On the Measurement of Stellar Fluxes for the Study of the Interstellar Dust Extinction of M35

Marina von Steinkirch

(laboratory partners: A. Massari and M. von Hippel)

State University of New York at Stony Brook

Department of Physics and Astronomy

Received : April 7, 2012; accepted : April 4, 2012

ABSTRACT

BVI CCD photometry has been obtained for the intermediate-age open cluster M35 (NGC 2168) and and the standard star HD37557. We demonstrated every step of the calibration, astrometry, photrometry, and analysis. From the photometric proprieties of the objects, we studied the color indexes proprieties, deriving the visual extinction, A_V. The distance was calculated for two methods and the derivation of the age of the cluster was showed.

Subject headings: Imaging and Photometry: general — Imaging and Photometry: Interstellar dust extinction

1. Introduction to Interstellar Extinction

The study of *intermediate-age open clusters* can be used to the learn about the *stellar evolution* of intermediate-mass stars and the *dynamical evolution of clusters* together with their *interstellar extinction* (1).

The interstellar extinction is a sum of *absorption* and *scattering* of the light within the interstellar medium, being inversely proportional to the light wavelength. Consequently, a general increase in absorption toward shorter wavelengths is observed, resulting to the effect of *reddening*.

The effect of reddening can be measured by observing many wavelength images from the region and calculating the color excess. For instance, for the photometric bands given by B ($\lambda = 450$ nm) and V ($\lambda = 550$ nm), we define V_0, B_0, and $(B-V)_0$ as the intrinsic values for the magnitudes of the star and for the color index of the star, respectively. Moreover, we define A_B and A_V as the total extinction in these bands, such as

$$V = V_0 + A_V, \text{ and } B = B_0 + A_B. \tag{1}$$

For these wavelengths, the color excess can be then given by

$$\begin{aligned} E_{B-V} &= A_B - A_V, \\ &= (B-V) - (B-V)_0, \\ &= \left(\frac{A_B}{A_V} - 1\right) A_V, \end{aligned} \tag{2}$$

where the last result can be compared empirical values from our galaxy (1) and incorpored in the equation 1,

$$V_0 = V - 3.1 E_{B-V}, \tag{3}$$

giving the extinction corrected magnitudes.

The measurements of interstellar reddening (extinction), A_V, is a tool for estimating extinction on galactic open clusters. The *non-extincted color-magnitude diagram* can be constructed by correcting the cluster's *color-magnitude diagram* for the effect of extinction. Once the former is obtained, the distance and the age of the stellar population can be estimated by *(i) main sequence fitting*, and *(ii) model fitting* (2).

1.1. Efficiency on Photometry

Consider the light from a star like the Sun, this star will output ~ 1 solar luminosity. If there were no losses, the flux (power per unit area) measured in earth would be

$$F_0 = \frac{L}{4\pi d^2}$$

where d is the distance to the star. However there luminosity is lost by

- Interstellar space, which are tiny dust grains that absorb light.
- Earth's atmosphere, where the light is attenuated by dust scattering (the amount of dust in each cubic centimeter of air and how much air the starlight is going through). If a star is being observed on the zenith, the amount of air in which the light is going is small (1 airmass). However, if the star is on the horizon, it goes to several airmasses, and the light is reduced by

  $$F = F_0 e^{-\kappa z}$$

  where κ is an extinction coefficient depending on the amount of dust in the atmosphere and z is the airmass.

On the telescope, $\sim$ % 20 of the light is lost on the mirrors. After, the light encounter the filters, but they are also not 100% transparent. Finally the detector, with a range of efficiencies, and the CCD may be 90% efficient (3). Moreover, all these quantities are wavelength dependent and even though blue light is more energetic than red, in the detector both produce only one electron.

Naming all these efficiencies as d, and being t the exposure time, D the diameter of the telescope, c the speed of light, h the *Planck constant*, λ the mean wavelength of light by the filter, and $\Delta\lambda$ the filter bandpass (width of the filter), the number of counts that a detector on the ground detect is

$$count \propto F_0 t \frac{\pi D^2}{4} \frac{\lambda}{hc} \Delta\lambda e^{-\kappa z} f \tag{4}$$

We can divide both sides by the exposure time and write in terms of magnitude and considering that the constants do not change during the observations. Writing $\mathcal{C}$ as the count rate, one has

$$m = -2.5 \log \mathcal{C} + Az + Km, \tag{5}$$

To handle the problems of inefficiences, we observe a *standard start* with a known apparent magnitude, m, though different air masses, this give the constant K, so one can solve for A by noting how the number of counts we detect decreases as the airmass increases. This results can be used on our object to transform the observed count rate into apparent magnitude. From measuring the standard time for two different airmass and for the three filters, we can obtain the airmass correction by fiting (4),

$$V_{standard} - V_{measured} = \phi_V + \epsilon(B_{standard} - V_{standard}), \tag{6}$$

and

$$B_{standard} - V_{standard} = \phi_{BV} + \mu_{BV}(B_{measured} - V_{measured}), \tag{7}$$

into the data, where ϕ_{BV} and ϕ_V are zero-points, and μ_{BV} and ϵ are the transformation coefficients.

1.2. Main Sequence Fitting

The existence of a main sequence gives a tool for estimating distances to stars, since stellar parallax can only be applied accurately within ∼20 pc of sun. To determine the color of a star, one measures its apparent magnitude in two different filters (*e.g.,* B and V) and subtracts them. If the star is sitting on the main sequence, the color of the stars uniquely specifies its absolute magnitude. In order to get the distance to the star, one compares the star's apparent and absolute magnitude by applying

$$(m - M) = 5 \log d - 5, \tag{8}$$

i.e., the observed star is fit to the main sequence. Not all stars are on the main sequence, however, due the fact that all the stars of a star clusters are at the same distance, when one measures the entire group and plot apparent magnitude versus color, the non-main sequence stars become obvious.

2. Observational Methodology

By presenting a *CCD photometry* for the *M35 cluster*, we select photometric members and derive many of the the cluster evolution parameters, such as *interstellar reddening and extinction*, *distance to the cluster*, and *age*.

The reddening effects described in the last session is most clearly seen for a dust-obscured young (<500 Myr) open cluster, with early type stars (4). Messier 35 (NGC 2168), being an intermediate-age open cluster, is a suitable object for these studies. In addition, its celestial position at RA= $6^h8^m56.5^s$ and $\delta = 24^o21'.6$ (J2000), close to the galactic plane (M35 has galactic latitude of roughly 2.2^o and galactic longitude of nearly 186.6^o (5)), is convenient to our telescope localization. To derive the absolute magnitudes and to calculate the airmass, we performed also photometry on the standard HD37557, with 5h 40m 35.78s and declination 28 deg 58'36.9".

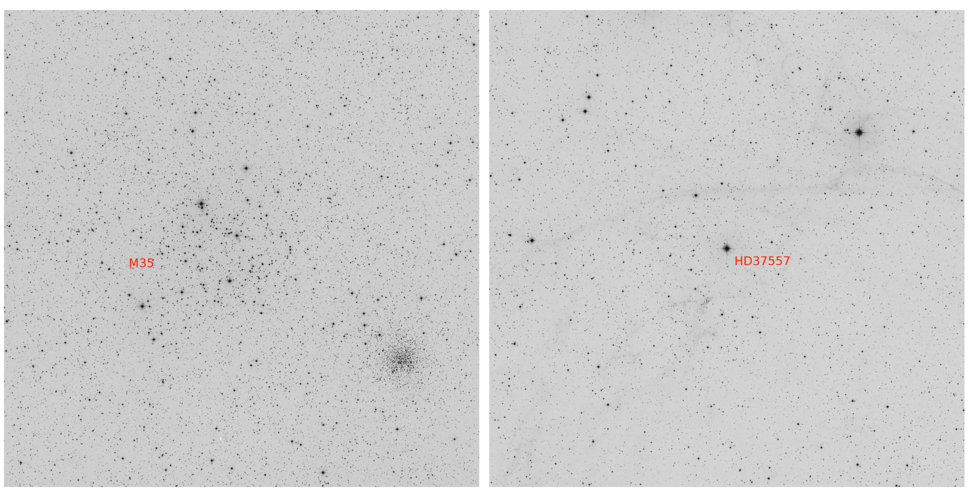

Fig. 1.— M35 and HD37557 Finding charts, obtained from (6).

These objects were observed with a SBIG STL-1001E CCD together with the M.t. Stony Brook 14-inch telescope, located at 4h52m 30s W Long, 40 deg 54'53" Lat, on March 26th, 2012, from around 9pm to 11pm Eastern Time. The sky conditions were clean but windy, and the moon was lunar crescent. We took exposures in three broad-band filters, B,V and I, exposing deep enough (through repeated exposures) to probe most of the main sequence. To calibrate our sciences, we took calibration exposures, *i.e.,* darks (in all the exposures) and flats, with high and low exposures in each of the filters.

For the absolute calibration of the apparent photometry, we performed similar measurements to a photometric standard, satisfying the requirement of being around 5^o of our science field and same airmass. The standards were measures at regular intervals (around 30 minutes) in each of the filters (B,V, and I) to monitor changes in the atmosphere transmissions. The choice of the standard was based on (8). The table with number of images, exposition time, and airmass are shown in the table 1, in the appendix.

3. Data Reduction

The CCD observations are registered as astronomical images (FITS, Flexibble Image Transport System), and they contain text headers that can be viewed to quickly identify contents of the file. All the observations were taken with 2x2 binning, that is, each pixel in the final FITS image is the average of for pixels. This has the advantage of reducing the readout time as well as the noise in each pixel. All the analysis were hard coded in IDL and the source codes are include in the appendix.

3.1. Calibration Process

The first step for the data reduction is to remove the various instrumental artifacts of the CCD, *i.e.,* the calibration of the images. First we subtract from them the *master dark frames*, *i.e.,* high signal-to-noise dark. We then median combine all the images by filter type (and by exposure time).

There is variation in the sensitivity of the CCD from one pixel to another (3), therefore dividing the images by the *flat field frames* is necessary for taking out responsivity variations. From equation 4 , we can claim that the number of counts recorded in the image on a pixel is a linear function of the flux falling in that pixel in the CCD,

$$\text{counts } = a * \text{ flux} + b,$$

where b is the bias (irrelevant here), and a is derived from the uniform/blank screen. Therefore, when we flat field the image, we rescale each pixel so a has a common value for all pixels. The flat field calibration frames are shown in the picture 2.

Finally, a bad pixel frame was constructed to all filters and reduced from all the images. Each step of the calibration of the Cluster and the standard images, in each of the three

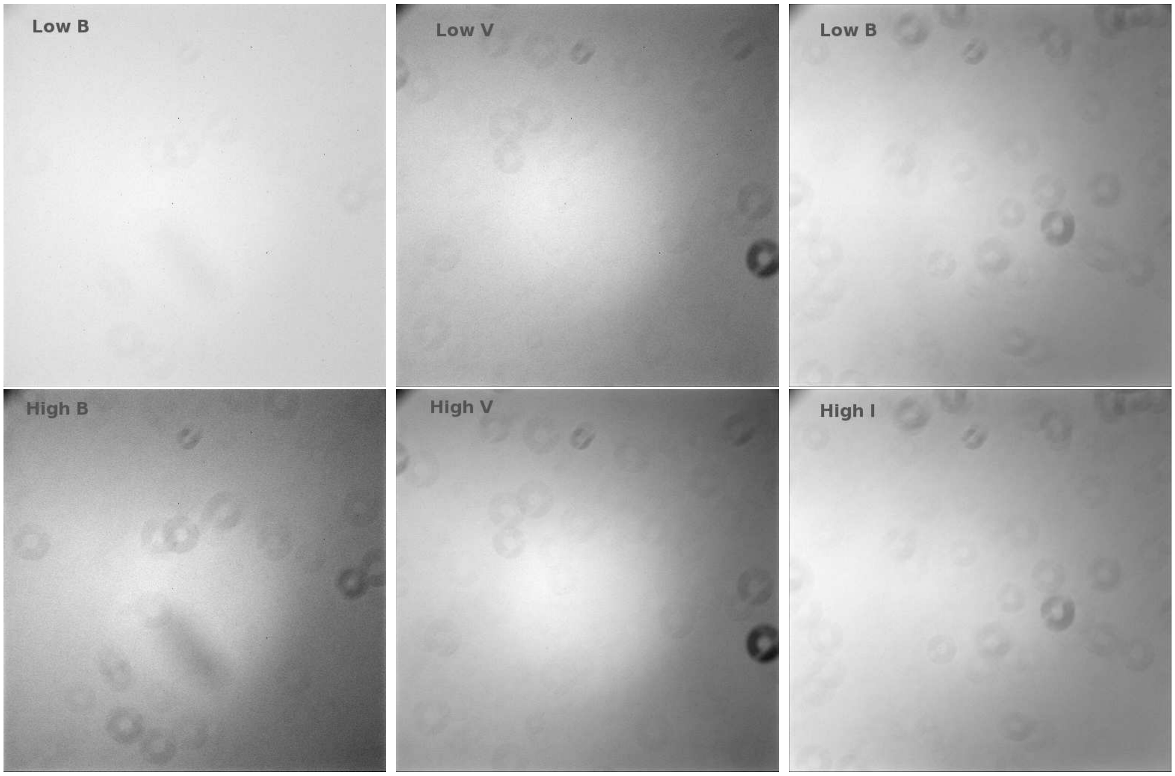

Fig. 2.— Flat frame, for low and high luminosity (exposures) and for the three filters, B, V, I, respectively. This process increases the values in the image where the CCD is less sensitive and decrease them where the CCD is more sensitive.

filters, is shown in the figures 3, 4, and 5.

3.2. Astrometry

We rotate the three-filters calibrated frames by 90^o left to have them on the normal orientation (N, E, S, W, by `rotate` in IDL). The astrometry of the calibrated images was performed under the software downloaded from (10) (`solve-field`). It builds a standard header, containing the coordinate transformation between the (X,Y) coordinates on the image and the equatorial coordinate system (RA, DEC) in J2000.

3.3. Airmass Curve

We measure the photometry for the standard star, and plot it as a function of airmass in each filter, to obtain the *airmass curve*, for each filter, figure 6.

Following equations 6 and 7, we linear fit the plots of $V_{standard} - V_{measured}$ vs $(B-V)_{standard}$ and $(B-V)_{standard}$ vs $(B-V)_{measured}$, and obtain the transformation coefficients. All the magnitudes of cluster elements were corrected from these results.

.

3.4. Photometry

Two IDL procedures are used to get magnitudes. The procedure `find` finds stars in an image and lists them. The procedure aper takes that list and measures the magnitudes or fluxes of the stars at the positions in the same list.

For the procedure `find`, the *minimum value above the background* for the brightest star should be corrected inspected. If setting too high, many faint stars will be lost. If it is too low, we will have too many fluctuations that are not real stars. Using `curval` command in IDL, we find approximated background values for the three filters, for M35 and the standard. We add three RMS for it (which for bright stars can be approximated by Poisson, *i.e.,* $\sim \sqrt{N}$, (12)). The FWHM should not be set too, which would results on getting "hot" pixels but not stars. Setting it too large would result on getting regions of anomalous sensitivity that the flat-fielding didn't complete removed. We inspect values some of these variables for the filters, and chose the FWHM to be 5. This results on around 300 values above the threshold, *i.e.,* 300 candidate to stars.

For the procedure `aper`, we aim a size of aperture that includes as much of the star's light as possible, without including too much sky's background. Since we trusted the telescope focus was properly calibrated not be properly calibrated, we do not need to choose a bigger aperture, which might result on lose the dim star in the noise and lose stars ins the overlap of the apertures. We will use the same values as the FWHM=5 from `find` and the around 100 stars were found.

3.5. Members Identification

Finally we match the star found with We identify the cluster member stars through comparisons with published literature, as in (11). In IDL, from routines such as `s`rcor, `a`d2xy, `m`akesastr, etc. (see code in the appendix section), we tried to matched as many as possible the clusters members. However, after some different algorithm tries we are able to numerically match not more than 5 stars with all the 3 filters.

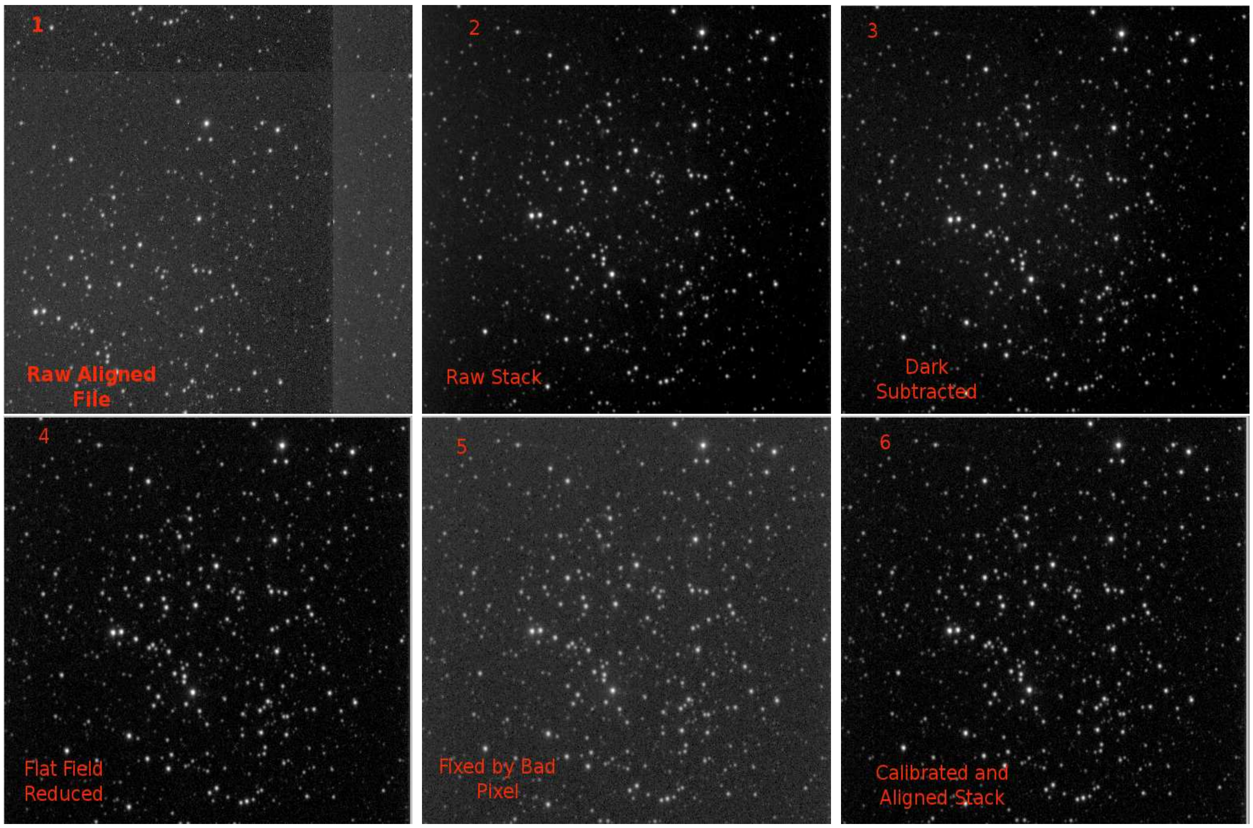

Fig. 3.— Calibration steps for M35 images, for the filter B: (1) Single 30s exposition after Gaussian alignment, (2) the raw stack of all 10 images, (3) the stack after being subtracted by the dark frame, (4) the stack after being divided by the flat field, (5) the image after the bad pixel mask, (6) the image after alignment with the other two filters (after astronometry).

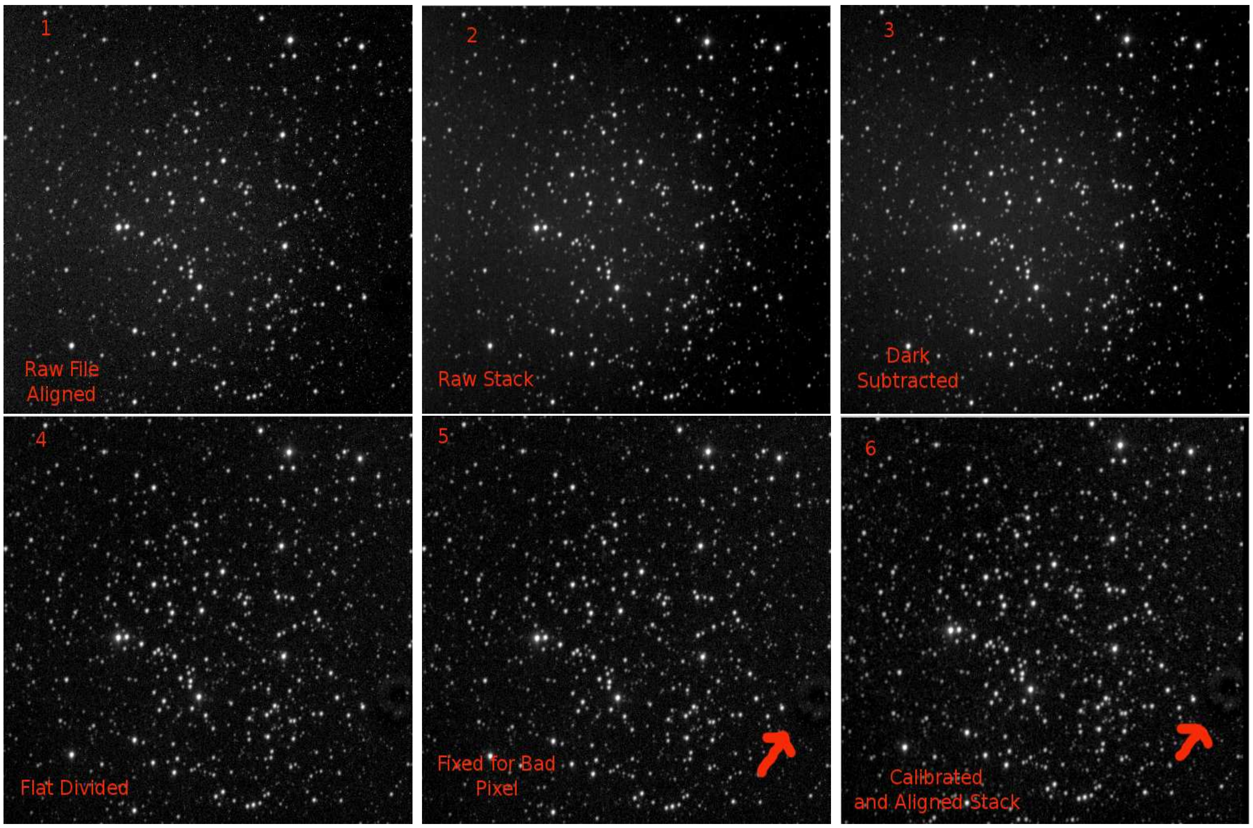

Fig. 4.— Calibration steps for the filter V: (1) Single 30s exposition after Gaussian alignment, (2) the raw stack of images, (3) the stack after being subtracted by the dark frame, (4) the stack after being divided by the flat field, (5) the image after the bad pixel mask, (6) the image after alignment with the other two filters (after astronomy).

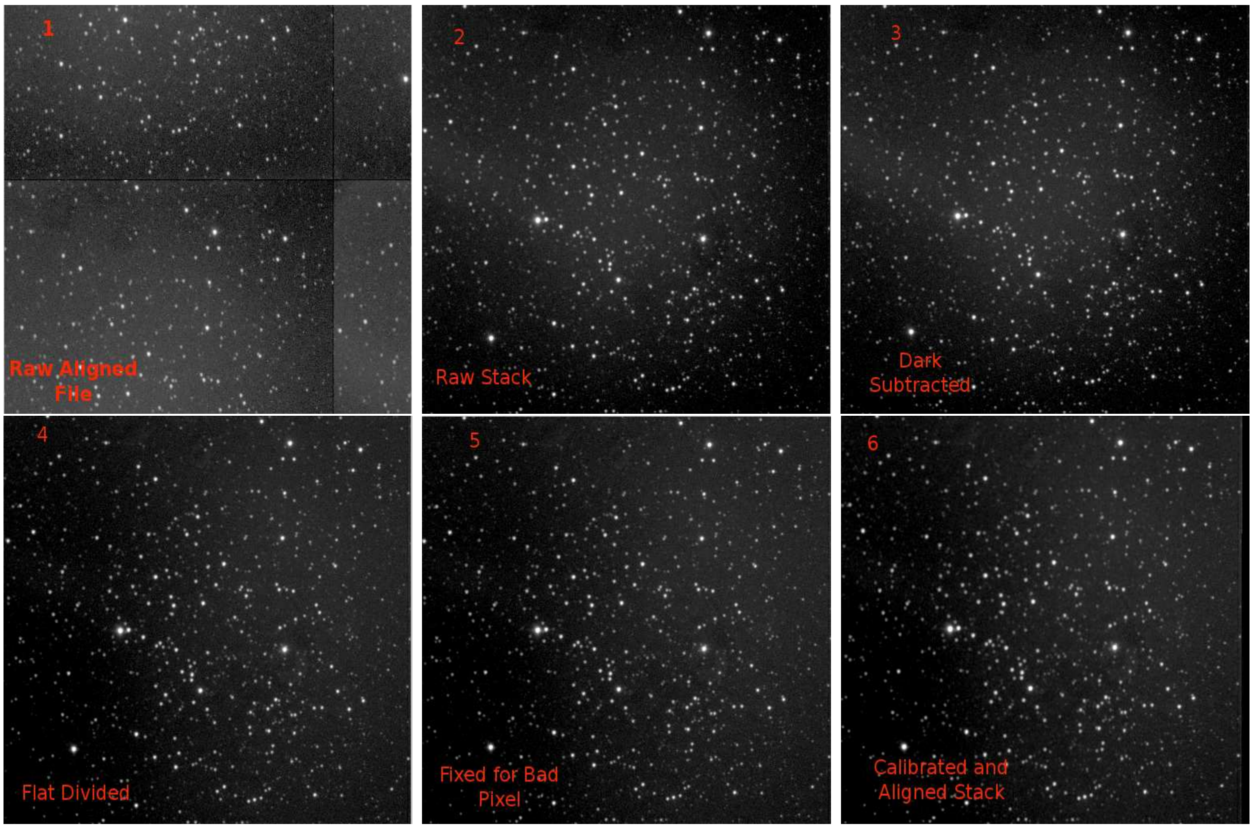

Fig. 5.— Calibration steps for the filter I: (1) Single 30s exposition after Gaussian alignment, (2) the raw stack of images, (3) the stack after being subtracted by the dark frame, (4) the stack after being divided by the flat field, (5) the image after the bad pixel mask, (6)) the image after alignment with the other two filters (after astronometry).

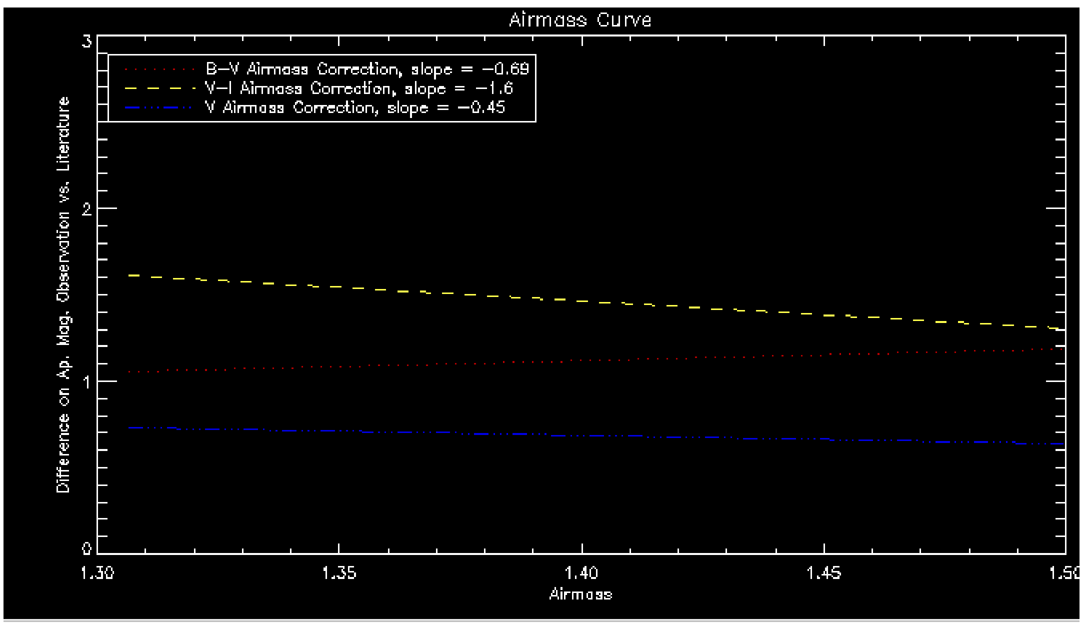

Fig. 6.— Airmass curve for our standard.

4. Data Analysis and Results

4.1. Infering the Extinction from Color-Color Diagram

We consider M32 as relative young open cluster, therefore we should expect many B-type stars in the main sequence. From (7), we choose the locus of the bluest and most probable element, B_0, table 7. A color-color diagram with our data, the reference data for M35, and the that main sequence locus is shown in the figure 10.

Sp	*M*(*V*)	*B* − *V*	*U* − *B*	*V* − *R*	*R* − *I*	$T_{\rm eff}$	BC
MAIN SEQUENCE, V							
O5	−5.7	−0.33	−1.19	−0.15	−0.32	42 000	−4.40
O9	−4.5	−0.31	−1.12	−0.15	−0.32	34 000	−3.33
B0	−4.0	−0.30	−1.08	−0.13	−0.29	30 000	−3.16
B2	−2.45	−0.24	−0.84	−0.10	−0.22	20 900	−2.35
B5	−1.2	−0.17	−0.58	−0.06	−0.16	15 200	−1.46
B8	−0.25	−0.11	−0.34	−0.02	−0.10	11 400	−0.80
A0	+0.65	−0.02	−0.02	0.02	−0.02	9 790	−0.30
A2	+1.3	+0.05	+0.05	0.08	0.01	9 000	−0.20
A5	+1.95	+0.15	+0.10	0.16	0.06	8 180	−0.15
F0	+2.7	+0.30	+0.03	0.30	0.17	7 300	−0.09
F2	+3.6	+0.35	0.00	0.35	0.20	7 000	−0.11
F5	+3.5	+0.44	−0.02	0.40	0.24	6 650	−0.14
F8	+4.0	+0.52	+0.02	0.47	0.29	6 250	−0.16
G0	+4.4	+0.58	+0.06	0.50	0.31	5 940	−0.18
G2	+4.7	+0.63	+0.12	0.53	0.33	5 790	−0.20
G5	+5.1	+0.68	+0.20	0.54	0.35	5 560	−0.21
G8	+5.5	+0.74	+0.30	0.58	0.38	5 310	−0.40
K0	+5.9	+0.81	+0.45	0.64	0.42	5 150	−0.31
K2	+6.4	+0.91	+0.64	0.74	0.48	4 830	−0.42
K5	+7.35	+1.15	+1.08	0.99	0.63	4 410	−0.72
M0	+8.8	+1.40	+1.22	1.28	0.91	3 840	−1.38
M2	+9.9	+1.49	+1.18	1.50	1.19	3 520	−1.89
M5	+12.3	+1.64	+1.24	1.80	1.67	3 170	−2.73
GIANTS, III							
G5	+0.9	+0.86	+0.56	0.69	0.48	5 050	−0.34
G8	+0.8	+0.94	+0.70	0.70	0.48	4 800	−0.42
K0	+0.7	+1.00	+0.84	0.77	0.53	4 660	−0.50
K2	+0.5	+1.16	+1.16	0.84	0.58	4 390	−0.61
K5	−0.2	+1.50	+1.81	1.20	0.90	4 050	−1.02
M0	−0.4	+1.56	+1.87	1.23	0.94	3 690	−1.25
M2	−0.6	+1.60	+1.89	1.34	1.10	3 540	−1.62
M5	−0.3	+1.63	+1.58	2.18	1.96	3 380	−2.48

Fig. 7.— Calibration of MK spectral types, (7)

The amount of reddening toward the open cluster M35 was inferred, $E_{b-V} = 0.74$, and

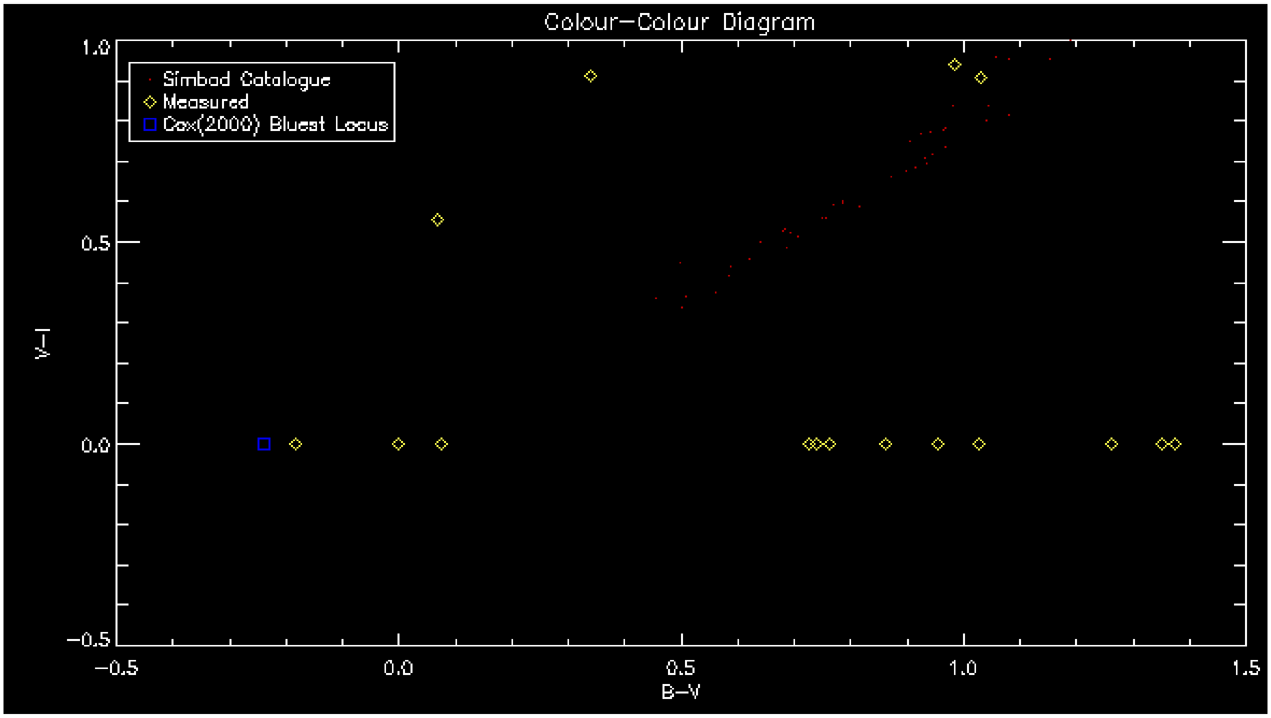

Fig. 8.— Color-color diagram of M35.

extinction, $A_V = 2.27$

4.2. Inferring Distance from a Color-Magnitude Diagram

We reduce A_V from our measured magnitude and E_{BV} from our measured color index. A color-magnitude diagram can be seen in the figure 9. We now compare to the entire main sequence from (7). We also include again the reference data for M35.

The amount of shift from the main sequence to our data is the *distance module* of the cluster, as in equation 8. We calculated numerically this distances and compare to the distance assuming the galactic extinction law (see the last part of the code, in the appendix). The first resulted $D_{shift} = 722 \pm 170$ pc, the second was calculated as $D_{galactic} = 593 \pm 170$

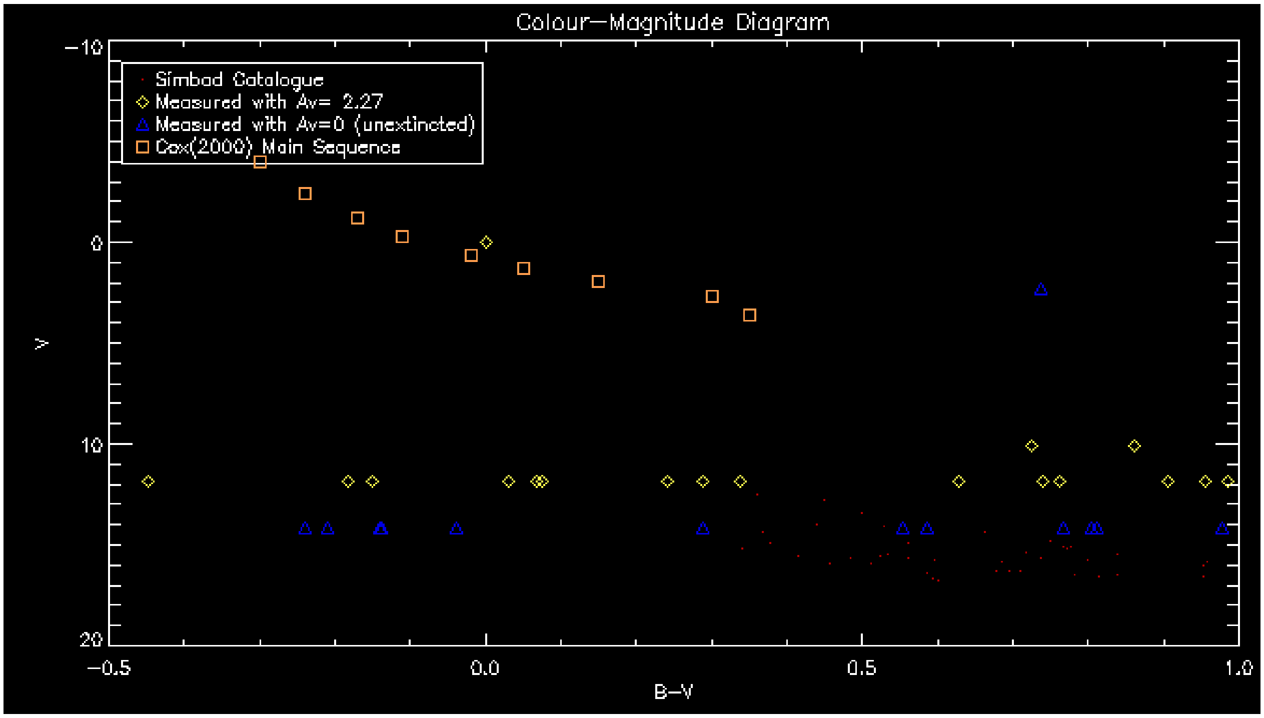

Fig. 9.— Color-magnitude diagram of M35.

pc. These values are somewhat close to the accepted value in the literature, $D_{literature} = 850$ pc.

4.3. Estimating the Age of the Cluster

We plot and fit two theoretical isochrones (obtained from (2)) in the color-magnitude diagram. These are absolute V magnitudes and B-V indexes for stellar models of many masses at fixed ages. To check how it varies with the extinction, and considering our A_V might be unsure, we fit our data to two isochornes, with $A_V = 0.6$ and $A_V = 2$. Both fits to the data, in an attempt to reproduce the actually value of an age of 100 millions of yers (**??**), returned values with $\chi^2 > 100$

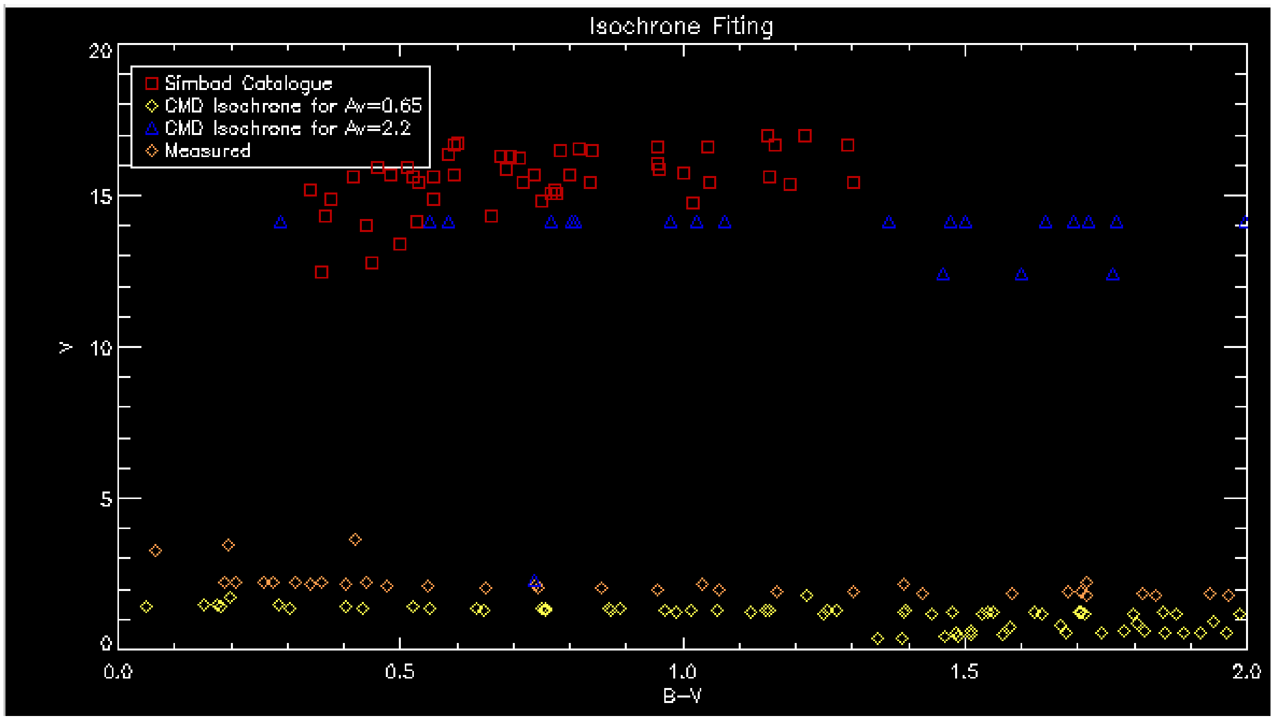

Fig. 10.— Color-magnitude diagram of M35 together with two isochrones and the reference data from literature.

Moreover, plotting the isochrones against to B-V should give a better approximation for the age of the cluster. This would be the most accurate way of measuring the distance, and can be performed in future studies.

5. Conclusion

We have showed every step of how to perform photometry in an astronomical object. Great emphasis in a good calibration and median sum of images frames were demonstrated. The calculation for corrections for the airmass was obtained through fitting, however these values were probably very off from the actual ones. This might be one of the reasons for the fact that the data was not very feasible when it was performed analysis. On the other hand, extensive and detailed code for calibration, astrometry, photometry, and analysis were developed and are available in the appendix. The quality of this set of data did not show in all the glory the relevant results that photometry yields, but a detailed study and reference for future studies were developed.

Some results were on the other hand compatible with the reference values. For instance, the distance of the cluster was correctly inferred by color shift, proving being superior than the results from only galactic extinction.

REFERENCES

Experiment description.

http://stev.oapd.inaf.it/cgi-bin/cmd

CCD Handbook.

Wikipedia, as in 03/2012.

D. Millsap, *Photometric Estimates of Metallicity of M35*, 2009.

http://archive.stsci.edu/cgi-bin/dss

Cox, A. N., Allens Astrophysical Quantities, 2000.

A. U. Landolt, *UBVRI Photometry of Stars Useful for Checking Equipment Orientation Stability*, 1983.,

http://www.briancasey.org/artifacts/astro/airmass.cgi, as in 03/2012.

astrometry.net

http://simbad.u-strasbg.fr/simbad

http://www.obs-hp.fr/ecole-ete/Photometry.pdf

Table 1. Log Sheet

File Number	Object	Exp. Time (s)	Filter	UT	Zenith (degrees)	Comments
1-10	Standard	15	B	02:19	40^o	
68-77	Standard	15	B	02:49	37^o	
12-20	Standard	15	V	02:24	40^o	
78-87	Standard	15	V	02:53	37^o	
25-34	Standard	5	I	02:30	40^o	
88-97	Standard	5	I	02:56	37^o	Dome in the way.
99-108	Standard	5	I	02:59	36^o	
36-45	M35	15	B	02:36	39^o	
47-56	M35	15	V	02:40	38^o	
58-67	M35	15	I	02:44	37.5^o	
115-124	Flat (high)	1	V			
126-135	Flat (low)	0.25	V			
176-185	Flat (high)	1	B			
186-195	Flat (low)	0.25	B			
204-213	Flat (high)	0.1	I			
215-224	Flat (low)	0.05	I			
225-234	Dark	0.1	I			
235-244	Dark	0.05	I			
136-145	Dark	0.25	V			
146-155	Dark	1	V			
156-165	Dark	5	I			
166-175	Dark	15	V			

```
@fixpix
@strc

pro calibration


;*******************************************************************************
;*******************************************************************************
;
; [MARINA VON STEINKIRCH, SPRING/2012]
;
; TO COMPILE IT IN IDL:
;	1) Make sure you have the folders defined in constants.pro.
;	2) Set the initial contant names and values in constants.pro.
;	3) Type IDL> .compile calibration
;	   Type IDL> calibration
;
; COLLECTION OF MACROS IN IDL TO PERFORM CALIBRATION, ASTROMETRY, PHOTOMETRY,
; AND TO STUDY THE INTERSTELLAR DUST EXTINCTION OF A OPEN CLUSTER:
;	1) aligning.pro
;	2) calibration.pro  (this)
;	3) astrometry.pro
;	4) airmass.pro
;	5) photometry.pro
;	6) diagrams.pro
;
; THIS MACRO WILL:
;	1) Read the images into array, and median combine to _raw_ stack.
;		a) The science in different filters (and align/astrometry).
;		b) The standard in different filters (and align/astrometry).
;		c) The flat in different filters and exposure times (low and high).
;		d) The dark in different exposure times (low, high, science, standard).
;	2) Create Master Dark Frame, reduce from science, standard, and flats, and
;	median combine to _dark_ stack.
;	3) Create Mater Flat Frame, reduce from science and stardard, and combine
;	the resulting science and standard images to _flat_ stack.
;	4) Create Bad Pixel Frame, reduce from combine from science and standard,
;	and median combine to _pix_ stack.
;
; (variable names c for constants, a for array, f for file names, i for images)
;
;*******************************************************************************
;*******************************************************************************

@constants
@calibration_arrays



;-------------------------------------------------------------------
; 1) Read the all the images of our science, put into an array (raw)
;-------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
	for i=0, cNumberFilesScience-1 do begin
		fSc =  cScienceFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align' + cNameExt
		iSc = readfits(fSc,hSc)
		iSc = float(iSc)
		aScience(j,i,*,*) = iSc
```

```
        endfor
        for i=0, 9 do begin
                fSt =  cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align_A1' + cNameExt
                iSt = readfits(fSt,hSt)
                iSt = float(iSt)
                aStandard(j,i,*,*) = iSt
        endfor
        for i=0, 9 do begin
                fSt2 =  cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align_A2' + cNameExt
                iSt2 = readfits(fSt2,hSt2)
                iSt2 = float(iSt2)
                aStandard2(j,i,*,*) = iSt2
        endfor
        for i=0, cNumberFilesFlat-1 do begin
                fFl = cFlatFolder + 'low/' + cFilterFolder[j]  + strtrim(i+1,2) + cNameExt
                iFl = readfits(fFl,hFl)
                iFl = float(iFl)
                aFlatLow(j,i,*,*) = iFl
        endfor
        for i=0, cNumberFilesFlat-1 do begin
                fFh = cFlatFolder + 'high/' + cFilterFolder[j]  + strtrim(i+1,2) + cNameExt
                iFh = readfits(fFh,hFh)
                iFh = float(iFh)
                aFlatHigh(j,i,*,*) = iFh
        endfor
endfor
for j=0, cNumberExpTime-1 do begin
        for i=0, cNumberFilesDark-1 do begin
                fDa = cDarkFolder[j]  + strtrim(i+1,2) + cNameExt
                iDa = readfits(fDa, hDa)
                iDa = float(iDa)
                aDark(j,i,*,*) = iDa
        endfor
endfor

for j=0, cNumberFilters-1 do begin
        for x=0,cFrameSize-1 do begin
                for y=0,cFrameSize-1 do begin
                        for i=0,cNumberFilesScience-1 do begin
                                aScienceFilter(i,x,y) = aScience(j,i,x,y)
                        endfor
                        for y=0,cFrameSize-1 do begin
                                for i=0,9 do begin
                                        aStandardFilter(i,x,y) = aStandard(j,i,x,y)
                                endfor
                                for i=0, 9 do begin
                                        aStandardFilter2(i,x,y) = aStandard2(j,i,x,y)
                                endfor
                        for i=0,cNumberFilesFlat-1 do begin
                                aFlatHighFilter(i,x,y) = aFlatHigh(j,i,x,y)
                                aFlatLowFilter(i,x,y) = aFlatLow(j,i,x,y)
                        endfor
                endfor
        endfor
        for x=0,cFrameSize-1 do begin
                for y=0,cFrameSize-1 do begin
                        aScienceFrame(x,y) = median( aScienceFilter(*,x,y), /even )
                        aStandardFrame(x,y) = median( aStandardFilter(*,x,y), /even )
```

```
				aStandardFrame2(x,y) = median( aStandardFilter2(*,x,y), /even )
				aFlatLowFrame(x,y) =  median( aFlatLowFilter(*,x,y), /even )
				aFlatHighFrame(x,y) =  median( aFlatHighFilter(*,x,y), /even )
			endfor
		endfor
		writefits, cOutStanFolder + 'standard_raw_A1_' + cFilters[j] + cNameExt, aStandardFrame, hSt
		writefits, cOutStanFolder + 'standard_raw_A2_' + cFilters[j] + cNameExt, aStandardFrame2, hSt2
		writefits, cOutSciFolder  + cNameScience + '_raw_' + cFilters[j] + cNameExt, aScienceFrame, hSc
		writefits, cOutCalFolder + 'FlatLow_' + cFilters[j] + cNameExt, aFlatLowFrame, hFl
		writefits, cOutCalFolder + 'FlatHigh_' + cFilters[j] + cNameExt, aFlatHighFrame, hFh
endfor



;-----------------------------------------------------------------------
;2) Substract dark.
;-----------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
		for x=0,cFrameSize-1 do begin
			for y=0,cFrameSize-1 do begin
				aDarkMedian(j,x,y) =  median( aDark(*,*,x,y), /even )
			endfor
		endfor
endfor

for j=0, cNumberFilters-1 do begin
		for m=0, cNumberExpTime-1 do begin
			for i=0, cNumberFilesScience-1 do begin
				if cExposureTimes(m) eq cExposureTimeScience then begin
					aScience(j,i,*,*) = aScience(j,i,*,*) - aDarkMedian(j,*,*)
				endif
			endfor

			for i=0, 9 do begin
				if cExposureTimes(m) eq cExposureTimeStandard then begin
					aStandard(j,i,*,*) = aStandard(j,i,*,*) - aDarkMedian(j,*,*)
				endif
			endfor

			for i=0, 9 do begin
				if cExposureTimes(m) eq cExposureTimeStandard then begin
					aStandard2(j,i,*,*) = aStandard2(j,i,*,*) - aDarkMedian(j,*,*)
				endif
			endfor

			for i=0, cNumberFilesFlat-1 do begin
				for k=0, cNumberFilters-1 do begin
					if cExposureTimes(m) eq cExposureTimesFlatLow(k) then begin
						aFlatLow(k,i,*,*) = aFlatLow(k,i,*,*) - aDarkMedian(k,*,*)
					endif
					if cExposureTimes(m) eq cExposureTimesFlatHigh(k)  then begin
						aFlatHigh(k,i,*,*) = aFlatHigh(k,i,*,*) - aDarkMedian(k,*,*)
					endif
				endfor
			endfor
		endfor
endfor
```

```
for j=0, cNumberFilters-1 do begin
	for x=0,cFrameSize-1 do begin
		for y=0,cFrameSize-1 do begin
			for i=0,cNumberFilesScience-1 do begin
				aScienceFilter(i,x,y) = aScience(j,i,x,y)
			endfor
			for i=0,9 do begin
				aStandardFilter(i,x,y) = aStandard(j,i,x,y)
			endfor
			for i=0, 9 do begin
				aStandardFilter2(i,x,y) = aStandard2(j,i,x,y)
			endfor
			for i=0,cNumberFilesFlat-1 do begin
				aFlatHighFilter(i,x,y) = aFlatHigh(j,i,x,y)
				aFlatLowFilter(i,x,y) = aFlatLow(j,i,x,y)
			endfor
		endfor
	endfor
	for x=0,cFrameSize-1 do begin
		for y=0,cFrameSize-1 do begin
			aStandardFrame(x,y) = median( aStandardFilter(*,x,y), /even )
			aStandardFrame2(x,y) = median( aStandardFilter2(*,x,y), /even )
			aFlatLowFrame(x,y) =  median( aFlatLowFilter(*,x,y), /even )
			aFlatHighFrame(x,y) =  median( aFlatHighFilter(*,x,y), /even )
		endfor
	endfor
	writefits, cOutStanFolder + 'standard_dark_A1_' + cFilters[j] + cNameExt, aStandardFrame, hSt
	writefits, cOutStanFolder + 'standard_dark_A2_' + cFilters[j] + cNameExt, aStandardFrame2, hSt2

	writefits, cOutSciFolder  + cNameScience + '_dark_' + cFilters[j] + cNameExt, aScienceFrame, hSc
	writefits, cOutCalFolder + 'FlatLow_' + cFilters[j] + cNameExt, aFlatLowFrame, hFl
	writefits, cOutCalFolder + 'FlatHigh_' + cFilters[j] + cNameExt, aFlatHighFrame, hFh
	for x=0,cFrameSize-1 do begin
		for y=0,cFrameSize-1 do begin
			aFlatHighMedian(j,x,y) = aFlatHighFrame(x,y)
			aFlatLowMedian (j,x,y) = aFlatLowFrame(x,y)
			for i=0,cNumberFilesScience-1 do begin
				aScience(j,i,x,y) = aScienceFrame(x,y)
			endfor
			for i=0,9 do begin
				aStandard(j,i,x,y) = aStandardFrame(x,y)
			endfor
			for i=0, 9 do begin
				aStandard2(j,i,x,y) = aStandardFrame2(x,y)
			endfor
		endfor
	endfor
endfor



;---------------------------------------------------------------------
; 3) Base-process by dividing by flat.
;---------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
	aVoid = Max(Histogram(aFlatHighMedian(j,*,*),OMIN=mn), mxpos)
	cMode = mn + mxpos
	aFlatHighNorm(j,*,*) = aFlatHighMedian(j,*,*)/cMode
```

```
endfor
for j=0, cNumberFilters-1 do begin
	for i=0, cNumberFilesScience-1 do begin
		aScience(j,i,*,*) = aScience(j,i,*,*)/aFlatHighNorm(j,*,*)
	endfor
	for i=0,9 do begin
		aStandard(j,i,*,*) = aStandard(j,i,*,*)/aFlatHighNorm(j,*,*)
	endfor
	for i=0, 9 do begin
		aStandard2(j,i,*,*) = aStandard2(j,i,*,*)/aFlatHighNorm(j,*,*)
	endfor
endfor

for j=0, cNumberFilters-1 do begin
	for x=0,cFrameSize-1 do begin
		for y=0,cFrameSize-1 do begin
			for i=0,cNumberFilesScience-1 do begin
				aScienceFilter(i,x,y) = aScience(j,i,x,y)
			endfor
			for i=0,9 do begin
				aStandardFilter(i,x,y) = aStandard(j,i,x,y)
			endfor
			for i=0, 9 do begin
				aStandardFilter2(i,x,y) = aStandard2(j,i,x,y)
			endfor
		endfor
	endfor
	for x=0,cFrameSize-1 do begin
		for y=0,cFrameSize-1 do begin
			aScienceFrame(x,y) = median( aScienceFilter(*,x,y), /even )
			aStandardFrame(x,y) = median( aStandardFilter(*,x,y), /even )
			aStandardFrame2(x,y) = median( aStandardFilter2(*,x,y), /even )
		endfor
	endfor

	writefits, cOutStanFolder + 'standard_flat_A1_' + cFilters[j] + cNameExt, aStandardFrame, hSt
	writefits, cOutStanFolder + 'standard_flat_A2_' + cFilters[j] + cNameExt, aStandardFrame2, hSt2
endfor
	writefits, cOutSciFolder  + cNameScience + '_flat_' + cFilters[j] + cNameExt, aScienceFrame, hSc
endfor


;--------------------------------------------------------------------
; 4) Fixing bad pixels.
;--------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
	aPixFlat(j,*,*) =  aFlatHighMedian(j,*,*)/aFlatLowMedian(j,*,*)
	for x=0, cFrameSize-1 do begin
		for y=0, cFrameSize-1 do begin
			aPix(x,y) = aPixFlat(j,x,y)
		endfor
	endfor
	aMean = mean(aPix)
	aDev = stddev(aPix)
	for x=0, cFrameSize-1 do begin
		for y=0, cFrameSize-1 do begin
			if aPix(x,y) - aMean lt 5*aDev then begin
				aCalFrame(x,y)=1
```

```
				endif else begin
					aCalFrame(x,y)=0.
				endelse
			endfor
		endfor
endfor

for j=0, cNumberFilters-1 do begin
		for x=0,cFrameSize-1 do begin
			for y=0,cFrameSize-1 do begin
				for i=0,cNumberFilesScience-1 do begin
					aScienceFrame(x,y) = aScience(j,i,x,y)
				endfor
				for i=0,9 do begin
					aStandardFrame(x,y) = aStandard(j,i,x,y)
				endfor
				for i=0, 9 do begin
					aStandardFrame2(x,y) = aStandard2(j,i,x,y)
				endfor
			endfor
		endfor
		fixpix, aScienceFrame, aCalFrame, aScienceFinal
		fixpix, aStandardFrame2, aCalFrame, aStandardFinal2
		fixpix, aStandardFrame2, aCalFrame, aStandardFinal2
		aScienceFinal = sigma_filter(aScienceFinal, N_SIGMA=5)
		aStandardFinal = sigma_filter(aStandardFinal, N_SIGMA=5)
		writefits, cOutStanFolder + 'standard_badpix_A1_' + cFilters[j] + cNameExt, aStandardFrame, hSt
		writefits, cOutStanFolder + 'standard_badpix_A2_' + cFilters[j] + cNameExt, aStandardFrame2, hSt2
		writefits, cOutSciFolder  + cNameScience  + '_badpix_'+ cFilters[j] + cNameExt, aScienceFrame, hSc
endfor

end
```

```
aScience = fltarr (cNumberFilters,cNumberFilesScience,cFrameSize,cFrameSize)
aStandard = fltarr (cNumberFilters, 10,cFrameSize,cFrameSize)
aStandard2 = fltarr (cNumberFilters, 10,cFrameSize,cFrameSize)
aFlatLow = fltarr (cNumberFilters, cNumberFilesFlat,cFrameSize,cFrameSize)
aFlatHigh = fltarr (cNumberFilters, cNumberFilesFlat,cFrameSize,cFrameSize)
aDark = fltarr (cNumberExpTime, cNumberFilesDark,cFrameSize,cFrameSize)

aScienceFilter = fltarr (cNumberFilesScience,cFrameSize,cFrameSize)
aStandardFilter = fltarr (10,cFrameSize,cFrameSize)
aStandardFilter2 = fltarr (10,cFrameSize,cFrameSize)
aFlatHighFilter = fltarr (cNumberFilesFlat,cFrameSize,cFrameSize)
aFlatLowFilter = fltarr (cNumberFilesFlat,cFrameSize,cFrameSize)

aFlatHighMedian = fltarr (cNumberFilters, cFrameSize,cFrameSize)
aFlatLowMedian = fltarr (cNumberFilters,cFrameSize,cFrameSize)
aScienceMedian = fltarr (cNumberFilters,cFrameSize,cFrameSize)
aStandardMedian = fltarr (cNumberFilters, cFrameSize,cFrameSize)
aStandardMedian2 = fltarr (cNumberFilters, cFrameSize,cFrameSize)
aDarkMedian = fltarr(cNumberExpTime,cFrameSize,cFrameSize)
aFlatHighNorm = fltarr (cNumberFilters,cFrameSize,cFrameSize)

aScienceFrame = fltarr(cFrameSize,cFrameSize)
aStandardFrame = fltarr(cFrameSize,cFrameSize)
aStandardFrame2 = fltarr(cFrameSize,cFrameSize)
aFlatLowFrame = fltarr (cFrameSize,cFrameSize)
aFlatHighFrame = fltarr (cFrameSize,cFrameSize)
aCalFrame = fltarr (cFrameSize,cFrameSize)

aScienceFinal = fltarr (cFrameSize,cFrameSize)
aStandardFinal = fltarr (cFrameSize,cFrameSize)
aStandardFinal2 = fltarr (cFrameSize,cFrameSize)
aPix = fltarr(cFrameSize,cFrameSize)
aPixFlat = fltarr(cNumberFilters,cFrameSize,cFrameSize)

aMean = fltarr(cNumberFilters)
aDev = fltarr(cNumberFilters)
```

```
;------------------------------------------------------------------------------------------
; Constants and variables.
; I use the following subfolders for data:
;                                   i) /Name_of_your_science with subfolders for each filter;
;                                   ii) /standard with subfolders for each filter;
;                                   iii) /flat with subfolders /high and /low and subsubfoldres
;                                               for each filter;
;                                   iv) /dark with subfolders for each exposure time.
;                                   v) output/science, output/calibration, output/standard,
;                                               output/photometry, output/airmass.
;
;------------------------------------------------------------------------------------------

cNameScience = 'm35'
cFrameSize = 512
cNumberFilesScience = 10
cNumberFilesStandard = 20
cNumberFilesFlat = 10
cNumberFilesDark = 10
cNameExt = '.fit'
cNameExtAlig = '.fits'

cNumberFilters = 3
cFilters=['B','V','I']
cFilterFolder=['B/','V/','I/']


az = [37,40, 39, 38, 37.5]
z = [1./(cos(az[0])),- 1./(cos(az[1])), 1./(cos(az[2])), 1./(cos(az[3])), 1./(cos(az[4])) ]


; -------------- Folder and File Destination:----------------------
cOutCalFolder = 'output/calibration/'
cOutSciFolder = 'output/science/'
cOutStanFolder = 'output/standard/'
cOutPhotoFolder = 'output/photometry/'
cOutAirFolder = 'output/airmass/'
cOutDiagFolder = 'output/diagrams/'


;------------ Calibration Constants ------------------------------
cNumberExpTime = 6
cScienceFolder = cNameScience + '/'
cStandardFolder =  'standard/'
cFlatFolder = 'flat/'
cExposureTimes=['005','01','025','1','5','15']
cDarkFolder = ['dark/005/','dark/01/','dark/025/','dark/1/','dark/5/','dark/15/']
cExposureTimeScience = cExposureTimes[5]
cExposureTimeStandard = [cExposureTimes[5], cExposureTimes[5], cExposureTimes[4]]
cExposureTimesFlatHigh = [ cExposureTimes[3], cExposureTimes[3],  cExposureTimes[1]] ; B,V,I
cExposureTimesFlatLow = [cExposureTimes[2],cExposureTimes[2], cExposureTimes[0]]


; ---------------- Photometry Constants --------------------------
;---- find
cHmin =[279+3*sqrt(270),850+3*sqrt(850), 3120+3*sqrt(3120)]
cFWHM = 5
```

```
;--- aper
cSharp= [0.2,1.0]
cRound = [-1.0,1.0]
cPhDig = 2.0
cApertures= [5]
cSkyRad = [10,20]
cBadPix= [0,0]
```

```
pro aligning


;**********************************************************************************
;**********************************************************************************
;
; [MARINA VON STEINKIRCH, SPRING/2012]
;
; TO COMPILE IT IN IDL:
;           1) Make sure you have the folders defined in constants.pro.
;           2) Set the initial contant names and values in constants.pro.
;           3) Type IDL> .compile aligning
;              Type IDL> aligning
;
; COLLECTION OF MACROS IN IDL TO PERFORM CALIBRATION, ASTROMETRY, PHOTOMETRY,
; AND TO STUDY THE INTERSTELLAR DUST EXTINCTION OF A OPEN CLUSTER:
;           1) aligning.pro (this)
;           2) calibration.pro
;           3) astrometry.pro
;           4) airmass.pro
;           5) photometry.pro
;           6) diagrams.pro
;
; THIS MACRO WILL:
;           1) Read the science and standard images into array, for differente filters.
;           2) Align all the images of the stack with the first one.
;           7) Run a gaussian fit to improve the alignment.
;
; (variable names c for constants, a for array, f for file names, i for images)
;
;**********************************************************************************
;**********************************************************************************

@constants


;----------------------------------------------------------------------
; 1) Read the all the images and their headers, align them with the
; first of the stack.
;----------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
        for i=0, cNumberFilesScience-1 do begin
                fSc =  cScienceFolder + cFilterFolder[j]  + strtrim(i+1,2) + cNameExt
                iSc = readfits(fSc,hSc)
                cRA = sxpar(hSc,'CRVAL1')
                cDEC = sxpar(hSc,'CRVAL2')
                if i eq 0 then begin
                        cRAref = cRA
                        cDECref = cDEC
                endif
                cRAoffset = cRAref - cRA
                cDECoffset = cDECref - cDEC
                cRAoffsetPixels = cRAoffset * 90000.            ; Converting degrees to pixels.
                cDECoffsetPixels = cDECoffset * 90000.
                iSc2 = shift(iSc, round(cRAoffsetPixels), round(cDECoffsetPixels) )
                writefits,  cScienceFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align0' + cNameExt, iSc2, hSc
        endfor
```

```
for i=0, 9 do begin
		fSt =  cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) + cNameExt
		iSt = readfits(fSt,hSt)
		cRA = sxpar(hSt,'CRVAL1')
		cDEC = sxpar(hSt,'CRVAL2')
		if i eq 0 then begin
			cRAref = cRA
			cDECref = cDEC
		endif
		cRAoffset = cRAref - cRA
		cDECoffset = cDECref - cDEC
		cRAoffsetPixels = cRAoffset * 90000.
		cDECoffsetPixels = cDECoffset * 90000.
		iSt2 = shift(iSt, round(cRAoffsetPixels), round(cDECoffsetPixels) )
		writefits, cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align0_A1' + cNameExt, iSt2, hSt
	endfor
	for i=0, 9 do begin
		fSt =  cStandardFolder + cFilterFolder[j]  + strtrim(i+11,2) + cNameExt
		iSt = readfits(fSt,hSt)
		cRA = sxpar(hSt,'CRVAL1')
		cDEC = sxpar(hSt,'CRVAL2')
		if i eq 10 then begin
			cRAref = cRA
			cDECref = cDEC
		endif
		cRAoffset = cRAref - cRA
		cDECoffset = cDECref - cDEC
		cRAoffsetPixels = cRAoffset * 90000.
		cDECoffsetPixels = cDECoffset * 90000.
		iSt2 = shift(iSt, round(cRAoffsetPixels), round(cDECoffsetPixels) )
		writefits, cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align0_A2' + cNameExt, iSt2, hSt

	endfor
endfor

;--------------------------------------------------------------------
; 2) Gaussian fit to improve the alignment.
;--------------------------------------------------------------------
for j=0, cNumberFilters-1 do begin
	for i=0, cNumberFilesScience-1 do begin
		fSc =  cScienceFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align0' + cNameExt
		iSc = readfits(fSc,hSc)
		cCutOut = iSc(291:497,215:446)
		aDummy = gauss2dfit(cCutOut,cCoefficients)
		cXcenter = cCoefficients(4)
		cYcenter = cCoefficients(5)
		if i eq 0 then begin
			cXref = cXcenter
			cYref = cYcenter
		endif
		if i eq 10 then begin
			cXref = cXcenter
			cYref = cYcenter
		endif
		cXoffset = cXref - cXcenter
		cYoffset = cYref - cYcenter
		iSc2 = shift(iSc, round(cXoffset), round(cYoffset) )
		writefits,  cScienceFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align' + cNameExt, iSc2, hSc
```

```
		endfor
		for i=0, 9 do begin
			fSt =  cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align0_A1' + cNameExt
			iSt = readfits(fSt,hSt)
			cCutout = iSt(291:497,215:446)
			aDummy = gauss2dfit(cCutout,cCoefficients)
			cXcenter = cCoefficients(4)
			cYcenter = cCoefficients(5)
			if i eq 0 then begin
				cXref = cXcenter
				cYref = cYcenter
			endif
			cXoffset = cXref - cXcenter
			cYoffset = cYref - cYcenter
			iSt2 = shift(iSt, round(cXoffset), round(cYoffset) )
			writefits, cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align_A1' + cNameExt, iSt2, hSt
		endfor
		for i=0 , 9 do begin
			fSt =  cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) + '_align0_A2' + cNameExt
			iSt = readfits(fSt,hSt)
			cCutout = iSt(291:497,215:446)
			aDummy = gauss2dfit(cCutout,cCoefficients)
			cXcenter = cCoefficients(4)
			cYcenter = cCoefficients(5)
			if i eq 0 then begin
				cXref = cXcenter
				cYref = cYcenter
			endif
			cXoffset = cXref - cXcenter
			cYoffset = cYref - cYcenter
			iSt2 = shift(iSt, round(cXoffset), round(cYoffset) )
			writefits, cStandardFolder + cFilterFolder[j]  + strtrim(i+1,2) +  '_align_A2' + cNameExt, iSt2, hSt
	endfor



	end
```

```
pro  photometry


;********************************************************************************
;********************************************************************************
;
; [MARINA VON STEINKIRCH, SPRING/2012]
;
; TO COMPILE IT IN IDL:
;            1) Make sure you have the folders defined in constants.pro.
;            2) Set the initial contant names and values in constants.pro.
;            3) Type IDL> .compile photometry
;               Type IDL> photometry
;
; COLLECTION OF MACROS IN IDL TO PERFORM CALIBRATION, ASTROMETRY, PHOTOMETRY,
; AND TO STUDY THE INTERSTELLAR DUST EXTINCTION OF A OPEN CLUSTER:
;            1) aligning.pro
;            2) calibration.pro
;            3) astrometry.pro
;            4) airmass.pro
;            5) photometry.pro  (this)
;            6) diagrams.pro
;
; THIS MACRO WILL:
;            1) Read the science in 3 filters.
;            2) Find positions and fluxes(find).
;            3) Do aperture photometry (aper).
;            4) Do cluster identification.
;            5) Save the magnitudes for each color.
;
;********************************************************************************
;********************************************************************************

@constants
@standard

;-------- Reading Catalogue from simbad
readcol,'m35.dat', NumberMember, NumberMember, oRA,oDE, VData, VIData, BVData, eVData, eVIData, eBVData, RA,
DE


; ------ Reading imagings in three filters
fS=  cOutSciFolder  + cFilters[0] + cNameExtAlig
iS0 = readfits(fS,hS0)
fS=  cOutSciFolder  + cFilters[1] + cNameExtAlig
iS1 = readfits(fS,hS1)
fS=  cOutSciFolder  + cFilters[2] + cNameExtAlig
iS2 = readfits(fS,hS2)


; ------- Aligning Images
hastrom, iS1, hS1, hS0, MISSING = 0
hastrom, iS2, hS2, hS0, MISSING = 0


;-------- Aperture Photometry
find, iS0, xS0, yS0, Flux0, Sharp, Round, cHmin[0], cFWHM, cRound, cSharp
aper, iS0, xS0, yS0, MagAper0, MagErr0,  Sky0, SkyErr0, cPhDig, cApertures, cSkyRad, cBadPix, /exact
```

```
aper, iS0, xS0, yS0, FluxAper, MagErr,  Sky, SkyErr, cPhDig, cApertures, cSkyRad, cBadPix, /exact, /flux

find, iS1, xS1, yS1, Flux1, Sharp, Round, cHmin[1], cFWHM, cRound, cSharp
aper, iS1, xS1, yS1, MagAper1, MagErr1,  Sky1, SkyErr1, cPhDig, cApertures, cSkyRad, cBadPix, /exact
aper, iS1, xS1, yS1, FluxAper1, MagErr1,  Sky1, SkyErr1, cPhDig, cApertures, cSkyRad, cBadPix, /exact, /flux

find, iS2, xS2, yS2, Flux2, Sharp, Round, cHmin[2], cFWHM, cRound, cSharp
aper, iS2, xS2, yS2, MagAper2, MagErr2,  Sky2, SkyErr2, cPhDig, cApertures, cSkyRad, cBadPix, /exact
aper, iS2, xS2, yS2, FluxAper2, MagErr2,  Sky2, SkyErr2, cPhDig, cApertures, cSkyRad, cBadPix, /exact, /flux



;---------- fitting (same fashion as in the airmas to get more information about the conversion)
cMagGood = where(MagAper1 gt 1 and MagAper1 le 30)
Mag1= MagAper1[cMagGood]
point1 = Max(Mag1)
point2 = Min(Mag1)
x1 =  - 2.5 * alog(Max(Flux1)/cExposureTimeScience)
x2 =  - 2.5 * alog(Min(Flux1)/cExposureTimeScience)
a1 = (point1-point2)/(x1-x2)
b1 = (point1+point2 - a1*(x1+x2))/2

cMagGood = where(MagAper0 gt 1 and MagAper0 le 30)
Mag0= MagAper0[cMagGood]
point1 = Max(Mag0)
point2 = Min(Mag0)
x1 =  - 2.5 * alog(Max(Flux0)/cExposureTimeScience)
x2 =  - 2.5 * alog(Min(Flux0)/cExposureTimeScience)
a0 = (point1-point2)/(x1-x2)
b0 = (point1+point2 - a0*(x1+x2))/2

cMagGood2 = where(MagAper2 gt 1 and MagAper2 le 30)
Mag2= MagAper2[cMagGood]
point1 = Max(Mag2)
point2 = Min(Mag2)
print, point1
print, point2
x1 =  - 2.5 * alog(Max(Flux2)/cExposureTimeScience)
x2 =  - 2.5 * alog(Min(Flux2)/cExposureTimeScience)
a2= (point1-point2)/(x1-x2)
b2 = (point1+point2 - a2*(x1+x2))/2



;----------- correcting airmass and calculating the magnitude
aMag0= - 2.5 * alog(Flux0/cExposureTimeScience)*a0 +  b0; - aAir[0]*z[0+2]
aMag1 = - 2.5 * alog(Flux1/cExposureTimeScience)*a1 + b1  - aAir[1]*z[1+2]
aMag2 = - 2.5 * alog(Flux2/cExposureTimeScience)*a2 + b2 - aAir[2]*z[2+2]

;---------- cluster identification
extast, hS0,astr
xy2ad, xS0, yS0,astr, A0, D0
srcor, RA, DE, A0,D0,30, ind01,ind02;,option=2
extast, hS1,astr
xy2ad, xS1, yS1,astr, A1, D1
srcor, RA, DE, A1,D1,5, ind11,ind12;,option=2
extast, hS2,astr
xy2ad, xS2, yS2,astr, A2, D2
```

```
srcor, RA, DE, A2,D2,40, ind21,ind22;,option=2

RAm0 = RA[ind02]
RAm1 = RA[ind12]
RAm2 = RA[ind22]
DEm0 = DE[ind02]
DEm1 = DE[ind12]
DEm2 = DE[ind22]
Mag0 = aMag0[ind01]
Mag1 = aMag1[ind11]
Mag2 = aMag2[ind21]

n0 = n_elements(Mag0)
n1 = n_elements(Mag1)
n2 = n_elements(Mag2)

BV = fltarr(n1)
V = fltarr(n1)
VI = fltarr(n1)


for i=0, n0-1 do begin
        for j=0, n1-1 do begin
                if RAm0[i] eq RAm1[j] and DEm0[i] eq DEm1[j] then begin
                        BV(i) = Mag0[i] - Mag1[j]
                        V(i) =    Mag1[j]
                endif
        endfor
endfor

for i=0, n0-1 do begin
        for j=0, n1-1 do begin
                if RAm1[i] eq RAm2[j] and DEm2[i] eq DEm2[j] then begin
                        VI(i) = Mag1[i] - Mag2[j]
                endif
        endfor
endfor


;--------- saving the color data
save, BV  , VI,  V, filename= cOutPhotoFolder+ 'color.dat'

end
```

```
pro diagrams


;*******************************************************************************
;*******************************************************************************
;
; [MARINA VON STEINKIRCH, SPRING/2012]
;
; TO COMPILE IT IN IDL:
;       1) Make sure you have the folders defined in constants.pro.
;       2) Set the initial contant names and values in constants.pro.
;       3) Type IDL> .compile diagrams
;          Type IDL> diagrams
;
; COLLECTION OF MACROS IN IDL TO PERFORM CALIBRATION, ASTROMETRY, PHOTOMETRY,
; AND TO STUDY THE INTERSTELLAR DUST EXTINCTION OF A OPEN CLUSTER:
;       1) aligning.pro
;       2) calibration.pro
;       3) astrometry.pro
;       4) airmass.pro
;       5) photometry.pro
;       6) diagrams.pro  (this)
;
; THIS MACRO WILL:
;       1) Read data from photometry and plot diagrams.
;       2) Plot color-color and find extinction.
;       3) Plot magnitude-color and find distance and age.
;
;*******************************************************************************
;*******************************************************************************

@constants


; -------Restoring data from photometry.

restore,cOutPhotoFolder +'color.dat' ;        BV  , VI,  V
readcol,'m35.dat', NumberMember, NumberMember, oRA,oDE, VData, VIData, BVData, eVData, eVIData, eBVData, RA,
DE
readcol,'iso.sav', isoB, isoV
readcol,'iso2.sav', isoB2, isoV2


coxx = fltarr(1)
coxx(0)= -0.24
coxy = fltarr(1)
a = where(Min(BV))
coxy(0)= VI[a]

bluest = Min(BV) ; larger index, redder
Ebv = coxx - bluest
Av = 3.086*(Ebv)
print, Ebv
print, Av


;------- Colour-Colour Diagram
;device, decomposed=0
```

```
;loadct,5

names= ['Simbad Catalogue', 'Measured', 'Cox(2000) Bluest Locus']
dots = [3,4,6]
colors = [100, 200, 300]

;plot, VI , BV,/nodata, psym=3, title='Colour-Colour Diagram',xtitle='B-V',ytitle='V-I',xrange=[-0.5,1.5], yrange=[-0.5,1]
;legend, names, color = colors, psym = dots
;oplot,  VIData, BVData , psym=dots[0], color = colors[0]
;oplot, BV, VI , psym=dots[1], color = colors[1]
;oplot,  coxx, coxy, psym = dots[2], color = colors[2]
;iPic = tvrd()
;write_png, cOutDiagFolder +  'cc.png', iPic


;------ Colour-Magnitude Diagram
cox2x= [ -0.3,-0.24,-0.17,-0.11,-0.02, 0.05,0.15, 0.3, 0.35]
cox2y=[-4,-2.45,-1.2, -0.25,0.65,1.3, 1.95, 2.7, 3.6]
n = n_elements(BV)
Bun = fltarr(n)
Vun = fltarr(n)
for i=0, n-1 do begin
        Bun[i] = BV[i]+Ebv
        Vun[i] = V[i]+Av
endfor


names= ['Simbad Catalogue', 'Measured with Av= 2.27', 'Measured with Av=0 (unextincted)',  'Cox(2000) Main Sequence']
dots = [3,4,5,6]
colors = [100, 200,300, 400]

;plot,  BV,V,/nodata, psym=3, title='Colour-Magnitude Diagram',xtitle='B-V',ytitle='V',yrange=[20, -10], xrange=[-.5,1]
;legend, names, color = colors, psym = dots
;oplot,   BVData , VData, psym=dots[0], color = colors[0]
;oplot,  BV, V , psym=dots[1], color = colors[1]
;oplot, Bun, Vun , psym=dots[2], color = colors[2]
;oplot,  cox2x, cox2y, psym = dots[3], color = colors[3]
;iPic = tvrd()
;write_png, cOutDiagFolder +  'mc.png', iPic



; ----------- Calculating distances
;----- galactic ext law
m35galactic = 2.2
m= (Av*sin(m35galactic))/18
dgal= 10^((m-Vun+5)/5)
dgal=10^6*dgal
print, dgal


;----- main sequence fitting
deltaV=cox2y - Vun
d = 10^(1 + deltaV/5)
print, d*10^(6)


;------- Estimating age from isochones
```

```
names= ['Simbad Catalogue', 'CMD Isochrone for Av=0.65','CMD Isochrone for Av=2.2' , 'Measured']
dots = [6,4,5, 4]
colors = [100, 200,300, 400]

;plot,  isoV,isoB-isoV,/nodata, psym=3, title='Isochrone Fiting',xtitle='B-V',ytitle='V',yrange=[0, 20], xrange=[0,2]
;legend, names, color = colors, psym = dots
;oplot,   BVData , VData, psym=dots[0], color = colors[0]
;oplot,  isoV, isoB - isoV , psym=dots[1], color = colors[1]
;oplot,  isoV2, isoB2 - isoV2 , psym=dots[3], color = colors[3]
;oplot, Bun, Vun , psym=dots[2], color = colors[2]
;iPic = tvrd()
;write_png, cOutDiagFolder +  'iso.png', iPic


; ---- Fiting
Age=10^(8)
yfit = curvefit(Vun, Bun,1, Age)
print, isoV
print, chi




end
```