Laboratory

IRAF is a general Image Reduction and Analysis Facility, providing a wide range of image processing tools for the user

First time using IRAS:

1. Execute the command **mkiraf**
   - It creates a file called login.cl and a directory called UPARM
   - Login.cl => File executed at IRAF start-up time
   - UPARM: It is used by IRAF to save the customized parameters files
   - After that : choose a graphic terminal type: gterm, xterm, **xgterm...**

START:

- **xgterm –sb &**
- ximtool or **sd9 &**
- **cl**

OUT:

- Bye
- logout

TEXT EDITOR:

- emacs
- vi
- ...

RECOMMENDED BIBLIOGRAPHY

- A beginner's Guide to Using IRAF (Jannette Barnes)
- A User's Guide to Stellar CCD Photometry with IRAF(P. Massey, L.E. Davis, Apr 1992)
- A User's Guide to CCD Reduction with IRAF - by P. Massey, Feb 1997
- Cleaning Images of Bad Pixels and Cosmic Rays – by L. Wells, Sept 1994
- http://www.iac.es/galeria/ncaon/IRAFSoporte/Iraf-Manuals.html#Packages

Modifying the login.cl

emacs login.cl ↵

And follow the red notes

- **Uncomment (no #) the important parameters**
- **Change the "imtype" parameter**
- **Configure the printer**
- **Configure the text editor**

Then, we are ready to start to work!!!

cl ↵

```
# LOGIN.CL -- User login file for the IRAF command language.
# Identify login.cl version (checked in images.cl).
if (defpar ("logver"))
   logver = "IRAF V2.11 May 1997"
set             home                    = "/home/ovega/"
set             imdir                   = "HDR$pixels/"
set             uparm                   = "home$uparm/"
set             userid                  = "ovega"
# Set the terminal type.
if (envget("TERM") == "sun") {
   if (!access (".hushiraf"))
                print "setting terminal type to xgterm..."
   stty xterm
} else {
   if (!access (".hushiraf"))
                print "setting terminal type to xgterm..."
   stty xterm
}
# Uncomment and edit to change the defaults.
set             editor                  = emacs
set             printer                 = lpr
#set            stdimage                = imt800
#set            stdimcur                = stdimage
set             stdplot                 = lpr
#set            clobber                 = no
#set            filewait                = yes
#set            cmbuflen                = 512000
#set            min_lenuserarea         = 64000
set             imtype                  = "fits,inherit"
```

cl

Mensage:

Welcome to IRAF. To list the available commands, type ? or ??. To get detailed information about a command, type `help command'. To run a command or load a package, type its name. Type `bye' to exit a package, or `logout' to get out of the CL. Type `news' to find out what is new in the version of the system you are using. The following commands or packages are currently defined:

PACKAGE

color. images. mscred. plot. stsdas.
dataio. inaoe. nmisc. proto. system.
dbms. language. noao. r2rvsao. tables.
guiapps. lists. obsolete. softools. utilities.

Moving inside IRAF:

Directories: **cd, ls, del, mkdir, imdelete** …

name.= Package = set of task which made a determined procesess, x ej. Data reduction

name = task

helps name= information of name (task or package)

? : list the tasks in the package

- Redirection command (like en UNIX), x Ej **ls *.dat > data , help task |lprint dev=<lpr>**
- reference : If I only know part of the task name

Laboratory

IRAF works with files with **"fits"** format:

- **Text part** = **header**: It can be edited. It give us some information about the observations

+

- **Data part** = binary. It can not be edited

Our files are not in fits formats, so we have to rename them, so

STEPS:

It is only rename the extension of the file, not the total name

1. **Rename *.FIT fits field=extn**
2. To see the header : **imhead name_file lo+**
3. Display an image: **display name_file**

```
6047O[1024,1024][real]: NGC7006
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 2673 s.u.
Created Wed 21:48:03 27-Jul-2005, Last modified Wed 21:44:45 27-Jul-2005
Pixel file "6047O.fits" [ok]
EXTEND  =                    F / File may contain extensions
DATE    = '2005-07-26T23:28:22' / Date FITS file was generated
ORIGIN  = 'NOAO-IRAF FITS Image Kernel July 2003' / FITS file originator
IRAF-TLM= '21:44:45 (27/07/2005)' / Time of last modification
EXPTIME =                  300
FILTER  = 'V      '
IMAGETYP= 'OBJECT  '
CCDSUM  = '1 1     '
CCDSEC  = '[1:1024,1:1024]'
OBJECT  = 'NGC7006 '
RA      = '21:01:29.00'
DEC     = '16:11:15.00'
UT      = '11:20:11'
DATE-OBS= '2001-05-28'
TELESCOP= 'OAN/SPM 1.5-m'
LATITUDE= '+31:02:39'
LONGITUD= '-115:27:49'
ALTITUDE=                  2800
OBSERVER= 'Vahram Chavushyan'
INSTRUME= 'RUCA    '
DETECTOR= 'SITe1 1k'
GAINMODE=                    1
GAIN    =                    5
RDNOISE =                 14.5
HISTORY  PMIS macros of 2001-01-23
HISTORY  Written by Stephen Levine
HISTORY  Modified by Gaguik Tovmassian
HISTORY  Modified by Alan Watson & Michael Richer
```

Typical header

Reduction:

Now, we have the data ready to work with:

STEPS:

1. **REDUCTION PROCCES= Subtract all the noise that the instrumental part (telescope + CCD…) introduce**
2. **PHOTOMETRY = Calculate the instrumental magnitudes and transform them in real apparent magnitude**

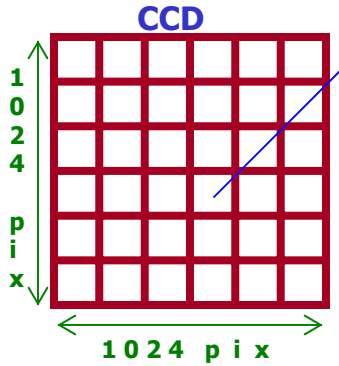
REDUCTION:

NOISES:

1. **Bias** : It has to be removed from all the data
2. ~~**Dark current** : It has to be removed from all the data excepts the bias frames~~
3. **Flat- fielding** : It depends on the filter. We have to create a flat field frame for each filter, and divide each frame of the objects by its corresponding flat
4. **Cosmic rays**

PACKAGE = CCDRED

Where? : **NOAO** ❯ **IMRED** ❯ **CCDRED** ❯



PIXEL i,j ➡ In each pixel we read S_{ij} (e^- ~ counts)

$$S_{ij}=B_{ij}+D_{ij}\,t_{exp}+(I_{ij}(\lambda)\,t_{exp})\,F_{ij}((\lambda)$$

CCD IMAGE at that filter — BIAS — DARK — **STAR IMAGE at that filter** — FLAT (filter)

$$I_{ij}(\lambda)\,t_{exp}=[((S_{ij}-B_{ij})-D_{ij}\,t_{exp}))]\,/\,F_{ij}(\lambda)$$

```
im> ccdred ❯
      badpiximage   ccdlist      combine      mkillumcor   setinstrument
      ccdgroups     ccdmask      darkcombine  mkillumflat  zerocombine
      ccdhedit      ccdproc      flatcombine  mkskycor
      ccdinstrument ccdtest      mkfringecor  mkskyflat
cc>
```

No . => task

Reduction:

STEPS:	Tasks:
1. Combine bias to make a bias frame :	zerocombine
2. Remove bias from all files :	ccdproc
3. Combine flat for each filter :	Flatcombine
4. Divide by flat all the images (filter):	ccdproc

Check in the log file

what files are bias : *b.fits

What files are flats : *.f.fits

What files are objects: *o.fits

It is a very good idea if we check the images, using the DISPLAY task, IMEXAMINE task , or IMARITH task. If any image has any problem, it is better to eliminate it DO NOT WORRY ABOUT SPENDING TIME, IT IS IMPORTANT TO DO IT!

1.- Combine bias: zerocombine

- Make a file ls *.fits > **tobias.lis** = It contains the files to remove the average bias frame.
- ls *b.fits >**bias.lis** => It cointains bias files that I will combine with **"zerocombine"**. OUTPUT : **Zero.fits** = **average bias frame**

im> epar zerocombine

Image Reduction and Analysis Facility

PACKAGE = ccdred

TASK = **zerocombine**

NOTE: It is good idea to check the result with imaxime task

input = **@bias.lis** List of zero level images to combine

(output = **Zero**) Output zero level name

(combine= **average**) Type of combine operation

(reject = **minmax**) Type of rejection (

(ccdtype=) CCD image type to combine

(process= no) Process images before combining?

(delete = no) Delete input images after combining?

(clobber= no) Clobber existing output image?

(scale = none) Image scaling

(statsec=) Image section for computing statistics

(nlow = 0) minmax: Number of low pixels to reject

(nhigh = 1) minmax: Number of high pixels to reject

(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)

(mclip = yes) Use median in sigma clipping algorithms?

(lsigma = 3.) Lower sigma clipping factor

(hsigma = 3.) Upper sigma clipping factor

(rdnoise= 8.1) ccdclip: CCD readout noise (electrons)

(gain = 1.2) ccdclip: CCD gain (electrons/DN)

(snoise = 0.) ccdclip: Sensitivity noise (fraction)

(pclip = -0.5) pclip: Percentile clipping parameter

(blank = 0.) Value if there are no pixels

(mode = ql)

: go => Save and run

Reduction:

2.- **SUBSTRACTING THE BIAS: CCDPROC**

im> epar ccdproc

- File **tobias.lis**
- Frame **Zero.fits**

- Determine which part of the chip contain useful data
- "trim sec = section of raw image that will be saved [1:1024,1:1024]
- "biassec" = section used in the bias level [1025:1072,1:1024]

These information are in the headers

```
I R A F
PACKAGE = ccdred
   TASK = ccdproc

images =      @tobias.lis  List of CCD images to correct
(output =                  ) List of output CCD images
(ccdtype=                  ) CCD image type to correct
(max_cac=                   0) Maximum image caching memory (in
Mbytes)
(noproc =                  no) List processing steps only?

(fixpix =                  no) Fix bad CCD lines and columns?
(oversca=                  no) Apply overscan strip correction?
(trim   =                 yes) Trim the image?
(zerocor=                 yes) Apply zero level correction?
(darkcor=                  no) Apply dark count correction?
(flatcor=                  no) Apply flat field correction?
(illumco=                  no) Apply illumination correction?
(fringec=                  no) Apply fringe correction?
(readcor=                  no) Convert zero level image to readout
correction?
(scancor=                  no) Convert flat field image to scan correction?

(readaxi=                line) Read out axis (column|line)
(fixfile=                  ) File describing the bad lines and columns
(biassec=  [1025:1072,1:1024]) Overscan strip image section
(trimsec=   [1:1024,1:1024]) Trim data section
(zero   =                Zero) Zero level calibration image
```

(cont. ccdproc)

```
(dark   =                  ) Dark count calibration image
(flat   =                  ) Flat field images
(illum  =                  ) Illumination correction images
(fringe =                  ) Fringe correction images
(minrepl=                 1.) Minimum flat field value
(scantyp=          shortscan) Scan type (shortscan|longscan)
(nscan  =                  1) Number of short scan lines
(interac=                 no) Fit overscan interactively?
(functio=          legendre) Fitting function
(order  =                  1) Number of polynomial terms or
spline pieces
(sample =                  *) Sample points to fit
(naverag=                  1) Number of sample points to
combine
(niterat=                  1) Number of rejection iterations
(low_rej=                 3.) Low sigma rejection factor
(high_re=                 3.) High sigma rejection factor
(grow   =                 0.) Rejection growing radius
(mode   =                 ql)
```

:go = > Save and run

NEXT STEP : FLAT-FIELDING correction

Reduction:

3.-

FLAT FIELDING CORRECTION:

1. CREATE a FLAT frame ➡TASK = Flatcombine

im > epar flatcombine

Flats depend on the filter(λ) => separate by filter

Files to create the flats	**Files to apply the flats**
B : flatB.lis	toflatB.lis
V: flatV.lis	toflatV.lis

Ej.

flatB.lis
6001F.fits
6002F.fits
6003F.fits

toflatB.lis
6016O.fits
6030O.fits
6044O.fits
6046O.fits
6048O.fits

OUTPUT: Flat frame ➡ **flatB.fits**

NOTE: It is good idea to check the result with imaxime task

NEXT STEP: we have to correct files in **toflatsB.lis** of flat-field ➡ **CCDPROC** again

PACKAGE = ccdred

TASK = flatcombine

input = **@flatB.lis** List of flat field images to
(output = **flatB** Image Reduction and Analysis Facility)
Output flat fie
(combine= **average**) Type of combine operation
(reject = **crreject)** Type of rejection
(ccdtype=) CCD image type to combine
(process= no) Process images before combining?
(subsets= no) Combine images by subset parameter?
(delete = no) Delete input images after combining?
(clobber= no) Clobber existing output image?
(scale = mode) Image scaling
(statsec=) Image section for computing statistics
(nlow = 1) minmax: Number of low pixels to reject
(nhigh = 1) minmax: Number of high pixels to reject
(nkeep = 1) Minimum to keep (pos) or maximum to reject (neg)
(mclip = yes) Use median in sigma clipping algorithms?
(lsigma = 3.) Lower sigma clipping factor
(hsigma = 3.) Upper sigma clipping factor
(rdnoise= **8.1**) ccdclip: CCD readout noise (electrons)
(gain = **1.2**) ccdclip: CCD gain (electrons/DN)
(snoise = 0.) ccdclip: Sensitivity noise (fraction)
(pclip = -0.5) pclip: Percentile clipping parameter
(blank = 1.) Value if there are no pixels
(mode =

: go **=> Save and run**

Reduction:

3.-

FLAT FIELDING CORRECTION:

2. Divide by the flat-field ➡TASK = ccdproc

im> epar ccdproc

Files to appy the flats : toflatsB.lis

Average Flat frame to use: flatB.fits

REPIT the process for all the filters...

CHECK that all the images look like well.

- Display the images
- Look for error... (dust grains, fringes...)
- Headers
- Imexam...

NEXT STEP Cosmic rays corrections

TASK = ccdproc

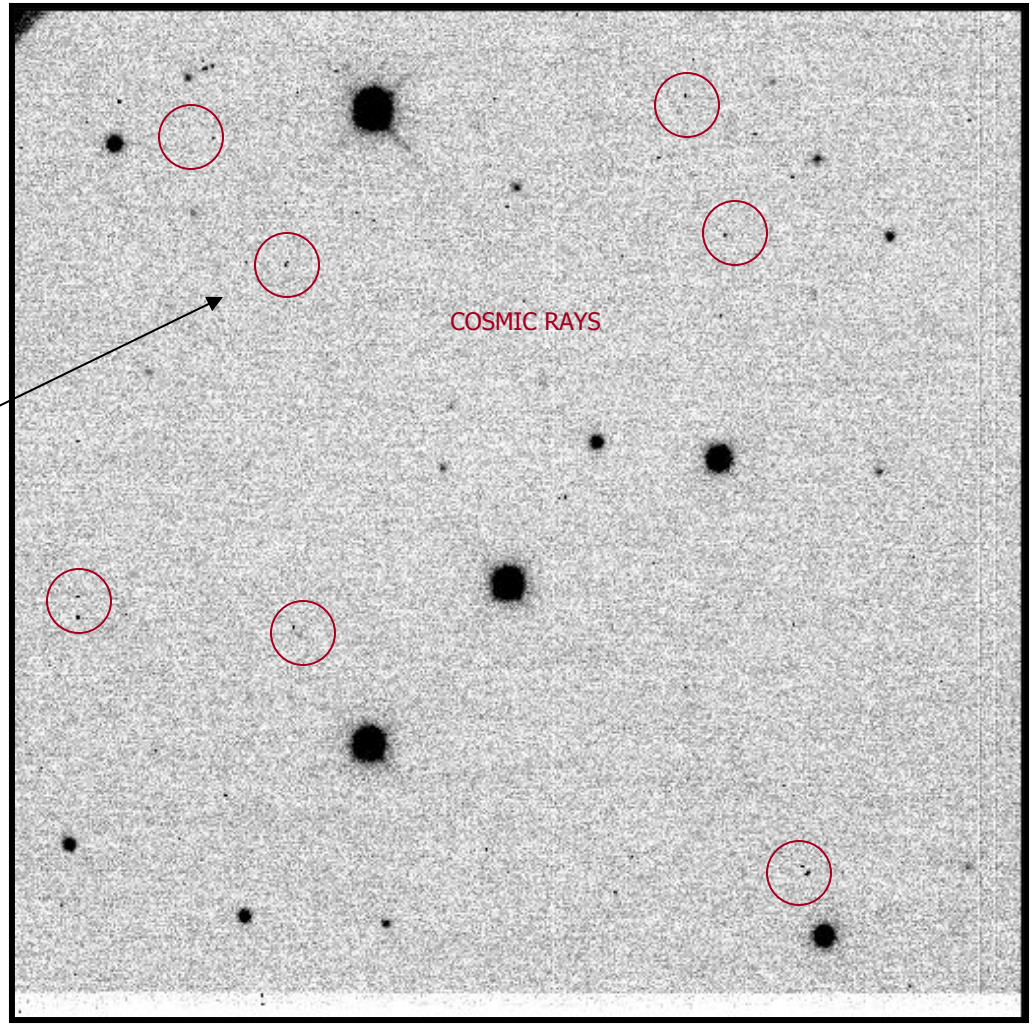
```
images =      @toflatB.lis  List of CCD images to correct
(output =                ) List of output CCD images
(ccdtype=                ) CCD image type to correct
(max_cac=                0) Maximum image caching memory (in Mbytes)
(noproc =               no) List processing steps only?
(fixpix =               no) Fix bad CCD lines and columns?
(oversca=               no) Apply overscan strip correction?
(trim   =              yes) Trim the image?
(zerocor=              yes) Apply zero level correction?
(darkcor=               no) Apply dark count correction?
(flatcor=              yes) Apply flat field correction?
(illumco=               no) Apply illumination correction?
(fringec=               no) Apply fringe correction?
(readcor=               no) Convert zero level image to readout correction?
(scancor=               no) Convert flat field image to scan correction?
(readaxi=             line) Read out axis (column|line)
(fixfile=                ) File describing the bad lines and columns
(biassec=  [1025:1072,1:1024]) Overscan strip image section
(trimsec=   [1:1024,1:1024]) Trim data section
(zero   =             Zero) Zero level calibration image
(dark   =                ) Dark count calibration image
(flat   =            flatB) Flat field images
(illum  =                ) Illumination correction images
(fringe =                ) Fringe correction images
(minrepl=              1.) Minimum flat field value
(scantyp=      shortscan) Scan type (shortscan|longscan)
(nscan  =              1) Number of short scan lines
(interac=              no) Fit overscan interactively?
(functio=      legendre) Fitting function
(order  =              1) Number of polynomial terms or spline pieces
(sample =              *) Sample points to fit
(naverag=              1) Number of sample points to combine
(niterat=              1) Number of rejection iterations
(low_rej=              3.) Low sigma rejection factor
(high_re=              3.) High sigma rejection factor
```

4.-

Cosmic rays: Random events which can occur at any place on an image. They are not correcting by F-F. So we have to use other methods to clean our image.

Normally they are seen as one very hot pixel

How to prevent them!



Cosmic rays: make several exposition and made an average image.

Bad pixels: several images lightly shifted each other, and combine.

If it is not possible : Use the task **cosmicrays**

HOMEWORK!!!!!

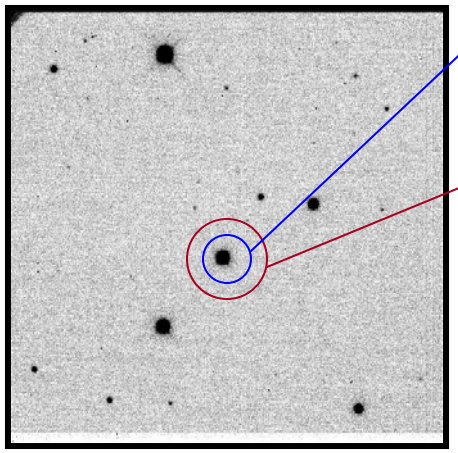
Use help and

Cleaning Images of Bad Pixels and Cosmic Rays Using IRAF by Lisa Wells

Photometry:

Photometry: Calculate the Instrumental Magnitudes

- We are going to make **photometry of simple stars => Package: Apphot (aperture photometry)**
- If we want to make crowed fields photometry => package : Digiphot



Aperture: We integrate all the counts inside the aperture = star + background counts = A1

Background annulus: We integrate all the counts in this annulus = background counts = A2

STAR COUNTS = A1-A2

PROBLEM in IRAF

- **There is not a good method to select the correct value for the aperture**
- **Too high aperture => errors by bad flat-fielding correction, bad pixels, cosmic rays...**
- **Too low aperture => we could not include all the energy from the star.**

STEPS

1. Put in the headers all the values that the task will need
2. In each finding chart, choose the stars that we want to calculate the photometry
3. Calculate the apertures, and background annulus for each star in the frame
4. From those, choose an average value for the aperture and background annulus
5. Fill the parameters files that the tasks will need
6. RUN THE TASK which calculates the aperture magnitude for each star = **phot**

Package:

cl> dighiphot ⤵ apphot ⤵ epar phot ⤵

1.-

Check the header and introduce the things that you need:

- Coordinates (RA, DEC)
- **UT**
- **Siderial time**
- **Epoch**
- Observatory parameters
- **Exposure times**
- **Date-obs**
- CCD parameters (gain, RN...)
- Filter (10=B, 20=V)
- **Airmass** (need the coordinates and the TS)

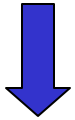
How to check in the header: imhead name lo+

How to introduce the values :

hedit *.fits "name parameter" ' value' add+ ver- up+ show+

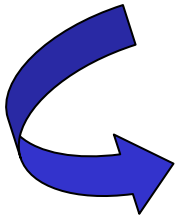
CHECK the log file: In all of them you will have some information

BUT, not for the airmass



AIRMASS calculation:

- It need the **Siderial Time** of the exposition
- It is not usually in the header
- We have to calculate it



Photometry:

ST: siderial time

epar asthedit

```
I R A F
                Image Reduction and Analysis Facility
PACKAGE = astutil
  TASK = asthedit

images  =          @toST.lis  Images to be operated upon
commands=          olga.spm  File of commands
(table  =          mi) File of values
(colname=          date) Column names in table file
(prompt =           ) Prompt for STDIN commands
(update =          yes) Update image header?
(verbose=          yes) Verbose output?
(oldstyl=          no) Use old style format?
(mode   =          ql)
```

File : mi	File: toST.lis
2001-05-28	6015O.fits
2001-05-28	6016O.fits
2001-05-28	6029O.fits
2001-05-28	6030O.fits
2001-05-28	6043O.fits
2001-05-28	6044O.fits
2001-05-28	6045O.fits
2001-05-28	6046O.fits
2001-05-28	6047O.fits
2001-05-28	6048O.fits

FILE which contains al the files that we want to update the headers

FILE: olga.smp

```
observat = "obspars"
    print($date)
    print(UT)
    st = mst ($date, ut, obsdb (observat, "longitude"))
    epoch = epoch ($date,UT)
```

OBSERVATORY DATA

Parameter in file MI

UNIVERSAL TIME in header

Tasks which calculate the ST, and the epoch of the observation.

Photometry:

Airmass: epar setairmass

```
I R A F
                Image Reduction and Analysis Facility
PACKAGE = astutil
  TASK = setairmass

images =          @toST.lis  Input images
(observa=             spm) Observatory for images
(intype =       beginning) Input keyword time stamp
(outtype=       effective) Output airmass time stamp
(ra     =              ra) Right acsension keyword (hours)
(dec    =             dec) Declination keyword (degrees)
(equinox=           epoch) Equinox keyword (years)
(st     =              st) Local siderial time keyword (hours)
(ut     =              ut) Universal time keyword (hours)
(date   =       date-obs) Observation date keyword
(exposur=         exptime) Exposure time keyword (seconds)
(airmass=         airmass) Airmass keyword (output)
(utmiddl=       utmiddle) Mid-observation UT keyword (output)
(scale  =          750.) The atmospheric scale height

(show   =             yes) Print the airmasses and mid-UT?
(update =             yes) Update the image header?
(overrid=             yes) Override previous assignments?
(mode   =              ql)
```

It is in IRAF!

HEADER

```
6030O.fits[1024,1024][real]: NGC4147
No bad pixels, min=0., max=0. (old)
Line storage mode, physdim [1024,1024], length of user area 2673 s.u.
Created Tue 18:29:49 26-Jul-2005, Last modified Wed 21:44:44 27-Jul-2005
Pixel file "6030O.fits" [ok]
EXTEND  =                    F / File may contain extensions
DATE    = '2005-07-26T23:29:49' / Date FITS file was generated
ORIGIN  = 'NOAO-IRAF FITS Image Kernel July 2003' / FITS file originator
IRAF-TLM= '21:44:44 (27/07/2005)' / Time of last modification
EXPTIME =                  300
FILTER  = 'B       '
IMAGETYP= 'OBJECT  '
CCDSUM  = '1 1     '
CCDSEC  = '[1:1024,1:1024]'
OBJECT  = 'NGC4147 '
RA      = '12:10:06.20'
DEC     = '18:32:32.00'
UT      = '08:00:16'
DATE-OBS= '2001-05-28'
TELESCOP= 'OAN/SPM 1.5-m'
LATITUDE= '+31:02:39'
LONGITUD= '-115:27:49'
ALTITUDE=                  2800
OBSERVER= 'Vahram Chavushyan'
INSTRUME= 'RUCA    '
DETECTOR= 'SITe1 1k'
GAINMODE=                     1
GAIN    =                     5
RDNOISE =                  14.5
HISTORY  PMIS macros of 2001-01-23
HISTORY  Written by Stephen Levine
HISTORY  Modified by Gaguik Tovmassian
HISTORY  Modified by Alan Watson & Michael Richer
```

CONT.

```
WCSDIM  =                     2
LTM1_1  =                    1.
LTM2_2  =                    1.
WAT0_001= 'system=physical'
WAT1_001= 'wtype=linear'
WAT2_001= 'wtype=linear'
TRIM    = 'Jul 26 17:16 Trim data section is
[1:1024,1:1024]'
ZEROCOR = 'Jul 26 17:16 Zero level correction image is
Zero'
CCDMEAN =              12.15698
CCDMEANT=             806869787
CCDPROC = 'Jul 26 18:29 CCD processing done'
FLATCOR = 'Jul 26 18:29 Flat field image is FlatB with
scale=15929.22'
FILTERS =                    10
OBSERVAT= 'obspars '
ST      = '16:42:03.20'
EPOCH   =      2001.40406165231
AIRMASS =              2.168013
UTMIDDLE= '8:02:46.0'
```

Photometry:

2.- Updating the parameter files

- Datapars
- Centerpars
- Fitskypars
- photpars

FWHM : for each frame.

1. **Display the frame**
2. **Select the stars**
3. **imexame**
4. **a**
5. **Mopphat profile = FWHM**

datamin= -3x RN/gain

epar datapars

PACKAGE = apphot

TASK = datapars

(scale = 1.) Image scale in units per pixel
(fwhmpsf= **7.)** FWHM of the PSF in scale units
(emissio= yes) Features are positive ?
(sigma = INDEF) Standard deviation of background in counts
(datamin= **-20.)** Minimum good data value
(datamax= **640000.)** Maximum good data value
(noise = poisson) Noise model
(ccdread=) CCD readout noise image header keyword
(gain =) CCD gain image header keyword
(readnoi= **8.1000003814697**) CCD readout noise in electrons
(epadu = **1.2000000476837**) Gain in electrons per count
(exposur= **exptime**) Exposure time image header keyword
(airmass= **airmass**) Airmass image header keyword
(filter = **filters**) Filter image header keyword
(obstime= **UTMIDDle**) Time of observation image header keyword
(itime = 300.) Exposure time
(xairmas= 1.061735033989) Airmass
(ifilter= 20) Filter
(otime = 11:22:41.0) Time of observation
(mode = ql)

header

Detector data

PACKAGE = apphot

TASK = centerpars

(calgori= **centroid**) Centering algorithm
(cbox = **14.)** Centering box width in scale units
(cthresh= 0.) Centering threshold in sigma above background
(minsnra= 1.) Minimum signal-to-noise ratio for centering algorithm
(cmaxite= 15) Maximum number of iterations for centering algorithm
(maxshif= 7.) Maximum center shift in scale units
(clean = no) Symmetry clean before centering ?
(rclean = 1.) Cleaning radius in scale units
(rclip = 2.) Clipping radius in scale units
(kclean = 3.) Rejection limit in sigma
(mkcente= no) Mark the computed center on display ?
(mode = ql)

BOX ~ 2x FWHM

epar fitskypars

The same that the aperture

Between 5-7 pixels are good values

PACKAGE = apphot

TASK = fitskypars

(salgori= **mode**) Sky fitting algorithm

(annulus= **28.**) Inner radius of sky annulus in scale units

(dannulu= **7.**) Width of sky annulus in scale units

(skyvalu= 0.) User sky value

(smaxite= 10) Maximum number of sky fitting iterations

(sloclip= 0.) Lower clipping factor in percent

(shiclip= 0.) Upper clipping factor in percent

(snrejec= 50) Maximum number of sky fitting rejection iterations

(sloreje= 3.) Lower K-sigma rejection limit in sky sigma

(shireje= 3.) Upper K-sigma rejection limit in sky sigma

(khist = 3.) Half width of histogram in sky sigma

(binsize= 0.10000000149012) Binsize of histogram in sky sigma

(smooth = no) Boxcar smooth the histogram

(rgrow = 0.) Region growing radius in scale units

(mksky = yes) Mark sky annuli on the display

(mode = ql)

APERTURE: (3 - 5) x FWHM is suitable

Use the same aperture for all the files

PACKAGE = apphot

TASK = photpars

(weighti= constant) Photometric weighting scheme for wphot

(**apertur=** **28**) List of aperture radii in scale units

(zmag = 0.) Zero point of magnitude scale

(mkapert= yes) Draw apertures on the display

(mode = ql)

Photometry:

At last photometry!!! Task= phot

1. Choose the standard stars frames
2. **ap> display name_field**
3. **ap> phot name_field**
4. **ap> Interactive mode: mark and space bar... q and w..**
5. **ap> display new object... And so on...**

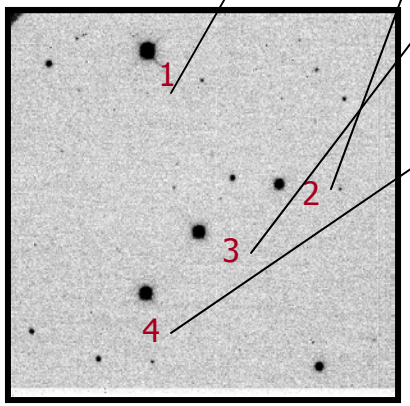
Do it for all the object frames.

It everything it is ok => output = **name_field.mag.1**...

OUTPUT: file.mag.1

Watch out!!!! **Use the finding charts and mark which are the stars and the position**.

You have to be secure which are the stars!!!!




```
#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
6044O          171.474  573.486  1    nullfile        0    \
 171.359  572.472  -0.115 -1.014 0.013  0.014      0   NoError  \
 8.955412    3.35599     1.292336    4756  20     0   NoError  \
 300.        1.392881    10              10:58:12.0         \
 28.00  84278.5    2463.271  62218.89   -5.792 0.005 0   NoError
6044O          459.482  621.488  2    nullfile        0    \
 457.901  622.247  -1.581 0.759  0.011  0.014      0   NoError  \
 9.052315    3.370655    1.395945    4703  68     0   NoError  \
 300.        1.392881    10              10:58:12.0         \
 28.00  87361.05   2463.345  65062.07   -5.841 0.005 0   NoError
6044O          507.484  549.486  3    nullfile        0    \
 503.432  548.867  -4.052 -0.619 0.012  0.014      0   NoError  \
 8.944       3.248924    0.9886579   4766  15     0   NoError  \
 300.        1.392881    10              10:58:12.0         \
 28.00  63776.81   2463.291  41745.13   -5.359 0.007 0   NoError
6044O          431.482  739.492  4    nullfile        0    \
 433.366  744.154  1.884  4.662  0.019  0.018      0   NoError  \
 8.753029    3.333308    1.379106    4746  30     0   NoError  \
 300.        1.392881    10              10:58:12.0         \
 28.00  46526.6    2463.267  24965.56   -4.801 0.011 0   NoError
6044O          291.478  313.478  5    nullfile        0    \
 289.711  313.905  -1.767 0.427  0.014  0.014      0   NoError  \
 8.978552    3.403276    0.7773539   4761  12     0   NoError  \
 300.        1.392881    10              10:58:12.0         \
 28.00  37700.41   2463.402  15582.63   -4.289 0.017 0   NoError
```

Calibration:

Now that we have instrumental magnitudes for our standard stars, we need to use these to find the transformations equations that will allow us to put our observations on the standard system.

$$b = B + const + colorterm \times (B-V) + Extinction \times Airmass$$

$$v = V + const + colorterm \times (B-V) + Extinction \times Airmass$$

Routines which do that are found in the **PHOTCAL** package

STEPS:

1. Create a catalogue that contains the magnitude and the color index of the standard stars on the standard systems (we have Landolt UBVRI standard) => it is included in IRAF = **nlandolt**
2. Create a "standard star observations" file which consist of the airmass, instrumental magnitudes and errors for each set of observations
3. Create a file containing the algebraic form of the transformation equations, and references to which columns in the tables contains which information.
4. Fit the transformation equations.

After that , you will need to create a "program star observations file" which consist of the airmass, instrumental magnitudes and errors for our problems stars. **And FINALLY you will then apply the transformations to these data.**

Calibration:

2. Create a "standard star observations" file which consist of the air-mass, instrumental magnitudes and errors for each set of observations : **mknobsfile**

FILE: standstars (input)

PG1047+003 : **6016O 6015O**

PG1657+078 : **6044O 6043O**

epar mknobsfile

PACKAGE = photcal

TASK = mknobsfile

photfile= ***.mag.1** The input list of APPHOT/DAOPHOT databases
idfilter= **10,20** The list of filter ids
imsets = **standstars**) The **input** image set file
observat= **standobs**) The **output** observations file
(wrap = yes) Format output file for easy reading ?
(obspara=) The input observing parameters file
(obscolu= 2,3,4,5) The format of obsparams
(minmage= 0.001) The minimum error magnitude
(shifts =) The input x and y coordinate shifts file
(apercor=) The input aperture corrections file
(apertur= 1) The aperture number of the extracted magnitude
(toleran= 10.) The tolerance in pixels for position matching
(allfilt= no) Output only objects matched in all filters
(verify = no) Verify interactive user input ?
(verbose= yes) Print status, warning and error messages ?
(mode = ql)

FILE : standobs (output)

# FIELD	FILTER	OTIME	AIRMASS	XCENTER	YCENTER	MAG	MERR
PG1047+003	10	4:59:52.0	1.589	357.998	276.332	-7.385	0.002
*	20	4:55:12.0	1.559	359.821	277.006	-8.114	0.002
PG1047+003A	10	4:59:52.0	1.589	499.673	439.014	-6.536	0.003
*	20	4:55:12.0	1.559	501.126	439.830	-8.062	0.002
PG1047+003B	10	4:59:52.0	1.589	714.124	565.891	-5.334	0.008
*	20	4:55:12.0	1.559	715.189	566.544	-6.819	0.005
PG1047+003C	10	4:59:52.0	1.589	362.005	919.935	-7.624	0.002
*	20	4:55:12.0	1.559	363.794	920.120	-9.112	0.001
PG1657+078	10	10:58:12.0	1.393	171.359	572.472	-5.792	0.005
*	20	10:53:09.0	1.370	172.976	573.256	-6.622	0.005
PG1657+078A	10	10:58:12.0	1.393	457.901	622.247	-5.841	0.005
*	20	10:53:09.0	1.370	458.901	623.337	-7.594	0.003
PG1657+078B	10	10:58:12.0	1.393	503.432	548.867	-5.359	0.007
*	20	10:53:09.0	1.370	504.408	549.882	-6.873	0.004
PG1657+078C	10	10:58:12.0	1.393	433.366	744.154	-4.801	0.011
*	20	10:53:09.0	1.370	434.438	745.065	-6.396	0.006
PG1657+078D	10	10:58:12.0	1.393	289.711	313.905	-4.289	0.017
*	20	10:53:09.0	1.370	290.969	315.036	-5.529	0.012

Calibration:

3. Create a file containing the algebraic form of the transformation equations, and references to which columns in the tables contains which information.

Standobs=input

I R A F

Image Reduction and Analysis Facility

PACKAGE = photcal

TASK = mkconfig

config = **spm.cfg** The new configuration file
catalog = **nlandolt** The source of the catalog format specification
observat= **standobs** The source of the observations file format specification
transfor= **nlandolt** The source of the transformation equations
(templat=) An existing template configuration file
(catdir =)_.catdir) The standard star catalog directory
(verify = no) Verify each new entry
(edit = yes) Edit the new configuration file
(check = yes) Check the configuration file
(verbose= yes) Verbose output
(mode = ql)

Watch out! We have to edit the output file and substitute **V by 20 and B by 10**

FILE:spm.cfg (output)

```
V          4                    # the V magnitude
BV         5                    # the (B-V) color
UB         6          # the (U-B) color
VR         7                    # the (V-R) color
RI         8                    # the (R-I) color
VI         9                    # the (V-I) color
error(V)  12                    # the V magnitude error
error(BV) 13                    # the (B-V) color error
error(UB) 14                    # the (U-B) color error
error(VR) 15                    # the (V-R) color error
error(RI) 16                    # the (R-I) color error
error(VI) 17                    # the (V-I) color error
# Declare the observations file variables
observations
T10        3          # time of observation in filter 10
X10        4          # airmass in filter 10
x10        5          # x coordinate in filter 10
y10        6          # y coordinate in filter 10
m10        7          # instrumental magnitude in filter 10
error(m10)  8         # magnitude error in filter 10
T20        10         # time of observation in filter 20
X20        11         # airmass in filter 20
x20        12         # x coordinate in filter 20
y20        13         # y coordinate in filter 20
m20        14         # instrumental magnitude in filter 20
error(m20)  15        # magnitude error in filter 20
# Sample transformation section for the new Landolt UBVRI system
fit   b1=0.0, b2=0.35, b3=0.000
const b4=0.0
BFIT : m10 = (BV + V) + b1 + b2 * X10 + b3 * BV + b4 * BV * X10
fit   v1=0.0, v2=0.17, v3=0.000
const v4=0.0
VFIT : m20 = V + v1 + v2 * X20 + v3 * BV + v4 * BV * X20
```

Calibration:

4. Let's fit the equations of transformations.

Task = fitparams (d (delete), f (new fit))

PACKAGE = photcal

TASK = fitparams

observat= **standobs** List of observations files
catalogs= **nlandolt** List of standard catalog files
config = **spm.cfg** Configuration file
paramete= **spm.ans** Output parameters file
(weighti= photometric) Weighting type (uniform,photometric,equations)
(addscat= yes) Add a scatter term to the weights ?
(toleran= 3.0000000000000E-5) Fit convergence tolerance
(maxiter= 15) Maximum number of fit iterations
(nreject= 0) Number of rejection iterations
(low_rej= 3.) Low sigma rejection factor
(high_re= 3.) High sigma rejection factor
(grow = 0.) Rejection growing radius
(interac= **yes**) Solve fit interactively ?
(logfile= testlog) Output log file
(log_unm= yes) Log any unmatched stars ?
(log_fit= yes) Log the fit parameters and statistics ?
(log_res= yes) Log the results ?
(catdir =)_.catdir) The standard star catalog directory
(graphic= stdgraph) Output graphics device
(cursor =) Graphics cursor input
(mode = ql)

FILE: spm.ans (output)

BFIT

status0(Solution converged)
variance 0.001274124
stdeviation 0.03569487
avsqerror 0.001272722
averror 0.03567524
avsqscatter 0.001151146
avscatter 0.03392854
chisqr 1.001101
msq 7.963273E-4
rms 0.02821927
reference m10
fitting (BV+V)+b1+b2*X10+b3*BV+b4*BV*X10
weights photometric
parameters 4
b1
(fit)
b2
(fit)
b3
(fit)
b4
(constant)
values 4
-21.32133
0.4513231
-0.198279
0.
errors 4
0.2033633
0.1333598
0.02866054

VFIT

status0(Solution converged)
variance 3.164825E-5
stdeviation 0.005625678
avsqerror 7.930226E-5
averror 0.008905182
avsqscatter 0.
avscatter 0.
chisqr 0.3990839
msq 1.582413E-5
rms 0.003977955
reference m20
fitting V+v1+v2*X20+v3*BV+v4*BV*X20
weights photometric
parameters 4
v1
(fit)
v2
(fit)
v3
(fit)
v4
(constant)
values 4
-22.03331
0.2888205
0.01111638
0.
errors 4
0.06382947
0.04226533
0.00680495

Photometry:

After that , you will need to create a "**program star observations file**" which consist of the air-mass, instrumental magnitudes and errors for **our problems stars** (NGC7006 in our case)

Image Reduction and Analysis Facility

PACKAGE = photcal

TASK = mknobsfile

photfile= ***.mag.2** The input list of APPHOT/DAOPHOT databases

idfilter= **10,20** The list of filter ids

imsets = **filestarsfin** The input image set file

observat= **filefin** The output observations file

(wrap = yes) Format output file for easy reading ?

(obspara=) The input observing parameters file

(obscolu= 2,3,4,5) The format of obsparams

(minmage= 0.001) The minimum error magnitude

(shifts =) The input x and y coordinate shifts file

(apercor=) The input aperture corrections file

(apertur= 1) The aperture number of the extracted magnitude

(toleran= 10.) The tolerance in pixels for position matching

(allfilt= no) Output only objects matched in all filters

(verify = no) Verify interactive user input ?

(verbose= yes) Print status, warning and error messages ?

(mode =

We had to run the PHOT task for the problems stars and obtain the file.mag.2

FILE IN: filestarfin

NGC7006 : 6048O 6047O

FILEOUT : filefin

# FIELD	FILTER	OTIME	AIRMASS	XCENTER	YCENTER	MAG	MERR
NGC7006-1	10	11:31:56.0	1.053	528.684	316.479	-6.127	0.003
*	20	11:22:41.0	1.062	529.543	318.365	-7.720	0.002
NGC7006-2	10	11:31:56.0	1.053	517.061	395.282	-5.789	0.004
*	20	11:22:41.0	1.062	517.981	397.084	-7.606	0.002
NGC7006-3	10	11:31:56.0	1.053	703.021	495.428	-4.685	0.008
*	20	11:22:41.0	1.062	703.591	497.176	-6.368	0.004
NGC7006-4	10	11:31:56.0	1.053	496.875	677.273	-3.525	0.022
*	20	11:22:41.0	1.062	497.947	678.703	-5.326	0.008

Photometry:

And FINALLY you will then apply the transformations to these data!!!!

I R A F

Image Reduction and Analysis Facility

PACKAGE = photcal

TASK = invertfit

observat= filefin **List of observations files**
config = spm.cfg **Configuration file**
paramete= spm.ans **Fitted parameters file**
calib = filefin.out **Output calibrated standard indices file**
(catalog= landolt) List of standard catalog files
(errors = obserrors) Error computation type (undefined,obserrors,equations)
(objects= all) Objects to be fit (all,program,standards)
(print =) Optional list of variables to print
(format =) Optional output format string
(append = no) Append output to an existing file ?
(catdir =)_.catdir) The standard star catalog directory
(mode = ql)

FILE (out): filefin.out

```
# Fri 01:52:00 29-Jul-2005
# List of observations files:
#                                   filefin
# Number of catalog files:
#                                   landolt
# Config:         spm.cfg
# Parameters:     spm.ans
#
# Computed indices for program and standard objects
#
# Columns:
#               1               object id
#               2               V
#               3               error(V)
#               4               resid(V)
#               5               BV
#               6               error(BV)
#               7               resid(BV)


NGC7006-1 13.997 0.002 INDEF 0.901 0.005 INDEF
NGC7006-2 14.107 0.002 INDEF 1.185 0.006 INDEF
NGC7006-3 15.347 0.004 INDEF 1.015 0.011 INDEF
NGC7006-4 16.388 0.008 INDEF 1.164 0.030 INDEF
```

Game over!!!!

Congratulations!!!

You have finished the first level!!!

This is only a very simple cookbook!

NOW you are ready to start the second level!

Read the manual!, use the help! Change the parameters! Experiment!

And ENJOYYYYYYYY!!!!

Adios, bye, adeus!!!