* by R.S.F. & A.H.
2.0.2e	2008 December 8	Minor corrections after the Winter School by R.S.F.
2.0.3e	2010 January 5	Minor cosmetic corrections by R.S.F.

2 Imaging Observations in Near-infrared (NIR) Regime

2.1 Characteristics of Near-infrared Emission

Different from the visible light, the ground based near-infrared (NIR) observations of astronomical objects are limited at a few wavelength ranges (see Figure 1) existing between 1 and 5 μm (the ranges are often referred to as bands), being hampered by the absorption by the terrestrial atmosphere. Almost all the filters employed in astronomical imaging at near-infrared (NIR) regime are designed by considering such atmospheric windows. Since the degree of absorption is proportional to column density of water vapor existing in the atmosphere, we can reduce such an effect by observing at high altitude site.

Because background radiation from atmosphere and telescope itself at NIR is much higher than those at optical wavelengths, expected sensitivity in NIR is generally worse than optical due to Poisson noise. We present background emission spectrum from optical through mid-infrared (MIR) in Figure 2. In the figure, one would realize that OH "airglow", i.e., OH vibration-rotation bands, is responsible for the background at $\lambda \lesssim 2.3\,\mu$m and that thermal (blackbody) emission from the atmosphere and telescope dominate the background at $\lambda \gtrsim 2.3\,\mu$m. It should be noted that the background level at the long wavelength dramatically increases at $\lambda \gtrsim 2.3$ μm.

In general, background emission in NIR is highly time variable (due to e.g., weather conditions, airmass variation etc), such that its brightness varies not only with a function of time but also pixel positions across the field of view. In order to remove background from raw data, it is also required to use the images that include your target. However, because the target frames are not necessarily emission free (of course!), we have to make an image that contains only the background emission by combining several images, with median averaging, which have been taken at the slightly offset'ed position around the target. The observing method, which sequentially points the telescope to slightly different positions in the plane-of-sky, is called dithering observation (see Figure 3). Dithering is also useful technique to eliminate effects of bad pixels in the detector.

Since CCDs used in the optical wavelengths do not have sensitivity in NIR, array detectors with Indium antimonide (InSb) or mercury cadmium telluride (HgCdTe) are mainly used for NIR observations. Infrared detectors are characterized by Direct Read Out (DRO) which directly reads output voltage from each pixel without transferring electrons as CCD does. This is a benefit in that DRO tends not to cause blooming that usually occurs at optical wavelength when observing very bright stars. Another characteristics of NIR imaging is usage of Non-Destructive Readout (NDR) which allows extraction of electron information without destroying the electrons during integration. NDR contributes significantly to reduce read-out noise. Recall that noise decreases with $1/\sqrt{n}$ for n times of NDRs.

In general, a short exposure time for an integration is required to avoid detector saturation due to high background. At the L- and M bands (3.5 and 4.8 μm, respectively) where thermal background emission is very intense, it is generally difficult to integrate more than 1 second coherently. We therefore adopt COADD method in order to allow long-integration exceeding such a coherence limit. COADD acquires many images taken with short integration times, and adds them on the memory to make a frame of a long integration at all the bands without saturating the signals of interests.

The electrons corrected by the infrared detector are amplified by the readout circuit, and is subsequently converted into digital. The digitized data are represented by the unit of ADU.

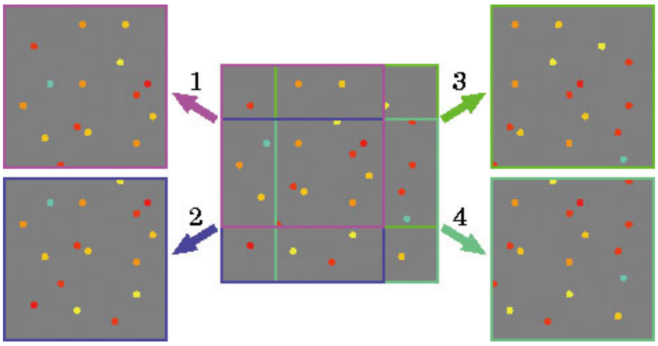


Figure 3: A sketch showing concept of dithering observations.This example shows on how dithering scans work; the four observing fields shown with labels 1 to 4 are combined by shifting to the position(s) of reference star(s) to increase signal-to-noise ratio of the resultant image. In addition, combining all the images without such a position-shifting will give us an image of the back ground emission, i.e., an image free from astronomical objects. (image courtesy: Prof. Iwamuro as the same URL for Figure 2).

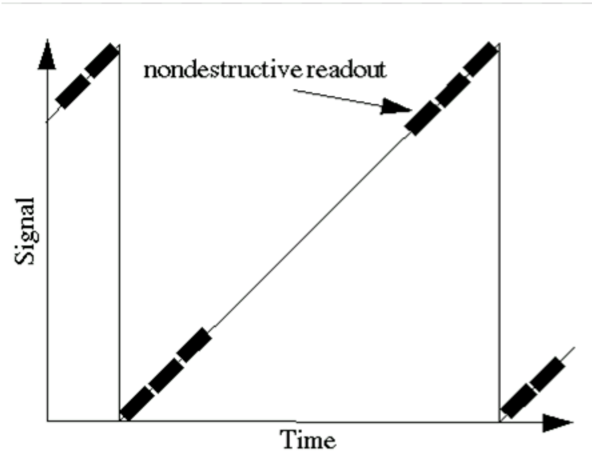


Figure 4: A sketch showing concept of Non-Destructive Readout (NDR). If you adopt the NDR numbers of n, readout will be done n times at the beginning and the ending of the integration.

2.2 Noise Sources in NIR bands and Calculating the Signal-to-Noise Ratio

The following are major noise sources in NIR imaging:

- Background radiation from sky and telescope : SKY [counts pixel^{-1} sec^{-1}]
- Dark current : $DARK$ [counts pixel^{-1} sec^{-1}]
- Readout noise : R [electron pixel^{-1} read^{-1}]
- Flux(es) from the target object : OBJ [counts sec^{-1}]

Given the above noise sources, one can write down signal-to-noise ratio (S/N) for an integration over t [sec] toward an object with a radius of r [pixel] as,

$$S/N = \frac{OBJ \times t \times g}{\sqrt{(OBJ + (SKY + DARK) \times \pi r^2) \times g \times t + \pi r^2 R^2}}$$

where g [electrons ADU^{-1}] is a conversion factor that represents 1 ADU corresponds to how many electrons.

In general, since the background noise dominates the other noise sources, the denominator in the above equation can be approximated as $\sqrt{SKY \times g \times t}$. Namely, the S/N is proportional to $t^{1/2}$, which

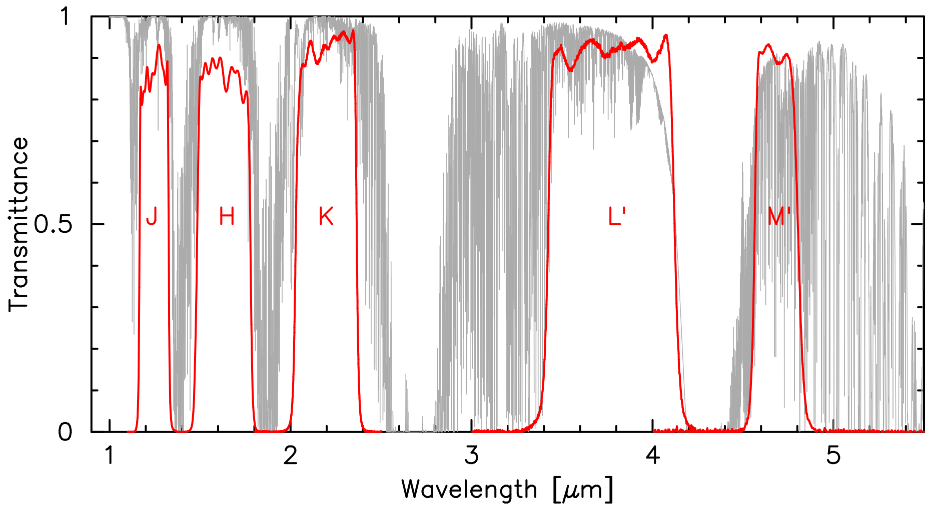


Figure 1: Atmospheric transmission at the summit of Mauna Kea under typical weather conditions (the gray line) and passband profiles in transparency of the broadband filters.

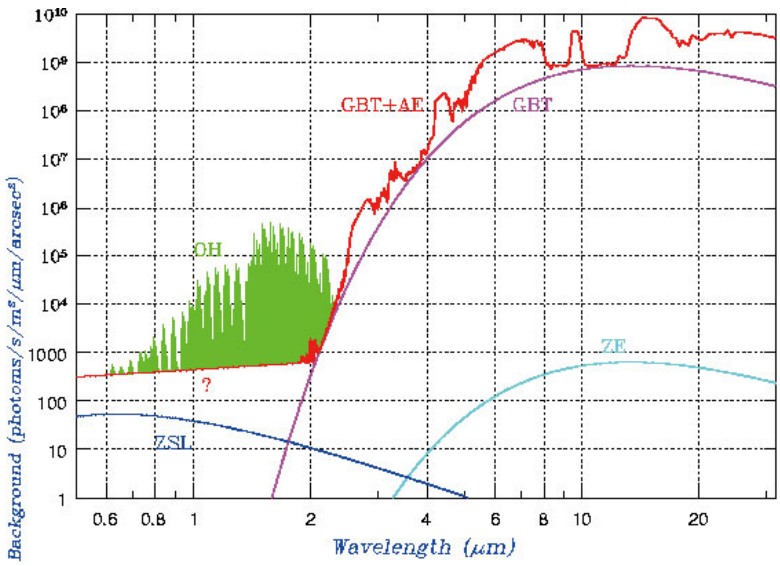


Figure 2: Spectrum of background emission from optical through NIR wavelengths. Note that background level increases toward the long wavelength. OH denotes OH-airglow, AE thermal radiation from the atmosphere ($\sim 273\,\mathrm{K}$), GBT thermal noise from telescope, ZSL scattered light from the Sun due to dust in zodical plane, and ZE thermal emission from such dust. The plot is from the "Astronomy Lecture Notebook" by Prof. Iwamuro at Kyoto University (http://www.kusastro.kyoto-u.ac.jp/˜ iwamuro/LECTURE/OBS/).

is referred to as "background limit". At the short-wavelength such as the J band, readout noise would dominate background noise in the case that integration time is very short, yielding that $S/N(t) \propto t$, instead of $S/N(t) \propto t^{1/2}$.

3 IRCS Imaging Observations: *First Fact*

3.1 IRCS Data

IRCS, Infrared Camera and Spectrograph, is one of the instruments installed at Subaru Telescope aimed for both imaging and spectroscopic observations in NIR. One of the great advantages in usage of the IRCS is a capability of Adaptive Optics (AO) system to minimize effect of atmospheric turbulence, yielding high resolution imaging less than $0\farcs1$. For the imaging mode, IRCS uses InSb detector called Aladdin III array with 1024×1024 pixel; the detector is sensitive at wavelength ranging from 0.9 to 5 μm. You can select a pixel scale of either the high-resolution mode of 0.020 arcsec pixel^{-1}, providing field of view of $20'' \times 20''$, or the low-resolution mode of 0.050 arcsec pixel^{-1}, giving a $50'' \times 50''$ filed of view. As for filters, you can select broadband ones designated for z, J, H, K', K, L', and M' band windows and several narrow-band ones.

Users need to be careful when using archival data as the pixel size scales have changed as of July 2005. IRCS had been operated at the Cassegrain focus since its beginning of the operation. In July 2005, the mounting point of IRCS has changed from the Cassegrain focus to the Nasmyth focus due to upgrades of the AO system, see the below for the detail.

3.1.1 Note for the Cassegrain IRCS data

When you use IRCS data taken at the former Cassegrain focus, keep in mind that the pixel scale of the Cassegrain IRCS data is different from those taken at the current Nasmyth focus. For the former Cassegrain IRCS data, the pixel scales are 0.023 arcsec pixel^{-1} for the high-resolution mode and 0.058 arcsec pixel^{-1} for the low-resolution mode. In many data sets taken with the low-resolution mode, you will see two ghosts lying to the north and west (in the case of P.A. = 0°) toward a bright star you targeted (see Figure 5). One of the ghost is due to plane parallel substrate which is used to split the beam into optical wavelength for the AO system and into infrared wavelength for guiding to the IRCS. The other ghost is known to be produced by the plane parallel substrate which compensates distortion when the gathered lights passes it. These ghosts are seen in the sample data, dealt with this CookBook, taken with the low-resolution mode. On the other hand, such ghosts would not be recognized in the high-resolution mode as long as you configured a bright star to the field center because field of view is enough small to guide the ghosts outside. If you wish to have more information, give a visit to the IRCS web page, `http://www.naoj.org/Observing/Instruments/IRCS/parameters.html`.

In this Cookbook, you should follow the above note, since the sample data in the following section is the Cassegrain IRCS data.

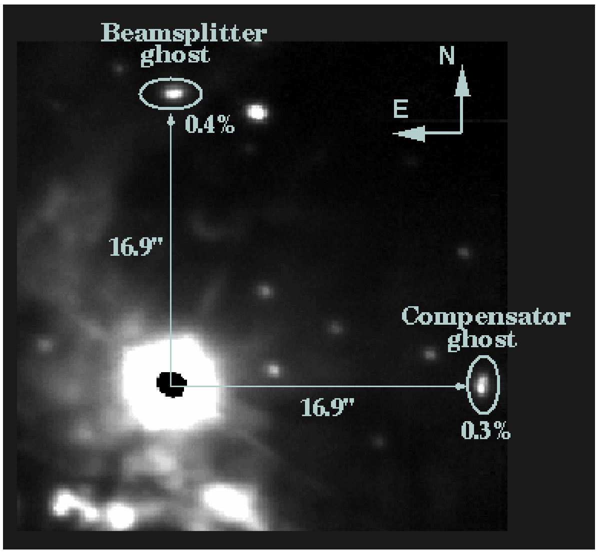


Figure 5: Artificial "ghosts" appeared around a bright star in IRCS images taken with the low-resolution mode. Notice that the "ghosts" were seen in the **old** data (before 2005 July) taken when IRCS was located at the Cassegrain focus.

3.2 A Typical Procedure of Infrared Imaging Observations

A typical procedure of IRCS imaging observations would be;

1. Tracking the target field

2. Configuring a filter and a pixel scale according to scientific requirements

3. Optimizing integration time
   Find the longest integration time that is within the linearity of the detector where the Poisson noise of the background emission dominates the readout noise, i.e., "background limit". Avoid saturating the digital signal by adopting an integration time that does not exceed the linearity range of the detector. In the case of IRCS, such a linearity is safely guaranteed as long as the background level remains below 4000 ADU, see Figure 6.

4. Repeating your integrations by slightly shifting the field ("dithering observation")
   Keep in mind that each set of dithering observation must be taken within a time-scale of background noise level variation; such a time over which you can integrate tends to be short toward long-wavelength. For a typical IRCS observations, we acquire a set of data consisting 5 or 9 images (i.e., pointing centers) with several – a few 10 seconds integrations per image (Figure 7).

5. Don't forget to observe standard stars[1] that are as close as possible to your targeted field.
   Select your standard star(s) accordingly so that it(they) can be observed with airmass close to that

[1] e.g., UKIRT faint standard star catalog (http://www.jach.hawaii.edu/UKIRT/astronomy/calib/phot_cal/faint_stds.html) or that compiled by Persson et al. (1998, AJ, 116, 2475)

for the target fields. Unless you wish to achieve very precise photometry accuracy such as an error of less than 0.1 mag, we believe that an airmass difference of 0.2 – 0.3 would not cause significant difference in the resultant photometry accuracy. Extinction due to airmass appears to be less than 0.1 mag per airmass for typical Mauna Kea weather conditions[2]

6. If you want to change observing mode, return to the Step 2. If you want to observe another source, go to the Step 1.
7. Last, before leaving the summit, don't forget to take data for making flat frames.

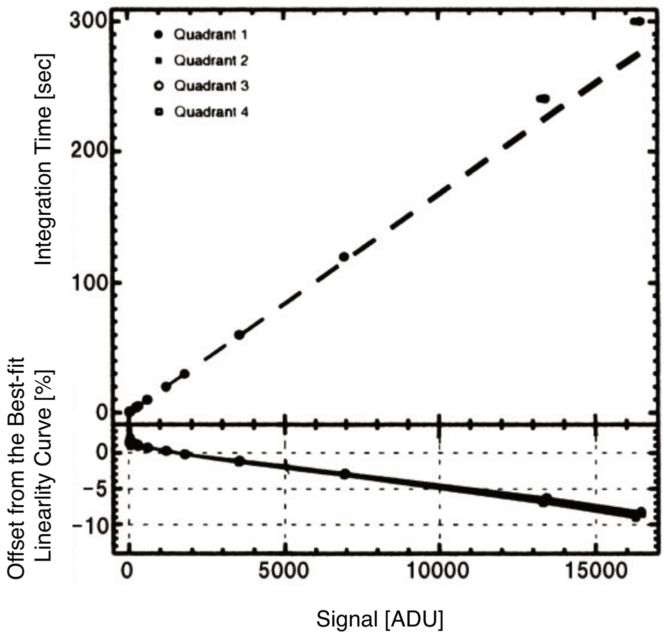


Figure 6: The upper (linear response) panel shows a plot of linearity of the detector at the camera side (Terada et al. 2004, Proc. SPIE, 5942, 1542).As long as the intensity of the signal is around 4000 ADU, the data will be within 3% of the linearity, see the lower panel for offset growth from the non-linearity curve as a function of integration time. To image faint object, adjust integration time to ensure signal of the background does not exceed 4000 ADU. To observe an extremely bright star whose signal level is much higher than that of the background, adjust the integration time so that the peak emission of the target is around 4000 ADU.

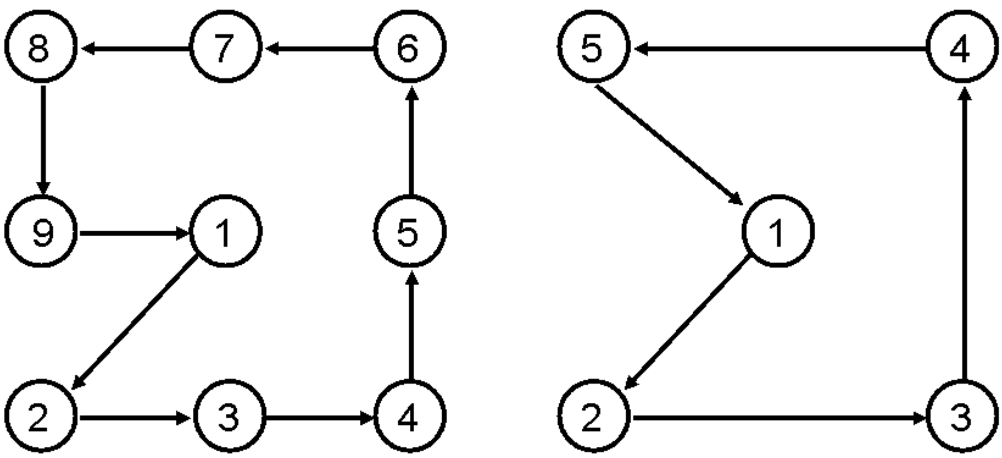


Figure 7: Typical dithering patterns used for IRCS imaging observations where either 5 or 9 fields whose centers are slightly shifted in positions are taken.

[2]http://www.jach.hawaii.edu/UKIRT/astronomy/utils/exts.html

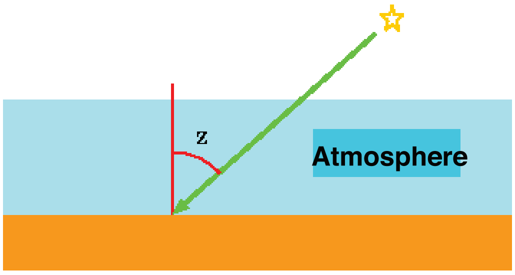


Figure 8: A sketch illustrating a concept of airmass. Airmass is defined by $\sec(z) = 1/\cos(z)$ where z is zenith angle. As is clear from the definition, the minimum of unity is achieved at the zenith. Emission from astronomical objects are absorbed by terrestrial atmosphere as airmass increases, i.e., the degree of absorption is proportional to the path length in the atmosphere. One should plan to observe your sources at an airmass as small as possible no more than ~ 1.5. (image courtesy: Prof. Iwamuro as the same URL for Figure 2).

3.3 A Typical Data Reduction Procedure for IRCS Imaging Observations

As we described in §2.2, raw data — the data without any reduction — contain not only signals from your target but also noises from many sources. In fact, we can write down a count value from a pixel at the position of (x, y) measured at time of t as,

$$RAW(x, y, t) = f(x, y)\,\{OBJ(x, y, t) + SKY(x, y, t)\} + DARK(x, y)$$

where $OBJ(x, y, t)$ denotes signal from the source, $DARK(x, y)$ dark current of the detector, $f(x, y)$ a weighting function that represents sensitivity difference between pixels, and $SKY(x, y, t)$ thermal emission from the sky, telescope as well as the instrument. OBJ and SKY are function of t, which represent time variations of seeing and transparency of the atmosphere as well as that of background radiation. We stress that the SKY term is highly time variable in the NIR regime compared with that in the optical. On the other hand, the contribution from the dark and flat can be considered as constant terms in the case of single-night IRCS Imaging observations. Given the equation above, it is clear that we have to (1) subtract the dark, (2) correct for the $f(x, y)$ term, and (3) subtract sky background in order to extract the desired signal only from your targets. Subsequently, we will combine these images after image shifting.

In the following section, we will describe how to reduce your IRCS Imaging data with the following steps.

1. Inspecting your raw data
2. Making dark frame
3. Making cal-flat frame
4. Making bad-pixel-masking
5. Making sky-flat frame
6. Flat field correcting
7. Making sky frame, and subtracting with the frame
8. Filling pixel values for those eliminated by the bad-pixel mask

9. Measuring position offsets for each image

10. Combining them with image-shifting

11. Analyzing standard star data by applying the above Steps 1 to 10.

12. Estimating the limiting magnitude

4 IRCS Data Analysis

4.1 The Sample Data

In this COOKBOOK, we present a typical procedure on how to reduce Imaging data taken with IRCS by analyzing a sample data set toward distant galaxies. The K' data were taken by one of us toward the blank field in the northern Galactic pole referred to as the Subaru Deep Field (SDF) using IRCS at the Cassegrain focus with the AO 36 (the data have been published in Minowa et al. 2005, ApJ, 629, 29). Although the published image was made from a long exposure over 27 hours, we will reduce a 3 hour portion of the data to illustrate a method for reducing IRCS data. Extract the sample data (`sample1.tar.gz`) and move the `raw/` directory with the following UNIX commands,

```
$ cd [Your_Working_Directory]
$ tar xvzf sample1.tar.gz
$ ls sample1
```

4.2 Preparing for Your Environment

In this COOKBOOK we suppose that a user environment with IRAF version 2.14 or later. If your IRAF version is an older release, some of the tasks used in this COOKBOOK may not work properly.

In addition, the user should install script package (`ircs_imgred`) that was originally developed by us. The following procedure will automatically install the `ircs_imgred` package. After installing, edit the `login.cl` file to customize your environment for reducing IRCS Imaging data.

```
$ cd [Your_Working_Directory]
$ tar xvzf ircs_imgred[mmddyy].tar.gz
$ ln -s ircs_imgred[mmddyy] ircs_red
$ cd ircs_imgred
$ perl setup_ircs.pl
-- creating a new uparm directory
Terminal types: xgterm,xterm,gterm,vt640,vt100,etc.
Enter terminal type: xgterm
A new LOGIN.CL file has been created in the current directory.
You may wish to review and edit this file to change the defaults.
Done
```

Every time when you work on the IRCS imaging data, you should start IRAF in the directory where this `login.cl` file exists, i.e., the `ircs_imgred` directory.

```
$ cd ircs_imgred --- Moving to the directory where login.cl file exists.
$ cl
cl> cd [your working directory]
```

We also need to have an image display program of `ds9`. Type `ds9` at any terminal. If you want to launch the program in xgterm where IRAF is running, simply type **!ds9 &**. Here the UNIX commands followed by `!` mark in IRAF window allows you to issue any UNIX commands, however, for the simplicity, we suggest separating the IRAF command window and the UNIX one.

In this COOKBOOK, we will show all the commands that should be issued on xgterm where IRAF is running with the prompt of

```
cl>
```

, while all the commands issued on the UNIX command line will be shown with

```
$
```

In addition, we are going to use a series of perl scripts. To use them, configure your path to the directory where the scripts are stored, as follows.

```
# If you are using bash --- add the following either in ~/.bashrc or ~/.bash_profile
  export PATH=$PATH:[full-path to the dir. where ircs_imgred are stored]/plscript
# If you are using tcsh --- add the following in ~/.cshrc
  set path ([full-path to the dir. where ircs_imgred are stored]/plscript $path)
```

The perl scripts uses a free software, `SExtractor`, to identify object in the given image (Bertin & Arnouts 1996, A&AS, 117, 393). If you do not have `SExtractor`, get it from `http://terapix.iap.fr/rubrique.p`

4.3 Inspecting "Raw" Data

We suppose that you have your own (perhaps, hand written) logs that were recorded during the observations; the logs will help you associate each observing sequence with its corresponding FITS file numbers. You may want to extract more information from each FITS because you obtained the data from the archive system and/or your logs are incomplete for some reason. The header information written in each FITS will help you in this case. For this purpose, use **imhead** command as follows. You will see something like:

```
cl> imhead sample1/IRCA00091688.fits l+ | page
sample1/IRCA00091688.fits[1024,1024][int]: SDF
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 7412 s.u.
```

```
Created Tue 00:00:00 01-Jan-1980, Last modified Tue 00:00:00 01-Jan-1980
Pixel file "sample1/IRCA00091688.fits" [ok]
FRAME-ID=                      / Frame Id
I_ARCH  =                    0 / ARCHIVED? 0:No 1:Yes
I_HDRVER=                 1.22 / IRCS HEADER VERSION
...
```

Each line in the header tells you name of the variable and its value with a short description. To extract specific information, use the `hselect` command.

```
cl> hselect sample1/IRCA*.fits > file_header.lst
fields to be extracted: $I,title,EXP1TIME,COADDS,NDR,FILTER02,I_MDFMST,I_DTHPOS
boolean expression governing selection: yes
```

The above example extracts several parameters from all the files (as we used "*" wildcard) starting with `IRCA` and ending with `.fits`, then writes the results into a file named `file_header.lst`. The extracted information from the header are the name of the file ($I), object name (title), integration time (EXP1TIME)[3], numbers of COADD (COADDS), numbers of non-destructive reads-out (NDR), filter information (FILTER02)[4], pixel size scale (I_MDFMST) and the dithering position (I_DTHPOS). Please note that some of the above variables, i.e., the header keywords, are designated for IRCS only, and may not be used for the other instruments at Subaru. The first column of the obtained file of `file_header.lst` shows you information for each FITS data sorted by the order given by `fields to be extracted`. Check these info., e.g., those for your target source(s), standard star(s), and dark and cal-flat frames. It is probably wise to delete row(s) corresponding to the unnecessary frames e.g., those taken for "check field", and having some errors in observations, through a comparison with your (hand written) observing log.

In the following example, we will analyze data referred to as 'SDF' for scientific target field and 'P330-E' for a standard star. Looking for rows containing these object names, you will find something like,

```
sample1/IRCA00091688.fits   SDF     120.0000    1   8   K   58MAS   1
sample1/IRCA00091860.fits   P330-E    3.0000    1  11   K   58MAS   1
```

The first line says that an object frame whose FITS file name is IRCA00091688.fits was obtained with a 120 second integration by setting COADD = 1 and NDR =8. Similarly, for line 2 that the standard star was taken with a 3 second exposure by setting COADD =1 and NDR = 11. The last 3 columns show the identical information that both of them are the first data in the dithering scans and are taken with the former 58 milli-arcsecond (mas) per pixel mode[5] at K-band.

[3]You will see that there are two variables of EXP1TIME and EXPTIME to describe exposure time, and they are related with EXPTIME=EXP1TIME×COADDS.

[4]IRCS equips 3 filter wheels; the broadband filters for the JHKLM bands are stored in the second filter wheel (FILTER02), the narrow band filters are in the first one (FILTER01), and the z band filter is in the third one (FILTER03). If you have used the z filter or one of the narrow band filters, we suggest checking these parameters as well.

[5]This low-resolution mode had been used when IRCS was at the Cassegrain focus.

After getting the minimum information, let us start our analysis by making a list of object fields, that's a pretty easy business with grep command,

```
$ grep SDF file_header.lst | awk '{print $1}' > object_raw.lst
```

By inspecting the images, make sure that the peak count values of your target(s) and typical values of the background emission in each frame are within the range where the linearity is guaranteed (see Figure 6). If you find frame(s) which contains count values beyond the linearity range, you definitely MUST get rid of them simply because such pixels lead to wrong results in Imaging. You should double-check that the sample data have sufficiently small count values around 4000 ADU for the background emission because the targets are the very faint galaxies in the deep space. This can be done by inspecting results from `imexam` command used for an image displayed with `ds9`. Type "`z`" at any position within the `ds9` window. You will get count values surrounding the pointed pixel. If you type "`m`", you will get statistical values charactering these pixels. For instance, display the image of sample1/IRCA00091689.fits to `ds9` using `imexam` command. Type "`m`" by putting your cursor around the position of (500, 280), you would get results similar to the below.

```
#            SECTION      NPIX     MEAN   MEDIAN   STDDEV      MIN      MAX
   [503:507,278:282]        25   35261.   35308.     576.   34176.   36276.
```

This example shows that the median is 35308, but we have to divide this by numbers of NDR. Moreover, if you used very short integration times, you may have COADDed the data. If this is the case, you have to divide them by the number of COADDs to check the count level. In the above example, we estimate the background level of $\sim$ 4400 ADU by dividing 8 which is the product of COADD and NDR numbers. You will see that inferred value securely falls in the range where the read-out signal increases with the integration time (see Figure 6). If you wish to obtain rough estimates of background emission for all the frames, use `imstat` commands as follows,

```
cl> imstat @object_raw.lst fields=image,midpt nclip=50 usigma=3 lsigma=3
```

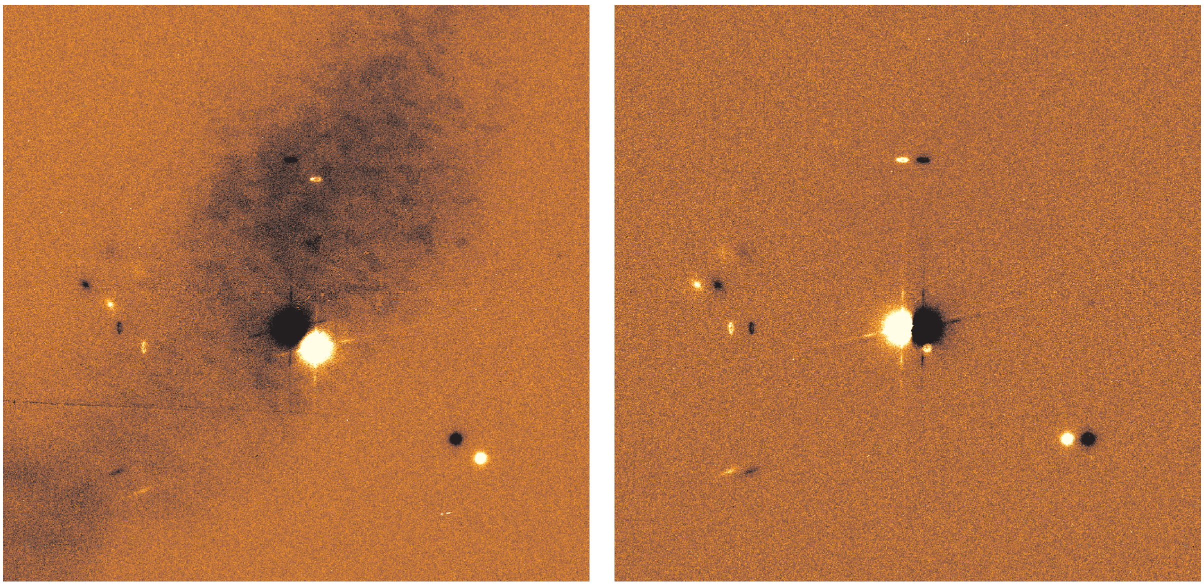
Subsequently, we should inspect all the images by eye. Figure 9 shows an example of bad image whose overall pattern obviously differs from the others. Such an artifact is known to appear in the frames immediately after starting observations or ending long-exposure observations — mostly seen in the first frame of dithering observation (sometimes in the second frame). In addition, we suggest removing identification numbers of the other bad frames due to e.g., instrumental troubles such as a tracking failure as well as those taken under poor weather conditions. To see if your data set includes such bad frame(s), it is wise to identify bad frames by subtracting or dividing two frames that have been taken at slightly different dithering positions. Remove these bad frame numbers from the list because they will degrade the S/N of the final image. Let us suppose that you have 9 images from #1 to #9 in a set of dithering data, you should verify these images by making sequential set of images that are obtained by subtracting the next image, i.e., #1–#2, #2–#3,...., and #8–#9, as well as those divided, i.e., #1/#2, #2/#3, ... and #8/#9. You can use one of our IRCS scripts named `frcheck` as shown below. Before running `frcheck`, don't forget to start `ds9`.

```
cl> frcheck object_raw.lst mklist=yes calc=subtract
```

You will see the subtracted image (or divided images in the case of division) on the ds9 window. Type 'n' on the ds9 window to proceed the next frame, and type 'q' once you have verified all the images. After quitting the inspection, you will see a list of frame names as shown below (probably you have to edit it with vi editor). Delete all the frame numbers of the bad frames. If you find frames that have odd/peculiar dark pattern, type 'dd' at each of the line to delete them (here we assume that you are using vi editor).

```
sample1/IRCA00091688.fits
sample1/IRCA00091689.fits
sample1/IRCA00091690.fits
sample1/IRCA00091691.fits
sample1/IRCA00091692.fits
  ...
```

Once you have finished deleting bad frames, exit from the editing mode by typing ':wq'. Inspect the resultant list of object_raw.lst-b. If you don't have further bad frames, the program will make a list file with '-b' by copying the original one. We show an example of subtraction between bad frames in Figure 9 where you should recognize a "hole" around the center due to difference in dark frames.



(a) Subtracted image between a usable frame and a non-usable frame.

(b) Subtracted images between two frames without problems

Figure 9: An example of bad (=non-usable) frame that was found by subtraction with the next one

The sample data have been taken with a set of 9 dithers toward the object. Since it is useful to have lists for individual sets of dithering sequences for making a sky frame, we should make a list by extracting dithering positions in the header. The `mksubset` command in ircs_red will make such a list for each dithering set.

```
cl> mksubset object_raw.lst-b object_raw
```

This will make 9 files whose names are object_raw[01-09].lst.

You should see that one of the created files, e.g., object_raw01.lst, contain names of 9 files. Display these files with `ds9`.

```
cl> !ds9 &
cl> imexam @object_raw01.lst
```

This will sequentially show the listed images, you can examine the next frame by typing 'n' on ds9. If you want to return to the previous one, type 'p'. Comparing the 9 frames, we hope that you will understand how IRCS does dithering scans as shown in Figure 7.

4.4 Making Dark Frame

We suppose that you have taken dark frames with exactly the same integration time, number of COADD and number of NDR as those for scientific exposures. These images will be used to make a dark frame using the following method.

1. Making a list of dark frames whose integration time and NDR parameters (integration time of 120 sec and NDR = 8 for the sample data) are identical to the object frames.

   ```
   $ grep DARK file_header.lst | awk '{if($3==120 && $5==8) print $1}'  > dark.lst
   ```

2. Verify the quality of these frames by making subtracted or divided images with those taken immediately before and/or after the frame. If you find such bad frames, remove them from the list (but, there are no bad frames in the sample data). This inspection can be done using the `frcheck` script described in the previous subsection.

   ```
   cl> frcheck dark.lst mklist=yes calc=subtract
   ```

3. Combine all the dark images with median; we should use the one listed in `object_dark.lst-b`. For this purpose, we suggest using `mkdark` in the ircs_imgred package. The script clips pixels whose intensities exceed $\pm 3\sigma$ level of each image.

   ```
   cl> mkdark @dark.lst-b  dark.fits
   ```

Notice that we will not use the dark frames made at this step for analysis of object frames. This is because the dark counts in IRCS data are intrinsically small enough to neglect. Another reason is that we subtract the sky-frames from the object ones, thus the contribution from the dark, even if it exists, will be removed by this subtraction. Instead, these dark frames will be used to make the bad pixel mask described in §4.6.

4.5 Making Calflat Frames

A flat frame is generally obtained by dividing pixel count values at each pixel read by the detector when it sees uniform incident light. Namely, an ideal detector without inequity of sensitivity differences between the detector pixels would have pixel value of unity entire the frame. In practice, pixels that have better sensitivity will show count values higher than 1.0, whereas worse pixels less than 1.0. With this reasoning, we can correct for sensitivity differences over the detector by dividing the flat frame.

For the specific case of IRCS imaging, we usually make such a frame from images that have been taken looking of uniform incident light. We refer to such frames as "Calflat". Many observers take calflat images by observing lamp after observations by configuring all the instrumental parameters (e.g., filter, pixel size scale etc) to be identical to those used in the scientific observations. It should be noted that one must obtain calflat frames with (i.e., ON) and without (OFF) lamp light because the ON frame has not only the incident uniform light but also the dark current and the thermal emission from the telescope and the instrument. On the other hand, OFF frames consist of dark current and thermal emission. We therefore can obtain a frame that have sole the uniform light by making ON – OFF image. Below is an example how the above procedure works on the sample data.

1. Extracting file names for calflat from ' file_header.lst'. Calflat ON and OFF images can be identified with object names of "IMAGE_Kp_ON" and "IMAGE_Kp_OFF", respectively.

   ```
   $ grep IMAGE_Kp_ON file_header.lst | awk '{print $1}' >  calflat_on.lst
   $ grep IMAGE_Kp_OFF file_header.lst | awk '{print $1}' > calflat_off.lst
   ```

2. Make sure whether or not each ON and OFF frame is OK. If you find bad frames, remove their names from the list (the sample data consist of bad frames of IRCA00091869, and IRCA00091869; you are advised to understand how bad they are.).

   ```
   cl> frcheck calflat_on.lst mklist=yes calc=subtract
   cl> frcheck calflat_off.lst mklist=yes calc=subtract
   ```

3. Making an ON – OFF frame that should contain solely the uniform light from the lamp. This will be obtained by subtracting an OFF frame that is combined with median for OFF frames from EACH ON frame. Subsequently we have to divide each pixel value of ON–OFF frames by mean value of each pixel before combining with median to make a normalized calflat frame. All the procedures can be done with `mkcalflat` in ircs_imgred.

   ```
   cl> mkcalflat @calflat_on.lst-b @calflat_off.lst-b calflat.fits mask=none
   ```

4.6 Making Bad Pixel Mask

In general, infrared detectors tend to have larger number of bad pixel than those in CCDs used in optical wavelength. Such bad pixels are mostly due to those

- less sensitive pixels than the average, and
- having elevated dark current

These bad pixels must be removed before further data reduction steps. You can remove bad pixels with `fixpix` command in IRAF; after removing bad pixels, the command interpolates pixel values using those from the adjacent ones. For this purpose, we have to supply a bad pixel mask that has count values of unity (or any of positive integer) for bad pixels and zero for good pixels. Once you have made a bad pixel mask for your observing run, you don't need to generate new masks as you can re-use the mask for each observing date. You can assume for almost all the cases that detector conditions will not change significantly during observations.

Bad pixel mask can be generated in the following way using the dark and calflat frames that we made in the previous. First, you have to determine threshold value using a histogram of the pixel count values (see Figure 10) in order to eliminate those showing elevated values. In the sample data, we suppose that pixel values exceeding 2500 ADU are bad. Second, we should find a threshold value for the less-sensitive pixels using a similar histogram shown in Figure 11; we selected ADU of 0.3 as the cutoff value for the sample data. Given these upper and lower cutoff levels, let us make a bad pixel mask with `mkbpmask` of the ircs_imgred package in the following fashion.

```
cl> mkbpmask dark.fits calflat.fits ircsimg_bpmask.fits
Showing histogram of dark.fits
Input threshold for high dark current pix mask:
2500
Showing histogram of calflat.fits
Input threshold for low sensitivity pix mask:
0.3
```

This should give you the desired bad pixel mask shown in Figure 12.

4.7 Making Sky Flat

The standard method to make a skyflat is to combine frames taken with dithering observations and then divide them by the mean values of the individual frames. Since the object positions in the dithered frames are not identical in each frame, your object image will be canceled out by combining with the median, producing a sky background image multiplied by inequity of sensitivity of the pixels. Since the sky background can be considered to be uniform in the combined images, we obtain a flat frame that contains only information of the inequity of the pixel sensitivities. Keep in mind that, given the above method, you have to have enough numbers of frames to make sky background. The most straightforward

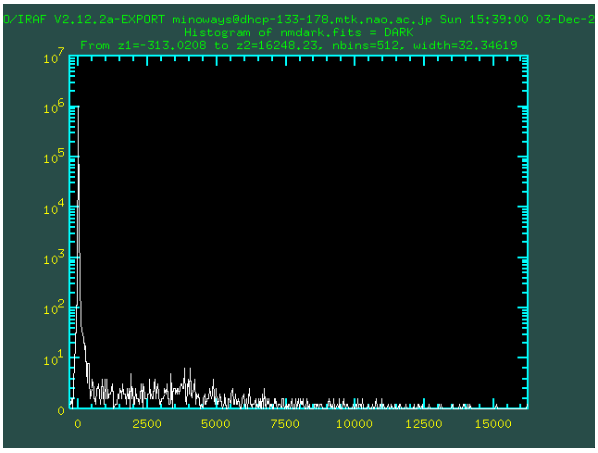


Figure 10: Histogram of pixel count values for dark frame. The horizontal and vertical axes are count values in ADU unit and numbers, respectively.

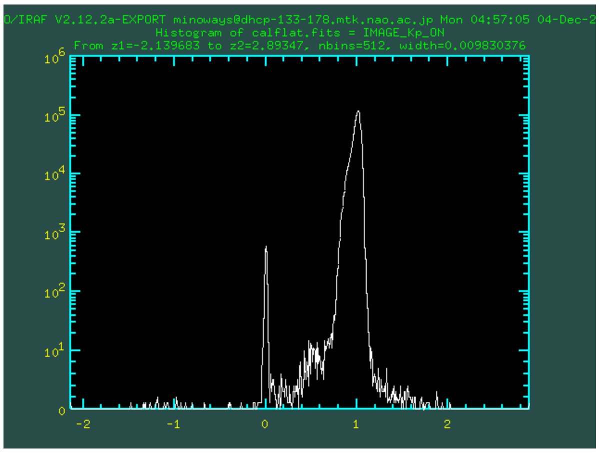


Figure 11: Histogram of pixel count values for flat frame shown in the same fashion as for Figure 10

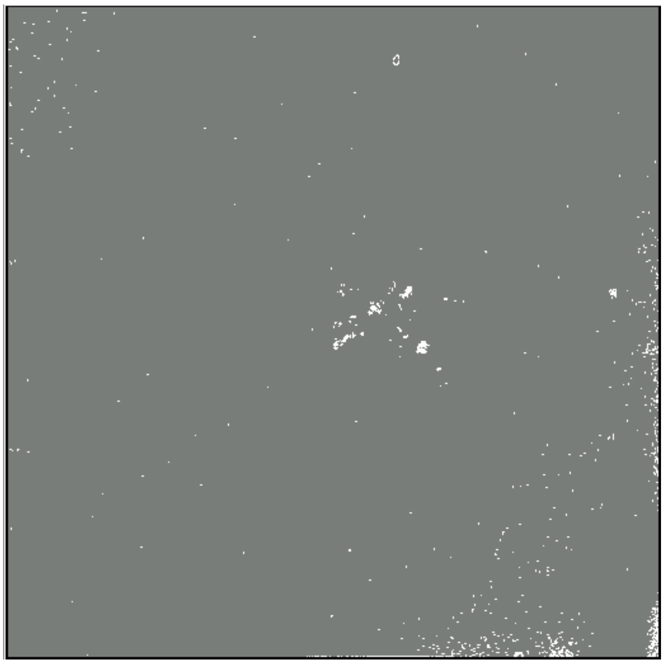


Figure 12: The resultant bad pixel mask that presents locations of bad pixels of the IRCS detector.

method is to divide each of them by the mean pixel counts, then combine them with median. However, this method may not work for the objects that are more extended than the dithering width. This case requires an extra step of masking before combining with median.

The procedure to make skyflat is summarized as follows,

1. Dividing raw data of object frames by mean value of pixel count of each frame
2. Combining the normalized object frames from above to make a tentative flat frame
3. Dividing the raw object frames by the tentative flat frame
4. Applying bad pixel mask made in §4.6
5. Selecting region(s) in which to mask out bright source(s) in each frame (see Figure 13)
6. Applying the bad pixel mask created previously to the raw object frames, then calculating the pixel mean values in order to divide the frames by the mean values
7. Making the desired skyflat by combining the masked and normalized frames created in the previous steps

The above procedures from #1 to #7 can be done with `mkskyflat` in ircs_imgred.

```
# Making a directly where a list of object mask and date will be stored
$ mkdir objmask
$ sed -e "s/sample1/objmask/" object_raw.lst-b > object_mask.lst
cl> epar mkskyflat
                                    I R A F
                     Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = mkskyflat

objlist =    @object_raw.lst-b  List of raw object frames with @mark
output  =  object_skyflat.fits  Output sky flat frame
masklist=     @object_mask.lst  Output object mask list with @mark
(bpmask =                  yes) Fix bad pixels?
(bpfile =  ircsimg_bpmask.fits) Name of bad pixel mask frame
(iternum=                   50) Number of clipping iterations
(nclip  =                    3) N-sigma rejection limit
(conv   =            block 3 3) Convolution kernel(objmask)
(hsig   =                   2.) Sigma threshold above sky (objmask)
(lsig   =                   10) Sigma threshold below sky (objmask)
(minpixe=                   50) Minimum number of pixels in detected objects(obj
```

```
(list1  =                    )
(list2  =                    )
(mode   =                  ql)
# :g
```

:g will execute the IRAF task. The mkskyflat script uses the objmask command implemented in the noao library in IRAF. First, this task makes a smoothed image by convolving some window function to include all adjacent pixels; the above example uses the 3×3 block average filter (conv=block 3 3). Second, using the smoothed image, the task makes a mask (i.e., a masking image) whose pixel values will be set to 1 for the pixels (in the original image) exceeding the given threshold and 0 for all others. The determination of threshold values are sort of empirical; we suggest masking pixels with counts higher than the 2σ level AND having > 50 pixel extension for detected objects. Of course, you should optimize these parameters for each data set by trial and error. To verify the results, use check_objmask in the ircs_imgred package.

```
cl> check_objmask object_raw.lst-b object_mask.lst
add sample1/IRCA00091689.fits,objmask = objmask/IRCA00091689.fits[pl]
sample1/IRCA00091689.fits updated

### sample1/IRCA00091689.fits with object mask objmask/IRCA00091689.fits
### Press [return]

z1=30784.63 z2=39642.23
z1=30784.63 z2=39642.23
...
```

4.8 Correction for Inequity of Pixel Sensitivities

Inequity of pixel sensitivity (i.e., pixel-to-pixel sensitivity variation) can be corrected by dividing object frames by the flat-flame that we have made so far. Our experience suggests that better results can be obtained by dividing using the skyflat frame rather than the calflat frame. Figure 14 compares the resultant images obtained through dividing using the calflat (left) and skyflat (right) frames where a sort of fringe pattern can be recognized in the former. We speculate that the difference can be attributed to how they were taken. Calflat is obtained by observing the lamp located close to the Cassegrain focus (currently at the Nasmyth focus), namely, the calflat light is not reflected off the main and secondary mirrors. Therefore the optical path is different from that used for astronomical observations. Moreover, in most cases, Calflat data are typically acquired using 0.5 second integration which is quite a bit shorter than than those used for observations. We favor this to be the cause of the most significant recognizable fringe pattern around the center is mainly caused by the integration time difference. To the best of our knowledge, we have never seen such an artifact in skyflat frames, and therefore suggest using skyflat

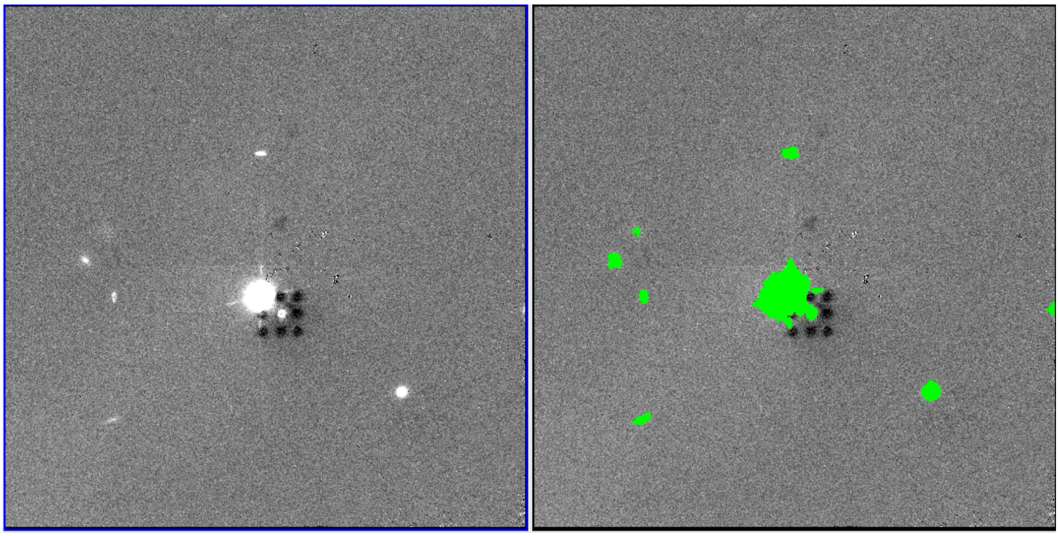


Figure 13: An example that shows how object masking works. The left and right panels, respectively, show before and after applying the object masking, made through `objmasks`, coded by green.

frames. If your image is not the "background limited" (see §2.2) (e.g., in the case of narrow band imaging or very short exposures of a bright star), we suggest using the calflat frames to correct for the inequity of pixel sensitivities.

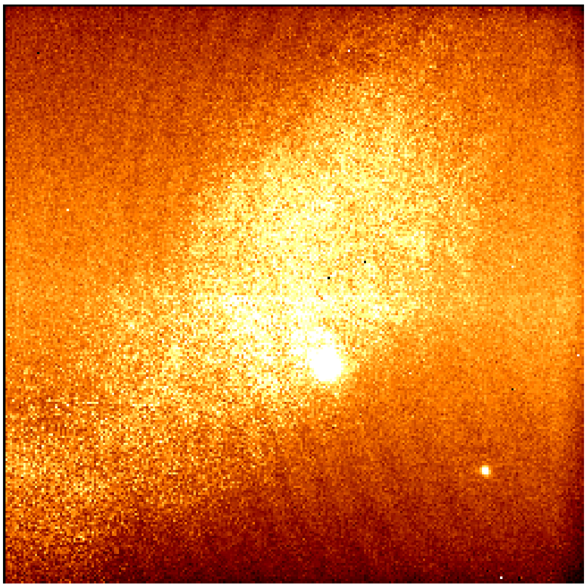
Keep in mind that, if the raw data frame contain any extended emission over the field of view, you cannot use those raw data frames to make a skyflat frames (this is not the case for the sample data). Many observers who image extended emission usually take blank sky images. If for some reason, you have not taken blank sky data, you will have to use calflats to flatten your data. Although we warned against using calflat data, the artifact seen in the calflat frame would disappear to some extent at a later stage of skyflat subtraction (§4.9) since the artifact is stable with time.

In practice, the correction is pretty straightforward; you just divide the object frames by the skyflat frames using `flatfield` command in ircs_imgred. This command normalizes each frame by dividing the product of the COADDS, NDR, and EXPTIME of the frame. After dividing by COADDS×NDR×EXPTIME, the unit of your image should be ADU sec^{-1}.

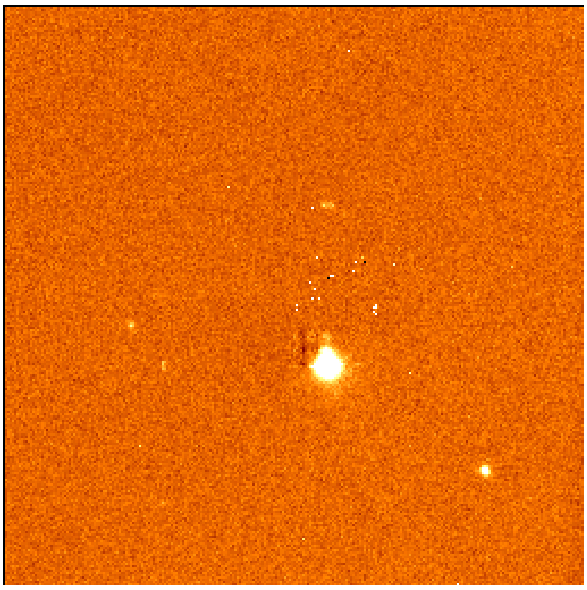
```
# Making a directory to store the output frames and their list
$ mkdir ff
$ sed -e "s/sample1/ff/" object_raw.lst-b > object_ff.lst
cl> epar flatfield
                                    I R A F
                    Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = flatfield
```

```
rawlist =    @object_raw.lst-b  Input raw data list with @mark
outlist =        @object_ff.lst  Output data list with @mark
flatimg =   object_skyflat.fits  Flat image name
(subdark=                     no) Subtract dark frame?
(darkimg=                       ) Dark frame normalized by COADDS*NDR
(bpfix   =                   yes) Fix badpixel?
(bpmask  =   ircsimg_bpmask.fits) Badpixel mask
(list1   =                      )
(list2   =                      )
(list3   =                      )
(mode    =                    ql)
# :g
```

Here, ":g" in the last line will execute the called task in IRAF.



(a) Produced with Calflat



(b) Produced with Skyflat

Figure 14: A comparison of frames to be used for pixel-sensitivity correction. The left hand frame is made with calflat, while the right hand frame with skyflat. In the left panel, a circular fringe pattern and a "dark pattern" which is seen in Figure 9a can be recognized.

4.9 Subtraction of Sky-background and Combining Multiple Images

Sky frames are usually generated by combining object frames that are corrected for pixel-sensitivity inequity. Keep in mind that sky background level is pretty high in NIR and highly time variable. We, therefore, have to make sky-frames from image data that have been taken as close as possible to the frame that is to be sky-subtracted. However, you may encounter a dilemma that the S/N ratio of the sky-background tends to be low if you are limited to only using closely taken frames. We suggest making a sky-frame for each dithering set that consists of 5 – 9 object frames. To make a sky frame, combine all the dithered images in a set with median. Before combining all the dithered images within a dither set, it is highly probable that bright objects will need to be masked (§4.7), Since each dither set consists of rather small numbers of the usable images compared to the case of making skyflat (§4.7), pixel values nearby the masked pixels would have different noise level compared to those of surrounding areas. To avoid this effect, we should interpolate pixel values for masked pixels from the combined ones.

1. Making tentative sky-subtracted frames using the tentative object-mask that was created to perform sky-flat

   We should select bright objects that will be masked. The sky frame will be created by combining all the images taken in each dithering set, and will be scaled to match the median of pixel counts in each flat-field image. This will be subtracted from flat filed image. This will be done using `maskskysub` in the ircs_imgred package.

   ```
   # Making a directory where we will store sky-subtracted frames and a list of them
   $ sed -e "s/ff/sstmp/" object_ff.lst > object_sstmp.lst
   $ mkdir sstmp
   cl> epar maskskysub
                       Image Reduction and Analysis Facility
   PACKAGE = ircs_imgred
      TASK = maskskysub

   fflist  =       @object_ff.lst  List of flat fielded frames with @mark
   sslist  =    @object_sstmp.lst  List of sky subtracted frames with @mark
   masklist=     @object_mask.lst  List of object mask with @mark
   (ndthpos=                    9) Number of dithering position
   (iteranu=                   50) Max iteration number for statistic
   (nsigma =                    3) Sigma rejection limit
   (list1  =                     )
   (list2  =                     )
   (list3  =                     )
   (mode   =                   ql)
   #:g execute the task
   ```

2. Measuring the position offsets of the tentatively sky-subtracted frames with respect to the position of the reference star in the reference frame

   We suggest selecting a reference frame from the first image of the dither pattern (corresponding to the IRCA00091697 frame in the sample data). The offset values can be estimated from the peak value of the cross-correlation coefficient between the reference and the frames that you want to position-shift. This can calculate with the `calcshift` command implemented in ircs_imgred. To calculate cross-correlations between two arbitrary images, it is ideal to have as many bright sources in the images as possible. However, like the sample data where we only have a bright, but a saturated star and faint galaxies, we should not to try to calculate cross-correlation over the whole image, but utilize an area around a galaxy as shown in Figure 15. We will use a 100 pixel × 100 pixel region centered on the galaxy because the sample data taken by dithering observations over a 4 arcsec × 4 arcsec (corresponding to 70 pixel × 70 pixel) region. Here our experience suggests using a factor of ∼ 1.5 times larger area to calculate such a cross-correlation compared to the dithered area size.

```
cl> epar calcshift
                                    I R A F
                     Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = calcshift

objlist =    @object_sstmp.lst  List of input images with @mark
output  =    object_offset.dat  Name of output offset data
(referen= sstmp/IRCA00091697.fits ) Reference image
(region =    [144:243,441:540]) Image region for cross-correlation
(crmargi=                    0) Margin to strip down (crossdriz)
(crtaper=                    1) Edge region to taper (crossdriz)
(crboxsi=                 100.) Box size where to search for peak (shiftfind)
(mode   =                   ql)
```

3. Combining the tentative sky-subtracted frames by the median averaging mode after position-shifting

   To combine images, we have to add (or subtract) offset values so that median of the sky level becomes zero. This procedure can be done with `imshiftcomb` in ircs_imgred which allows shifting of pixel positions with sub-order of pixel precision (i.e., smaller than the pixel size).

```
cl> epar imshiftcomb
                                    I R A F
                     Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = imshiftcomb

input   =    @object_sstmp.lst  List of input images with @mark
output  =  tmp_objcombine.fits  Output combined image
```

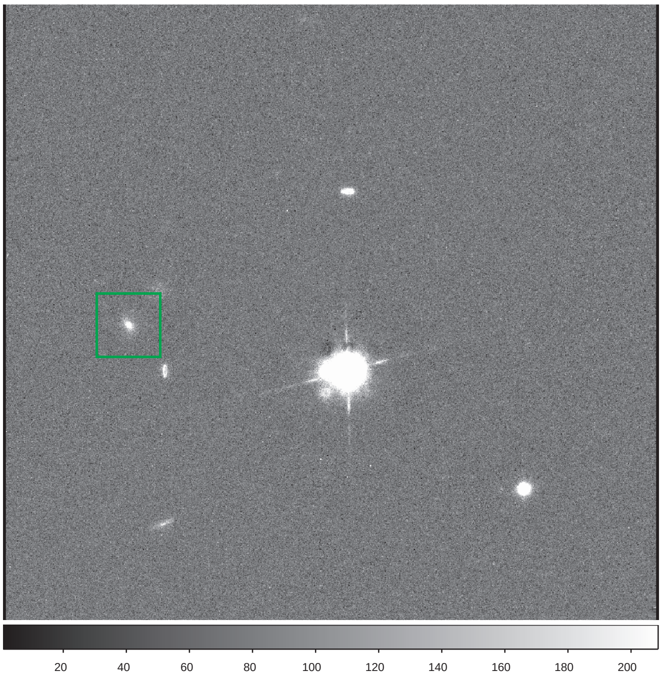


Figure 15: An example of selecting a region over which cross-correlation will be calculated to determine positional offset. Use a region containing as bright as possible stars and as many stars as possible. However, in the specific case of this sample image, we will use the green-boxed galaxy as it is the only non-saturated object.

```
offset  =    object_offset.dat  Input image offsets
(expw   =                   no) Create exposure map?
(expmap =                     ) Name of exposure map
(imgsize=                 1536) Print image size
(list1  =                     )
(list2  =                     )
(mode   =                   ql)
#:g executes the task.
```

This command attempts to copy the original image to a frame that has a dimension defined by `imgsize`, then "subpixel-shit" (i.e., shifting an image with an accuracy less than a pixel size) the copied image into a newly defined frame. In order to accommodate the entire region of the original image, the new frame must have a dimension larger than the original. In the above example, the original image has 1024×1024 pixels, whereas we set the dimension of the new image at 1536×1536 pixels using this reasoning. The `imgsize` parameter must be selected by considering the dithering width so that the whole region of the "subpixel-shifted" image is stored in the frame defined by the `imgsize`.

4. Selecting bright star(s) in the overlaid image to determine circular regions to be interpolated; such a circle will be defined by the pixel coordinate of the center and the radius The perl script of `objaperture.pl` will help you to determine such regions to be interpolated.

```
$ objaperture.pl tmp_objcombine.fits obj_aperture.dat
```

This command calls the program of `SExtractor` to detect stars with its default threshold values that can be changed later according to your choice. The selected interpolation regions can be checked by `showap` in ircs_imgred (see Figure 16).

```
cl> showap tmp_objcombine.fits obj_aperture.dat
```

Once you have executed the `objaperture.pl` script, it generates an input parameter file named `objeaperture_pl.sex`. If you wish to modify the parameters, edit the file. See the SExtractor COOKBOOK for the details. If you are happy with the new parameters, execute `objeaperture.pl` script to select regions to be interpolated.

```
$ objaperture.pl tmp_objcombine.fits obj_aperture.dat -c objeaperture_pl.sex
```

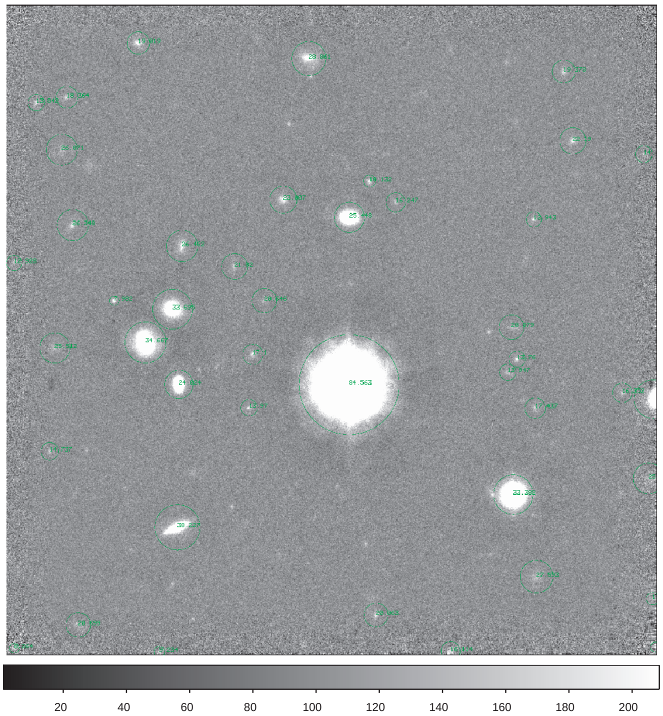


Figure 16: An example image that has a bright star whose pixels should be masked and interpolated from the surroundings.

The next step is to shift the interpolated regions with `shift_aperture.pl` so that they will be matched to the frames before the combining. The offsets required for the position-shifting are taken from those calculated for the tentative combining, i.e., `offset.dat`.

```
$ mkdir objarea
$ sed -e "s/ff/objarea/" -e "s/fits/dat/" object_ff.lst > object_area.lst
$ shift_aperture.pl obj_aperture.dat object_offset.dat object_area.lst
```

5. Interpolating pixels surrounding the bright star selected above
   Using the `eraseobj` command in ircs_imgred, this interpolation should be done to the frames whose pixel-inequity have been corrected.

```
# Making a directory where we will store the resultant frames and a list of them
$ mkdir ff-obj
        $ sed -e "s/ff/ff-obj/" object_ff.lst > object_ff-obj.lst
cl> epar eraseobj
                                   I R A F
                    Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
```

```
   TASK = eraseobj

inlist  =       @object_ff.lst  Input file list name with @mark
masklist=     @object_area.lst  Mask coordinate data file list with @mark
outlist =   @object_ff-obj.lst  Output file list name with @mark
(minsize=                    1) Minimum mask aperute size
(list1  =                     )
(list2  =                     )
(list3  =                     )
(list4  =                     )
(mode   =                   ql)
# :g executes the task
```

6. Subtracting sky-background level using sky-frames made for each dithering set
   Of course, the sky frame must be created from frames whose bright stars have been masked and interpolated, as described above. Before subtracting, we have to normalize each frame so that their median values match each other (see Step. 1). All the procedure can be done with `eraseskysub` in ircs_imgred.

```
# Making a directory where we store the sky-subtracted frames and a list of them
$ sed -e "s/ff/ss/" object_ff.lst > object_ss.lst
$ mkdir ss
cl> epar eraseskysub
                                        I R A F
                          Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = eraseskysub

fflst   =       @object_ff.lst  Flat fielded image list name with @mark
ffobjlst=   @object_ff-obj.lst  Object subtracted ff image list name with @mark
sslst   =       @object_ss.lst  Sky subtracted image list name with @mark
(ndthpos=                    9) Number of dithering position
(iteranu=                   50) Max iteration number for statistic
(nsigma =                    3) Sigma rejection limit
(list1  =                     )
(list2  =                     )
(list3  =                     )
(mode   =                   ql)
# :g executes the task
```

7. Measuring position offsets of the dithered frames with respected to the reference frame
   Here, we assume that you have selected the first dither-position image of the dithering sequence as the reference frame. This is done with `calshift` just as we did with the tentative sky-subtracted image (see the item 2).

```
cl> epar calcshift
                                            I R A F
                         Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = calcshift

objlist =    @object_ss.lst  List of input images with @mark
output  =    object_offset_final.dat  Name of output offset data
(referen= ss/IRCA00091697.fits ) Reference image
(region =    [144:243,441:540]) Image region for cross-correlation
(crmargi=                   0) Margin to strip down (crossdriz)
(crtaper=                   1) Edge region to taper (crossdriz)
(crboxsi=                100.) Box size where to search for peak (shiftfind)
(mode   =                  ql)
#:g executes the task
```

8. Making an image of the object by combining images (with averaging, instead of the "median filtering") whose sky-background have been subtracted; this is the desired final image (Figure 17)
   Before combining, we have to add/subtract an offset so that the skybackgroud level becomes zero. When combining the images, we suggest multiplying a weighting function inferred from integration times. The `imshiftcomb` command handles this task. We suggest keeping `expw = yes` to make an integration-time map (see Figure 17) that will be used not only for estimating the S/N ratio but also for calculating the weighting function. The obtained weighting function image will be useful if you want to use the automatic object finding programs (such as SExtractor).

```
cl> epar imshiftcomb
                                            I R A F
                         Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = imshiftcomb

input   =    @object_ss.lst     List of input images with @mark
```

```
output  =    object_final.fits  Output combined image
offset  = object_offset_final.dat  Input image offsets
(expw   =                   yes) Create exposure map?
(expmap =  object_expmap.fits ) Name of exposure map
(imgsize=                  1536) Print image size
(list1  =                      )
(list2  =                      )
(mode   =                    ql)
#:g executes the task
```

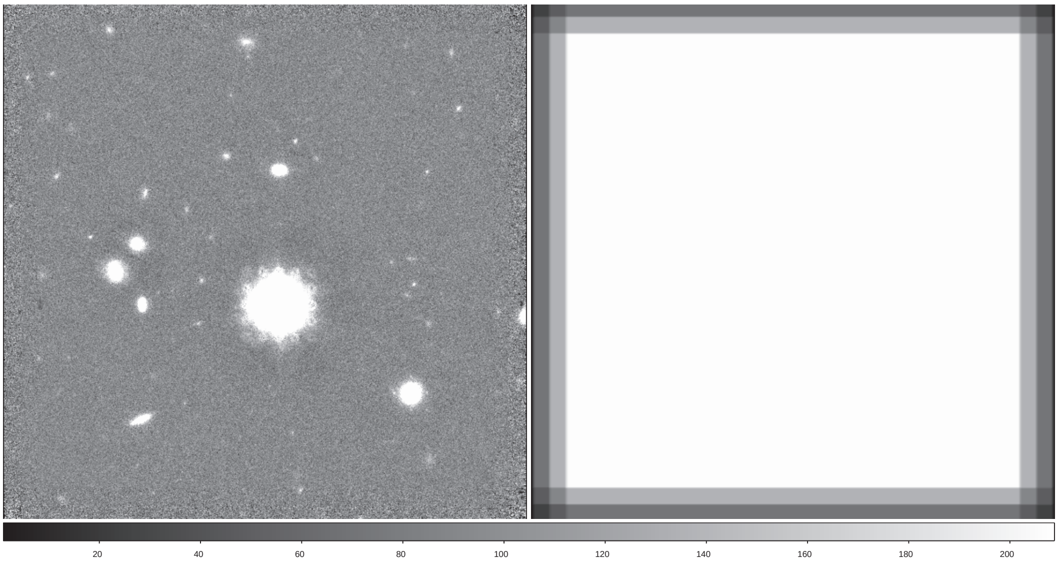


Figure 17: The final image of the scientific object (left) and integration-time map (right)

4.10 Analyzing Standard Star Images

You can reduce your standard star (P330-E for the sample data) frames with the same procedure applied to the object frames.

1. Making a list of the raw data, and inspecting their image quality

```
$ grep P330-E file_header.lst | awk '{print $1}' > stdstar_raw.lst
cl> frcheck stdstar_raw.lst mklist=yes calc=subtract
# Remove IRCA00091860 from the list in the case of the sample data
```

2. Making skyflat

```
$ sed -e "s/sample1/objmask/" stdstar_raw.lst-b > stdstar_mask.lst
```

```
cl> unlearn mkskyflat
cl> mkskyflat.bpmask = yes
cl> mkskyflat.bpfile = "ircsimg_bpmask.fits"
cl> mkskyflat.hsig = 2
cl> mkskyflat @stdstar_raw.lst-b stdstar_skyflat.fits @stdstar_mask.lst
```

3. Correcting for inequity of pixel sensitivity

```
$ sed -e "s/sample1/ff/" stdstar_raw.lst-b > stdstar_ff.lst
cl> unlearn flatfield
cl> flatfield.subdark = no
cl> flatfield.bpfix = yes
cl> flatfield.bpmask = "ircsimg_bpmask.fits"
cl> flatfield @stdstar_raw.lst-b @stdstar_ff.lst stdstar_skyflat.fits
```

4. Subtracting sky-background

```
$ sed -e "s/ff/ss/" stdstar_ff.lst > stdstar_ss.lst
cl> unlearn maskskysub
cl> maskskysub.ndthpos = 5
cl> maskskysub @stdstar_ff.lst @stdstar_ss.lst @stdstar_mask.lst
```

5. Measuring positional offset

```
cl> unlearn calcshift
cl> calcshift.referen = "ss/IRCA00091861.fits"
cl> calcshift.region = "[1:1024,1:1024]"
cl> calcshift.crmargin = 50
cl> calcshift.crtapersz = 50
cl> calcshift.crboxsi = 500
cl> calcshift @stdstar_ss.lst stdstar_offset.dat
```

6. Combining the position-shifted images

```
cl> unlearn imshiftcomb
cl> imshiftcomb @stdstar_ss.lst stdstar_final.fits stdstar_offset.dat
```

5 Estimate of Limiting Magnitude

After getting the fully reduced image, it would be interesting to see how deep one can integrate to detect faint objects. Such an image depth, i.e., the RMS noise level of the image, is usually represented by

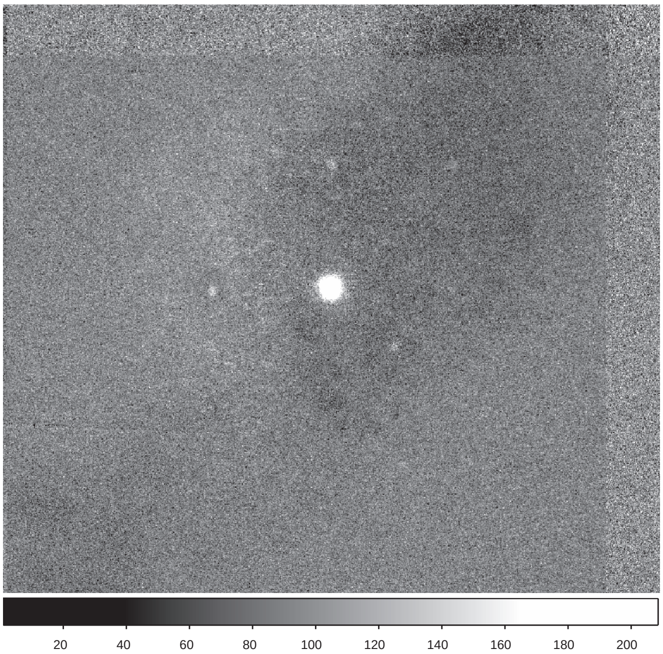


Figure 18: Final image of the standard star

the limiting magnitude. In many cases, the limiting magnitude is defined as 5 times the RMS noise level of the sky-background emission, namely, 5σ limiting magnitude. For extended source imaging (such as galaxies), an RMS calculated over a source-emission-free 1.0 arcsecond area should be used. On the other hand, some observers prefer to calculate the limiting magnitude using a compact region smaller than 1 arcsec^2 area under good seeing conditions (e.g., FWHM of a point source of less than 1 arcsec). To estimate limiting magnitude, we need to know the zero-magnitude of the image as well as the RMS of the sky-background. Here the zero-magnitude is given by a conversion factor representing the correspondence between the counts in ADU sec^{-1} and the magnitude.

5.1 First Step: Estimate of Zero-magnitude

To estimate the zero-magnitude, we mostly use the standard star data that have been taken on the same night, and at as close as possible to the airmass of the object(s). The sample data set consists of a standard star P 330-E adopted from Persson et al. 1998, AJ, 116, 2475. The star listed in the paper has $m_{K_s} = 11.429 \pm 0.006$ at the K_s band[6]. We need to know the star's brightness in ADU sec^{-1} unit in the fully reduced image in order to convert the count value into the zero-magnitude. This procedure is referred to as standard star photometry

It would be probably a good starting point to learn a method of photometry using the `APPHOT` command in IRAF. We suggest learning the other methods later. APPHOT implements a method of "aperture Imaging" which measures flux density received in a given circular aperture (see Figure 19).

[6]We assume that the star's flux density at the K_s band listed in the catalog can be applied to calibrate the sample data at the K' band because the fluxes at the two bands differ by less than 0.01 magnitude.

The task calculates noise level of the sky-level subtracted flux in an annulus which does not include the star (see the "width" annulus in Figure 19). The desired flux (F) can be written as

$$F = \sum_{r<r_1} S_{i,j} - Area\,(r < r_1) \times Median(r_2 < r < r_3)$$

where i, j denote pixel coordinates. $Area\,(r < r_1)$ is area of the circle with the "radius" in Figure 19. Here r_1 corresponds to `radius` in `APPHOT`. The input parameter for `APPHOT` of `buffer` can be written as $r_2 - r_1$, and `width` as $r_3 - r_2$.

We, therefore, have to supply an aperture radius to measure the flux of the target as well as inner radius and width of the annulus. A standard procedure to perform aperture imaging is summarized in the below.

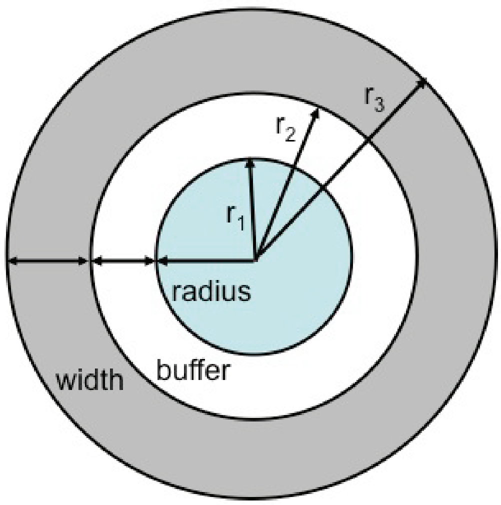


Figure 19: A sketch illustrating the IRAF `APPHOT` parameters used to perform aperture imaging of standard stars. See text.

1. Displaying an image to `ds9` containing the standard star

```
cl> display stdstar_final.fits 1
```

2. Configuring all the parameters for aperture photometry

   - Setting parameters to determine center position of the object that you want to measure (`centerpars`)

     If you wish to perform Imaging interactively, select the object in the `ds9` window. The center position of the selected star will be measured by `centerpars`. The following example demonstrates how the program searches for the star over a 5×5 pixel region that is specified with `cbox =5`.

```
cl> unlearn centerpars
cl> centerpars.calgori = "centroid"
cl> centerpars.cbox = 5
cl> centerpars.cthresh = 5
cl> centerpars.minsnra = 5
```

- Setting parameters to calculate sky-background level (`fitskypars`) The task `fitskypars` requires an inner radius (`annulus`) and width (`dannulus`) of the annulus where the sky-level is calculated. Also, the user has to specify which method (`salgorithm`) should be used to determine the representative pixel count in the annulus. For the sample data, a quick glance suggests adopting an `annulus` of 35 pixels and a `dannulus` of 10 pixels. If you are not happy with the resultant radial intensity profile of the star (Figure 20, you should optimize the two parameters. Here, we set `salgorithm` to "median" as the algorithm for calculating the sky-level.

```
cl> unlearn fitskypars
cl> fitskypars.salgori = "median"
cl> fitskypars.annulus = 35
cl> fitskypars.dannulus = 10
```

- Setting parameters for an aperture photometry (`photpars`)

  In the `photpars` section, we have to supply a radius of the aperture (`aperture`). We suggest optimizing the parameter to include all the pixels where the object is visible. As mentioned above, the user can repeat the measurements with different value of input parameters by inspecting resultant radial intensity profile. Of course, at this stage, it is useless to specify a conversion factor between flux density and magnitude since we don't know it yet. So just leave it as it is. We therefore should ignore the resultant magnitude, and just verify flux unit.

```
cl> unlearn photpars
cl> photpars.aperture = 30
```

3. Executing photometry

```
cl> unlearn phot
cl> phot.output = "stdstar_final_phot.dat"
cl> phot.interac = yes
cl> phot.radplot = yes
cl> phot stdstar_final.fits
```

After executing `phot`, back in the `ds9` window. Press the SPACE key to measure the flux of the desired star. This will display the desired photometry results on the IRAF terminal as shown below.

```
Warning: Graphics combine not available for display device.
stdstar_final.fits    632.63   693.71  -0.03782   12.766  ok
```

If you want to perform photometry on another object using the same parameters, just move your cursor to the next source, and press SPACE again once again. You will get a photometry result for the next star. To quit `phot` just type "q" on the `ds9` window, and type "q" once more on the terminal. If you specify the option `(radplot)` to be "yes", you will get a radial intensity profile of the object, as shown in Figure 20, as well as the aperture size and positions in the plot to verify the parameters specifying the annulus. This plot will help you to decide whether or not you need to re-configure the photometry parameters.

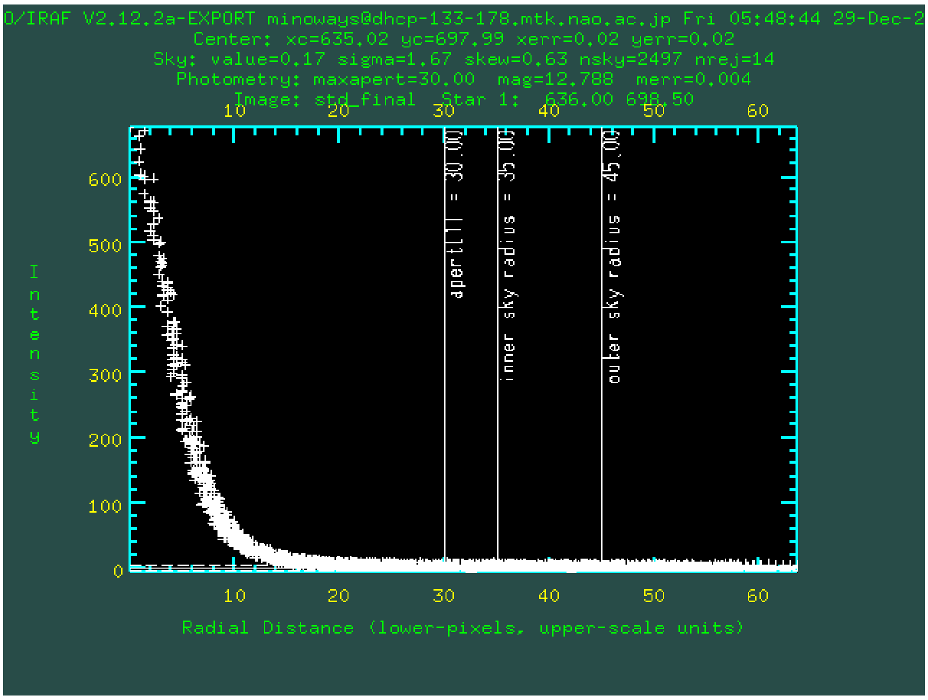


Figure 20: The radial intensity profile of the star used for photometry. This plot will appear together with the photometry results when using the `APPHOT` task. The vertical lines indicate the aperture radius, and inner radius of the annulus to define the sky. This information should be plotted to see the validity of the selected photometry parameters.

The photometry results will be stored in an ascii text file of `stdstar_final_phot.dat` with the selected parameters. Shown below is the desired photometry results that have been extracted from the file.

```
#N IMAGE                  XINIT      YINIT      ID     COORDS                  LID    \
#U imagename              pixels     pixels     ##     filename                ##     \
#F %-23s                  %-10.3f    %-10.3f    %-6d   %-23s                   %-6d
#
```

```
#N XCENTER    YCENTER    XSHIFT  YSHIFT  XERR    YERR            CIER CERROR   \
#U pixels     pixels     pixels  pixels  pixels  pixels          ##   cerrors  \
#F %-14.3f    %-11.3f    %-8.3f  %-8.3f  %-8.3f  %-15.3f         %-5d %-9s
#
#N MSKY           STDEV          SSKEW          NSKY   NSREJ     SIER SERROR   \
#U counts         counts         counts         npix   npix      ##   serrors  \
#F %-18.7g        %-15.7g        %-15.7g        %-7d   %-9d      %-5d %-9s
#
#N ITIME          XAIRMASS       IFILTER                OTIME                   \
#U timeunit       number         name                   timeunit                \
#F %-18.7g        %-15.7g        %-23s                  %-23s
#
#N RAPERT   SUM            AREA       FLUX           MAG    MERR   PIER PERROR   \
#U scale    counts         pixels     counts         mag    mag    ##   perrors  \
#F %-12.2f  %-14.7g        %-11.7g    %-14.7g        %-7.3f %-6.3f %-5d %-9s
#
tdstar_final          632.000   693.000   1     nullfile              0      \
   632.628    693.709    0.627   0.709   0.044   0.044          0    NoError   \
   -0.03782059    1.937127       -0.3873699     2492   20        0    NoError   \
   1.             INDEF          INDEF                  INDEF                   \
   30.00    78170.34       2827.482   78277.28       12.766 0.004 0    NoError
```

This result tells you that the standard star has a flux density of F =78277.28 [ADU $\sec^{-1}$]. To convert this flux into a magnitude, use the following equation,

$$m_0 = m + 2.5 \log(F)$$

where m is the magnitude of the star found in a catalog and m_0 denotes the zero-magnitude which corresponds to 1.0 ADU $\sec^{-1}$. Using $F = 78277.28$ [ADU $\sec^{-1}$ and $m = 11.429$, we obtain $m_0 = 23.66$.

5.2 Second Step: Estimate of Image Noise Level

Except for a very few cases, since the attainable image noise levels in NIR imaging tend to be limited by the background noise level (see §2.2). we should measure image noise level from the standard deviation (σ) of the source-emission-free (= sky-background only) region. The task `imstat` in IRAF calculates the RMS for the desired region. If your final image consists of point-sources only, like the sample data, you can just issue `imstat` command. If your image consists of extended emission, you will need to carefully select source-emission-free region(s) that has(have) only background radiation only to execute `imstat`. In the specific case of the sample data, we can utilize the image of `ff-obj` which is the frame where we have removed the contribution from bright sources. We should be able to make a source-emission-free image by subtracting the sky-background level from `ff-obj` and then performing "shift-and-add".

```
$ mkdir ss-obj
$ sed -e "s/ff/ss-obj/" object_ff.lst > object_ss-obj.lst
cl> unlearn eraseskysub
cl> eraseskysub @object_ff-obj.lst @object_ff-obj.lst @object_ss-obj.lst
cl> unlearn imshiftcomb
cl> imshiftcomb @object_ss-obj.lst object_background.fits object_offset_final.dat
```

Using this image, one can estimate noise level of the sky-background with `imstat` as follows.

```
cl> unlearn imstat
cl> imstat object_background.fits nclip=50
#                IMAGE      NPIX      MEAN    STDDEV       MIN       MAX
 object_background.fits   1190695  0.009196    0.0163  -0.03971    0.0581
```

This tells you that the standard deviation (STDDEV) of the emission-free region is $\sigma = 0.0163$ in ADU sec^{-1} unit.

As seen in the equation calculating S/N ratio (see §2.2), the noise level over an aperture with a radius of r arcsec can be estimated from the following equation,

$$\sigma_r = \sigma \times \sqrt{\pi (r/p)^2}$$

where r denotes a pixel size scale in arcsec pixels^{-1} unit. In practice, the relationship between the aperture radius r and the measured noise level does not show a linear relationship for many cases. Instead, our experience suggests that the σ_r tends to increase quadratically for many cases (see Figure 21). We speculate that this is because the standard deviation measured at each pixel is not independent, and correlates with some larger scale parameters[7]. However, if your aperture is small enough, you may consider using the standard deviation between the pixels as long as your aperture radius is less than 5 pixels where the dependence is linear.

If you wish to measure the limiting magnitude over a large aperture, you should adopt a standard deviation for several source-emission-free regions where the flux in each region is measured over an aperture with a radius of R arcsecond.

5.3 Estimate of Limiting Magnitude

To determine limiting magnitude of a point-like source (i.e., to convert the 5σ image noise level into a magnitude), we should calculate $S/N(r)$ obtained from the point source flux density, "encircled flux" $F(r)$, and the noise level of a point source aperture,

$$S/N(r) = F(r) / \left(\sigma \sqrt{\pi r^2 / p^2} \right)$$

[7]The possible reasons are: (1) the pixel size scale is smaller than the size characterizing the atmospheric conditions, and (2) we may have lost independence of the information contained in each pixel when we have performed "subpixel-shifting".

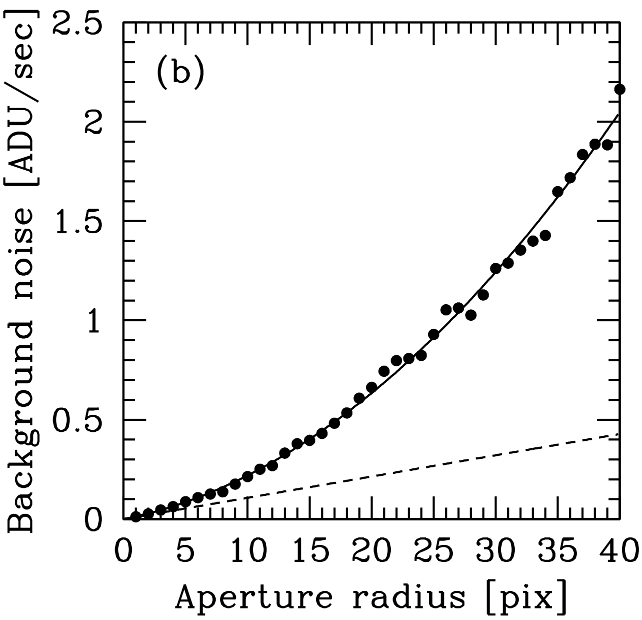


Figure 21: An empirical relationship between background noise level and aperture radius (Minowa et al. 2005, ApJ, 629, 29).

Let us consider the case whether S/N reaches its maximum at $r = r_{max}$ for a point-source with m_{psf}. We can calculate the 5σ limiting magnitude with

$$m_{lim} = m_{psf} + 2.5 \log(S/N(r_{max})/5)$$

The selected pointe source object circled in green in Figure 22 (a) is used to calculate the limiting magnitude. First, we have to know the encircled-flux $F(r)$ for the point-source using APPHOT implemented in `growth` task in package ircs_imgred. We suggest changing the aperture radius by hand with `APPHOT` to see how it varies, although `growth` does this automatically. The output of the `growth` task creates an ascii text file where the photometry results with various aperture are stored. The task can be used as follows.

```
cl> epar growth
PACKAGE = ircs_imgred
   TASK = growth


in      =         object_final  Input image file name
out     =    growth_objpsf.dat  Output growthcurve data
(rmin   =                   1.) Minimum Radius of growth curve
```

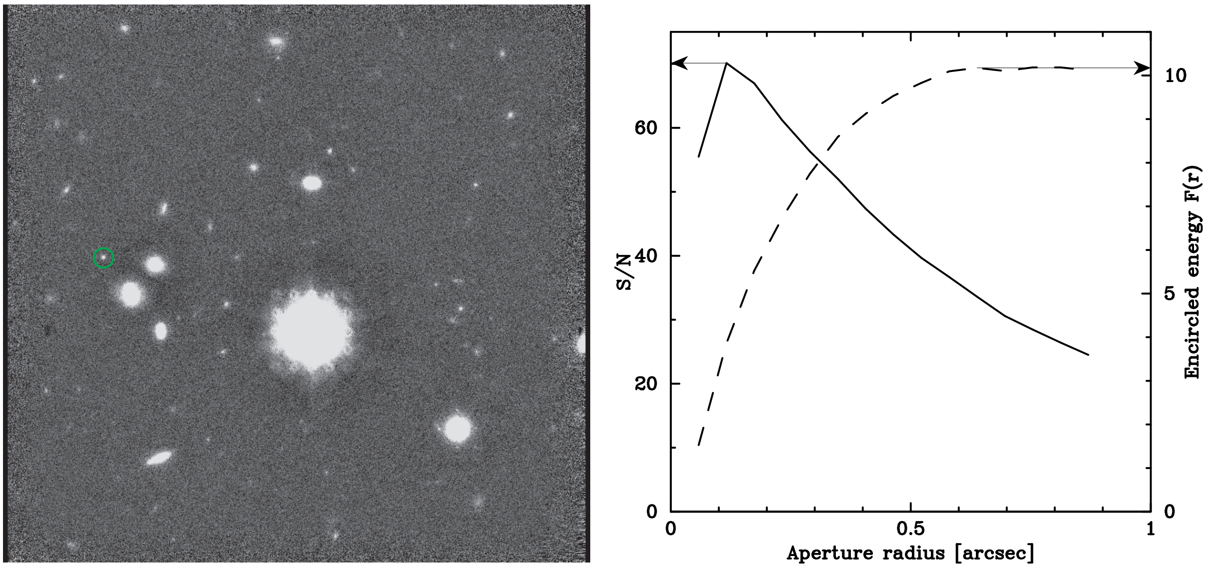
```
(rmax   =                   15.) Maximum Radius of growth curve
(rbin   =                    1.) Resolution of growth curve
(salgor =               median) Sky fitting algorithm
(annu   =                   15.) Inner Radius of sky annulus
(dannu  =                    5.) Width of sky annulus
(sky    =                    0.) Constant sky value (/pixel)
(mode   =                    ql)
# :g execute the task
```

After execution, you will see the image on the `ds9` window, move the cursor onto the source, type "a" to measure, then "q" to quit. The photometry results will be written into a file named `growth_objpsf.dat`. Here, you have to accordingly supply the minimum and maximum of the aperture, step width (to change the aperture radii), of course, an algorithm to define sky-level as well as inner radius and width of the annulus. These parameters should be optimized for your source because the above parameters are specific case to the sample data.

Next, calculate S/N with as a function of aperture radius using the encircled energy flux. Aperture (in pixel unit) and flux density (in ADU $\sec^{-1}$) can be found in the `growth_objpsf.dat` file in the 1st and 2nd columns, respectively. The following UNIX/Linux command will convert the aperture radius by multiplying with $p = 0.058$ arcsec pix^{-1} into pixel size scale, and convert the flux into S/N by dividing with $\sigma_r = \sigma\sqrt{\pi r^2}$.

```
$ awk '{print ($1*0.058),($2/(0.0163*sqrt(3.1416)*$1))}' growth_objpsf.dat
> snr_objpsf.dat
```

Figure 22 (b) is a plot of the S/N ratio and encircled flux as a function of aperture radius obtained by the `growth` command. Using this figure in conjunction with `growth_psf.dat`, we estimate that the point-source has a total flux density of 10.2 ADU $\sec^{-1}$ that corresponds to 21.14 magnitude using the conversion factor (recall §5) of $m_0 = 23.66$. In addition, we see that the S/N ratio of the point-source peaks ~ 66.8 at an aperture radius of $r = 0.116$, which yields a 5σ limiting magnitude of $21.14 + 2.5\log(66.8/5) \simeq 23.95$.



(a) Position of the point-sources in the sample data (b) Encircled energy for a point-source and S/N

Figure 22: Estimate of limiting magnitude for a point source

Subsequently, let us estimate the 5σ noise level of the image in surface brightness which is expressed by a limiting magnitude over 1 arcsecond area, μ_{lim} in mag arcsec^{-2}. For many cases, such a detection threshold is represented by the standard deviation of the flux over the 1 arcsecond area aperture ($\sigma_{1\mathrm{arcsec}^2}$) given by,

$$\mu_{lim}\,[\mathrm{mag/arcsec}^2] = m_0 - 2.5\log(5\sigma_{1\mathrm{arcsec}^2})$$

Notice that the $\sigma_{1\mathrm{arcsec}^2}$ is defined by standard deviation calculated over a circular aperture with a radius of 9.7 pixels. This tends to case that the calculated RMS to be larger than that from the standard deviation calculated on the pixel count values as discussed above. We therefore have to measure the fluxes that exist in a 1 arcsecond area, identical to the circular aperture with 9.7 pixel radius of the fully reduced final image of the object (`object_final.fits` in the case of the sample data), then calculate the standard deviation of the fluxes assuming a Gaussian distribution. These steps can be done with `calcnoise` in ircs_imgred. For the sample data, we should adopt such a circular aperture with a radius of 9.7 pixel, and accordingly set the `region` parameter that specifies which area we should use to measure the RMS of the flux. It should be noted that we have selected the central region where all the images have been combined by looking at `object_expmap.fits`. The parameter `maxaper` is the maximum number of apertures that will be located in the image randomly. You may have to optimize some parameters to obtain the best results by inspecting a histogram of the count values. The histogram is plotted by calling the `histogram` command in IRAF. You may have to look for proper set of the histogram plotting parameters (e.g., `nhistbin, hist_z1, hist_z2`) in order to display the results accordingly.

```
cl> epar calcnoise
```

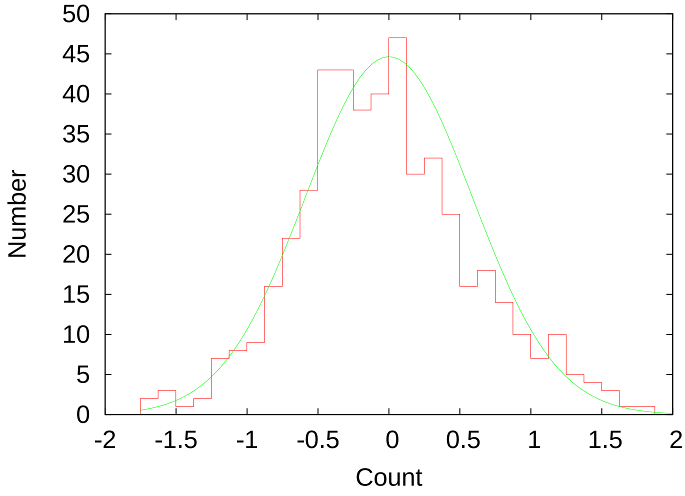


Figure 23: An estimate of the background noise level per 1 arcsec^2 aperture. The plot represents a histogram of the flux densities over a random aperture and the best-fit Gaussian profile. Since the positive part of the histogram consists of contributions from the objects, the Gaussian fitting was done solely using the negative side.

```
                              I R A F
                 Image Reduction and Analysis Facility
PACKAGE = ircs_imgred
   TASK = calcnoise

image   =         object_final  Image file name.
rapertur=                  9.7  Aperture radius
region  =     [71:1025,71:1025] Image region for noise estimation
(maxaper=                  500) Maximum number of apertures
(nhistbi=                   30) Number of bins in histogram
(hist_z1=                INDEF) Min bin value in histogram
(hist_z2=                   2.) Max bin value in histogram
(list1  =                     )
(list2  =                     )
(mode   =                   ql)
# :g executes the task.

### Result
#1 sigma[adu] in aperture radius 9.70[pix]
0.59
```

Figure 23 represents a histogram of fluxes in the 1 arcsecond aperture as well as the best-fit Gaussian profile. The plot tells us that RMS of the flux ($\sigma_{1\mathrm{arcsec}^2}$) approximately corresponds to 0.59 [ADU]. Applying the results to the above equation, we obtain the desired limiting magnitude in surface brightness of $\mu_{lim} = 22.5$ mag arcsec^{-2}.